Spatio-temporal Graph-based Demand and Capacity Balancing Considering Flight Uncertainties

Yutong Chen[a], Ramon Dalmau[b], Sameer Alam[a,*]
[a] Air Traffic Management Research Institution, Nanyang Technological University, Singapore
[b] EGSD/INO/ENG, EUROCONTROL Innovation Hub (EIH), Brétigny-Sur-Orge
{yutong.chen, sameeralam}@ntu.edu.sg, {ramon.dalmau-codina}@eurocontrol.int

***Abstract*—In uncertain environments, Demand and Capacity Balancing (DCB) operations may benefit from shifting focus towards the tactical phase. Currently, air traffic control (ATC) and air traffic flow management (ATFM) operate on different time scales—ATC is tactical, while ATFM is pre-tactical and strategic. Despite these differences, both aim to improve safety and efficiency. Integrating DCB into the tactical phase allows for more responsive management of dynamic conditions. The HYPERSOLVER project, part of the SESAR ER program, is focused on developing and implementing integrated ATFM and ATC methods to address these evolving challenges effectively. In this context, this paper proposes a spatio-temporal graph-based DCB approach to rapidly solve DCB problems in large-scale high-density scenarios considering uncertainty (e.g., flight speed). Simulation experiments based on real European airspace scenarios demonstrate the proposed method can quickly and efficiently solve large-scale DCB problems in high-density scenarios (solving a DCB instance for 2,000 flights in 9.68 seconds, while the rate of changed flights, the average delay time for delayed flights, and the rate of additional flight time for rerouted flights are only 8.46%, 12.2 minutes, and 9.34%, respectively).**

***Keywords*—demand and capacity balancing, air traffic flow management, ground delay, rerouting, uncertainty, weighted directed graph**

I. Introduction

One of the main challenges currently facing the development of global civil aviation is the widening gap between the increasing air traffic demand and the saturated airspace capacity, known as the demand-capacity imbalance. When demand surpasses capacity in a given area of airspace (e.g., a sector) during a time window, it increases the workload on controllers, causes airspace congestion, and results in flight delays [1]. This area, during the time window, is referred to as a *hotspot* [2]. Consequently, Demand and Capacity Balancing (DCB) has become crucial for the aviation industry. It is among the seven key operational concepts in Air Traffic Management (ATM) [3]. As highlighted by Single European Sky ATM Research (SESAR), DCB is expected to play a significant role in the future ATM system as part of network management, contributing to the reduction of flight delays [4].

DCB is to minimise the effects of ATM system constraints, which will be capable of evaluating system-wide traffic flows and capacities to implement necessary actions in a timely manner [5]. Although adjusting capacity (typically through dynamic sectorisation) is a feasible approach to DCB, adjusting demands (typically through trajectory modification) is a more commonly studied approach. In strategies, trajectory modification is normally achieved by methods such as ground delay (e.g., ground delay programs in U.S. and ATFM regulations in Europe) [6], rerouting [7], and/or level capping. Ground delay postpones the departure time of flights, along with the controlled times of arrival (CTAs) at all waypoints correspondingly, to effectively avoid hotspots. Rerouting involves altering a flight's path, allowing it to bypass hotspots by flying through less congested sectors. Both methods aim to balance demand and capacity by changing the spatio-temporal distribution of flights within the airspace. Compared to rerouting, Ground delay only adjusts the trajectory in the time dimension, resulting in lower computation complexity and easier implementation, but generally offers less optimisation. In algorithms, both exact solution methods and approximate solution methods are commonly considered. Exact solution algorithms [8] have the advantage of yielding a globally optimal solution. However, for large-scale problems, these methods may not guarantee timely completion. As a result, exact solution methods are hardly ever applied in practice. Approximate solution algorithms [9], on the other hand, often use heuristic frameworks to find locally optimal solutions within a reasonable time frame. The computation time for these methods is generally less sensitive to the problem scale compared to exact solution methods. Nevertheless, locally optimal solutions are often less desirable due to the potentially significant gap between local and global optimality.

Considering computational speed and ease of use, approximate algorithms based on ground delay are widely used in current practice. An example is the Computer-Assisted Slot Allocation (CASA) system employed in Europe [10]. CASA assigns delays on a First-Come, First-Served (FCFS) basis to maintain fairness by treating all flights equally, and it has been highly effective in delivering safe operations for over three decades. However, this method does not treat delay assignments as a formal optimisation problem. Reinforcement Learning (RL) is being studied for application to DCB problems to achieve a better balance between computational speed and optimisation. Rerouting and considering uncertainty are important ways to improve the optimisation and resilience of DCB methods, respectively, while both CASA and RL methods currently face challenges in these two aspects. Therefore, this study aims to develop a DCB method that balances

computational speed and optimisation. This method needs to integrate both ground delay and rerouting strategies while being compatible with uncertainties in flight.

To achieve this vision, this paper proposes a spatio-temporal graph-based DCB method that considers flight uncertainty. The contributions of this work are summarised as follows:

1) The DCB problem is transformed into a hotspot-free trajectory sequential planning problem based on FCFS to reduce the complexity of the problem. The hotspot-free trajectory planning problem is further converted into a path search problem based on a spatio-temporal graph, and the optimal solution is efficiently obtained using a path search with an admissible heuristic function.
2) A demand count model is designed to be compatible with flight speed uncertainty, aiming to identify hotspots in a dynamic environment. A complex hotspot identification model based on joint probabilities is derived into an iterative expression to significantly improve the speed of hotspot identification.
3) This method integrates rerouting and ground delay, where an adaptive delay strategy ensures feasible solutions.

The rest of the paper is organised as follows: section II describes the DCB problem under study, section III provides a detailed introduction to the proposed DCB method, section IV presents the experimental methods and analyses the results and section V summarises the main findings of the research.

II. Problem Description

This study was performed as a part of a SESAR exploratory research project, HYPERSOLVER: Artificial Intelligence controller able to manage Air traffic Control (ATC) and Air Traffic Flow Management (ATFM) within a single framework. In the HYPERSOLVER framework (as shown in Figure 1), ATFM and ATC are designed as a holistic closed-loop system, serving as inputs and feedback to each other. ATFM is used for density management, and due to the need to dynamically respond to changes in the tactical phase of ATC, its implementation time is set to one hour before departure. ATC is used for bunching, proximity, and conflict management (using a hierarchical strategy with decreasing granularity). Their implementation times are set to 20, 40, and 60 minutes before the Closest Point of Approach (CPA), respectively. Among them, bunching management is a tactical density management method that identifies and eliminates hotspots based on grid airspace (e.g., 50×50 square nautical miles); proximity management is used to eliminate potential conflicts in advance; conflict management is implemented to ensure safe separation when a conflict is imminent. AI component includes a DCB solver and an ATC solver, which are used for human-AI teaming with the Flow Manager (FM) and Air Traffic Controller (ATCer) through Human-Machine Interfaces (HMI) to support the holistic ATFM-ATC system operation. This research focuses on developing the DCB solver to meet HYPERSOLVER's design requirements:

1) **Dynamic Environment**: The solver needs to consider uncertainties to some extent in order to cope with changes in the tactical operating environment.
2) **High-Speed Computation**: It should be capable of performing rapid calculations close to the time of departure to enable continuous and fast loop iterations with ATC.
3) **Fairness and Transparency**: The algorithmic strategies should be fair and transparent to gain acceptance from the community.
4) **Human-AI teaming**: It should allow for a high degree of customisation in optimisation objectives, constraints, and DCB strategy selection (e.g., ground delay and rerouting), to ensure compatibility with the operational performance preferences of FMs or airspace users.

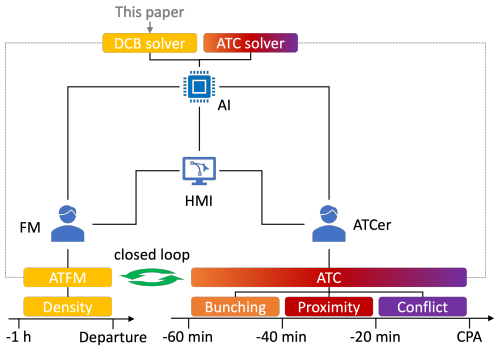


Figure 1. HYPERSOLVER framework: holistic ATFM-ATC system based on human-AI teaming. This paper focuses on the DCB solver.

In this study, the DCB problem is described as adjusting the spatio-temporal distribution of traffic through rerouting and ground delay to avoid hotspots in the airspace. A hotspot is defined as a situation where the demand within an air traffic service unit (ATSU) exceeds its capacity within a specific time window (as illustrated in Figure 2). To clarify further, the DCB problems are explained as follows:

- Assumptions
  1) The flight's scheduled speed is constant during the en-route phase, while the actual speed is uncertain (e.g., due to uncertain winds).
  2) The change in aircraft motion state is instantaneous.
  3) Only the 2D planar structure of ATSU is considered.
  4) Changes in flight altitude are not considered.
  5) Trajectory adjustment includes ground delay and rerouting.
  6) Demand counting uses the occupancy count [11], that is, a flight is counted as a demand for an ATSU within a specific time window as long as it is present in that ATSU during the time window.
  7) Each flight has a preset maximum flight distance in the airspace considering fuel consumption.
  8) The FCFS principle is applied, which fosters fairness, as CASA does nowadays in an excellent way.
  9) The planned trajectory has the shortest flight distance.

- Constraints
  1) DCB constraint: In any time window of any ATSU, demand must be always less than or equal to capacity.
  2) Flight distance constraint: The flight distance within the airspace must not exceed a preset value to avoid excessive fuel consumption.
- Objectives
  1) Minimise delay time.
  2) Minimise additional flight distance.

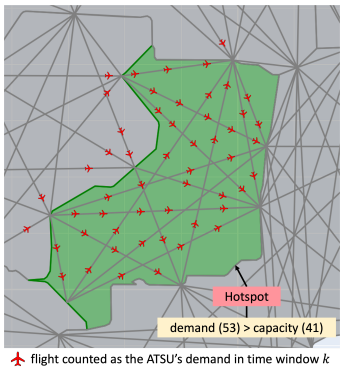


Figure 2. Demand and capacity balancing problem. Hotspots are identified based on time windows. Since this study uses occupancy count, some flights that are about to enter the ATSU (in green) within the snapshot will also be included in the demand. E.g., since the ATSU's demand and capacity are 53 and 41, respectively, it is identified as a hotspot during time window k.

III. Methodology

The proposed method transforms the DCB problem into a hotspot-free trajectory sequential planning problem. Figure 3 illustrates its technical scheme, which consists of five components: (1) the **demand count model considering flight speed uncertainty** (subsection III-A) is used to handle uncertainty in demand counting, supporting (2) the **DCB fast discriminant based on probability** (subsection III-B) for rapid hotspot identification. For hotspot-free trajectory planning for each flight, (3) an **adaptive weighted directed spatio-temporal graph generation** (subsection III-C) is employed, followed by (4) an **optimal path search with an admissible heuristic function** (subsection III-D) for optimisation. Finally, (5) an **FCFS-based adaptive postponement iteration algorithm** (subsection III-E) is used to ensure trajectory generation and sequential planning.

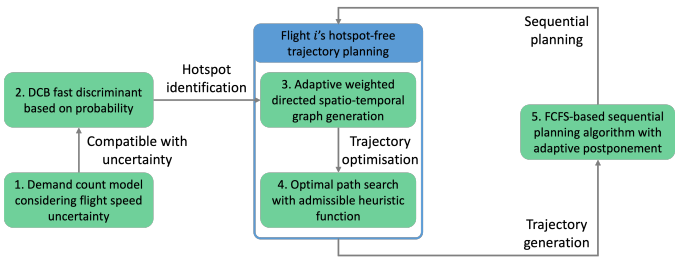


Figure 3. Technical scheme of the proposed DCB method.

A. Demand count model considering flight speed uncertainty

As shown in Figure 4, the planned trajectory of the flight passes through several ATSUs. For ease of calculation, flight speed uncertainty is represented by the uncertainty in the time at which the flight enters the ATSU. Therefore, the probability distribution of the time at which flight i enters ATSU j can be represented by a probability density function $f(t)$:

$$f(t) \sim U(t_{i,j}^{\text{Entry}}, \sigma^2(t_{i,j}^{\text{Before}})) \tag{1}$$

where $t_{i,j}^{\text{Entry}}$ represents the planned times at which flight i enters ATSU j. U can represent an arbitrary distribution with a mean of $t_{i,j}^{\text{Entry}}$ and a variance of $\sigma^2(t_{i,j}^{\text{Before}})$. $t_{i,j}^{\text{Before}}$ represents the flight i's flight time before entering ATSU j and $\sigma^2(t_{i,j}^{\text{Before}})$ is a function of it. $\sigma^2(t_{i,j}^{\text{Before}})$ increases with $t_{i,j}^{\text{Before}}$, since the uncertainty of the time to the waypoint usually increases with the duration of the flight.

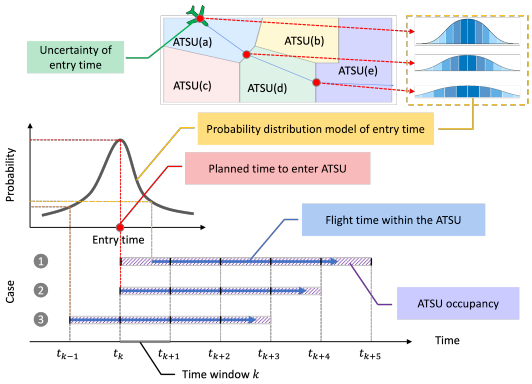


Figure 4. Demand count diagram. The three curves in the upper right show the probability distributions of a flight's arrival times at three waypoints (the entry points of the corresponding ATSU). As flight time progresses, uncertainty (variance) increases, leading to a flattening trend in the curves. The planned entry time for the ATSU is used as the distribution's mean. The lower part of the figure illustrates the flight's occupancy of the ATSU at three possible entry times for example.

The time window set is represented by $k \in K$. Time window k is expressed as $[t_k, t_{k+1})$, with time window duration ΔT. Therefore, $t_k = k\Delta T$. Since this study uses the occupancy count method for demand calculation, the occupation of ATSUs varies depending on the time at which a flight enters the ATSU. This is specifically defined as the demand within the corresponding time window for the ATSU.

It can be derived that when flight i enters the ATSU j within the time range $\left[k\Delta T - t_{i,j}^{\text{In}}, (k+1)\Delta T\right)$, flight i occupies time window k of ATSU j. $t_{i,j}^{\text{In}}$ is flight i's flight time in ATSU j and it can be expressed as $t_{i,j}^{\text{In}} = \frac{s_{i,j}^{\text{In}}}{v_i}$, where v_i is flight i's planned flight speed. Consequently, the probability $p_{i,j,k}$ that flight i occupies time window k of ATSU j can be expressed as:

$$p_{i,j,k} = \int_{k\Delta T - t_{i,j}^{\text{In}}}^{(k+1)\Delta T} f(t)dt \tag{2}$$

B. DCB fast discriminant based on probability

When calculating occupancy probability, the occupancy of time window k of ATSU j by different flights are assumed as independent events. Therefore, the probability $p_{j,k}^{\mathrm{NH}}$ that time window k of ATSU j is not a hotspot is expressed as:

$$p_{j,k}^{\mathrm{NH}} = \sum_{q=0}^{n_{j,k}^{\mathrm{C}}} p_{j,k}^{q} \tag{3}$$

where $n_{j,k}^{\mathrm{C}}$ is the capacity of ATSU j during time window k. $p_{j,k}^{q}$ is the probability that the demand of ATSU j during time window k is q, which can be expressed as:

$$p_{j,k}^{q} = \binom{q}{n_{j,k}^{\mathrm{MD}}} \prod_{i=1}^{n_{j,k}^{\mathrm{MD}}} (p_{i,j,k})^{\lambda_i} (1-p_{i,j,k})^{1-\lambda_i} \tag{4}$$

where $\binom{q}{n_{j,k}^{\mathrm{MD}}}$ represents a combinatorial number, indicating the selection of q events occurring from a total of $n_{j,k}^{\mathrm{MD}}$ events, with the remaining events not occurring. $n_{j,k}^{\mathrm{MD}}$ represents the maximum number of flights that may occupy time window k of ATSU j. λ_i is an event occurrence indicator, taking the value 1 when the corresponding event occurs and 0 otherwise. When $p_{j,k}^{\mathrm{NH}} = 1$ when $n_{j,k}^{\mathrm{MD}} > n_{j,k}^{\mathrm{C}}$, the following equation can be derived by substituting Equation 4 into Equation 3:

$$p_{j,k}^{\mathrm{NH}} = \sum_{q=0}^{n_{j,k}^{\mathrm{C}}} \left[\binom{q}{n_{j,k}^{\mathrm{MD}}} \prod_{i=1}^{n_{j,k}^{\mathrm{MD}}} (p_{i,j,k})^{\lambda_i} (1-p_{i,j,k})^{1-\lambda_i} \right] \tag{5}$$

It should be noted that $p_{j,k}^{\mathrm{NH}} = 1$ when $n_{j,k}^{\mathrm{MD}} \leq n_{j,k}^{\mathrm{C}}$.

To more flexibly and efficiently utilise airspace resources, an overload tolerance $\zeta \in [0, 1)$ for a time window of an ATSU is defined. Specifically, it ensures the probability that the demand within the ATSU is greater than its capacity during the time window is less than or equal to ζ. Therefore, the DCB constraint can be expressed as:

$$1 - p_{j,k}^{\mathrm{NH}} \leq \zeta \tag{6}$$

Assuming the probability of a newly added flight (e.g., flight i) occupying ATSU j during time window k is $p_{i,j,k}$, according to the current values of $p_{j,k}^{q}$ and $p_{j,k}^{q-1}$, the iteration can be carried out as follows:

$$p_{j,k}^{q} \leftarrow p_{j,k}^{q}(1-p_{i,j,k}) + p_{j,k}^{q-1} p_{i,j,k}, q = 0, 1, ..., n_{j,k}^{\mathrm{MD}} \tag{7}$$

where when no flight that may occupy ATSU j during time window k, $p_{j,k}^{1} = 0$ and $p_{j,k}^{0} = 1$. To maintain iterative consistency, we set $p_{j,k}^{-1} = 0$, which is consistent with the physical interpretation. Based on Equation 7, it can be derived:

$$\sum_{q=0}^{Q} p_{j,k}^{q} \leftarrow \sum_{q=0}^{Q} \left[p_{j,k}^{q}(1-p_{i,j,k}) + p_{j,k}^{q-1} p_{i,j,k} \right] \tag{8}$$

Considering $p_{j,k}^{-1} = 0$, Equation 8 can be written as:

$$\sum_{q=0}^{Q} p_{j,k}^{q} \leftarrow \sum_{q=0}^{Q} p_{j,k}^{q} - p_{j,k}^{Q} p_{i,j,k} \tag{9}$$

It should be noted that when $n_{j,k}^{\mathrm{MD}} < Q$, $p_{j,k}^{Q} = 0$.

By instantiating Equation 9, we obtain the iterative expression for the probability $p_{j,k}^{\mathrm{NH}}$:

$$p_{j,k}^{\mathrm{NH}} \leftarrow p_{j,k}^{\mathrm{NH}} - p_{j,k}^{n_{j,k}^{\mathrm{MD}}} p_{i,j,k} \tag{10}$$

Additionally, according to Equation 6, it can be obtained that, when $n_{j,k}^{\mathrm{MD}} \geq n_{j,k}^{\mathrm{C}}$ (i.e., $p_{j,k}^{n_{j,k}^{\mathrm{C}}} \neq 0$),

$$p_{i,j,k} \leq \frac{p_{j,k}^{\mathrm{NH}} + \zeta - 1}{p_{j,k}^{n_{j,k}^{\mathrm{C}}}} \tag{11}$$

Therefore, we use $p_{j,k}^{\mathrm{RM}}$ to represent the remaining maximum occupancy probability of ATSU j during time window k by the currently added flight, subject to satisfying the DCB constraint. It can be expressed as:

$$p_{j,k}^{\mathrm{RM}} = \begin{cases} \frac{p_{j,k}^{\mathrm{NH}} + \zeta - 1}{p_{j,k}^{n_{j,k}^{\mathrm{C}}}} & n_{j,k}^{\mathrm{MD}} \geq n_{j,k}^{\mathrm{C}} \\ 1 & \text{otherwise} \end{cases} \tag{12}$$

Ultimately, it can be conveniently determined whether the newly added flight i satisfies the DCB constraint by the following discriminative condition:

$$p_{i,j,k} \leq p_{j,k}^{\mathrm{RM}} \tag{13}$$

C. Adaptive weighted directed spatio-temporal graph generation

The purpose of adaptive weighted directed spatio-temporal graph generation is to enable the use of graph-based path search algorithms to obtain the optimal trajectory. Constructing a graph based on the spatio-temporal information of vehicle movements is a common method in road traffic management [12]. In this study, the primary method involves first establishing a planar waypoint-based graph within the airspace where DCB is to be implemented. Subsequently, a spatio-temporal graph is generated based on flight and DCB requirements to represent the feasible solution space, where any path from the origin to the destination on this graph represents a feasible solution. 'Adaptive' means that a unique spatio-temporal graph is generated for each flight.

In this study, waypoints on the ATSU boundaries (i.e., entry and exit points) and their connections within the ATSU are used as vertices and edges, respectively, to generate the planar graph G^{P} for the DCB scenario (Figure 5 as an example). Then, the specific steps for a flight's adaptive weighted directed spatio-temporal graph generation are as follows:

1) Start from the vertex where flight i enters the airspace (origin vertex) as the current vertex.
2) Connect the current vertex with its neighbours as directed edges if the edge meets the *constraints* (see below); otherwise, the neighbour is marked as closed.
3) Sequentially take each open tree tip as the current vertex and repeat Step 2 until all tree tips are closed.

The edge generation satisfies the following constraints:

1) **DCB constraint**: The flight must not cause the overload probability of the ATSU to exceed the tolerance.

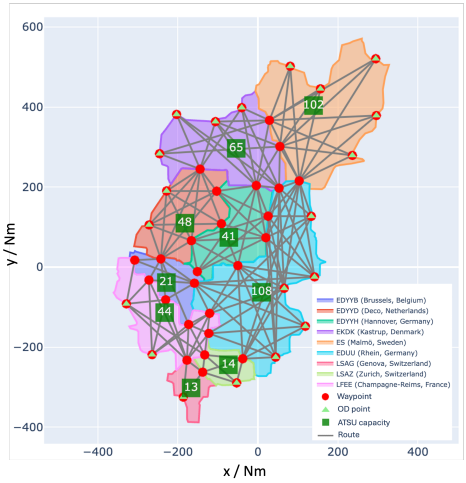


Figure 5. Airspace with a generated planar graph, also used as the scenario of this study's simulation experiments, including 9 ATSUs.

2) **Flight distance constraint**: The potential minimum flight distance of the flight must not exceed the preset maximum flight distance within the airspace of interest.
3) **ATSU continuity constraint**: Two connected edges must not lie within the same ATSU, in compliance with ATM operating rules (flights must be handed over to the next ATSU at waypoints on the ATSU boundary).
4) **Approaching constraint**: To improve the efficiency of graph generation, the tail (target) of a directed edge is always closer (straight-line distance) to the destination than its head (source).

For the sake of clarity, we use $v^{\oplus}$ to represent vertex $\oplus$ and $v^{\oplus} = (x^{\oplus}_{i,j}, y^{\oplus}_{i,j}, t^{\oplus}_{i,j})$, the former two parameters are position coordination and the third one is the corresponding CTA. $\oplus$ can be S, T, O and D representing directed edge e's source and target, flight i's origin and destination of the airspace of interest, respectively. S, T, O and D are for Source, Target, Origin and Destination, respectively. Therefore, $t^{\mathrm{T}}_{i,j} = t^{\mathrm{S}}_{i,j} + \frac{s^{\mathrm{In}}_{i,j}}{v_i}$ and $s^{\mathrm{In}}_{i,j} = \sqrt{(x^{\mathrm{T}}_{i,j} - x^{\mathrm{S}}_{i,j})^2 + (y^{\mathrm{T}}_{i,j} - y^{\mathrm{S}}_{i,j})^2}$.

Then, the DCB constraint can be represented as:

$$p_{i,j,k} \leq p^{\mathrm{RM}}_{j,k}, k \in K^{\mathrm{S,T}}_{i,j} \tag{14}$$

where $K^{\mathrm{S,T}}_{i,j}$ represents the set of time windows during which flight i occupies ATSU j, with $(x^{\mathrm{S}}_{i,j}, y^{\mathrm{S}}_{i,j}, t^{\mathrm{S}}_{i,j})$ and $(x^{\mathrm{T}}_{i,j}, y^{\mathrm{T}}_{i,j}, t^{\mathrm{T}}_{i,j})$ as the source and target (including CTAs).

The flight distance constraint can be expressed as:

$$t^{\mathrm{T}}_{i,j} + \frac{d^{G^{\mathrm{P}}}_{\min}(v^{\mathrm{T}}, v^{\mathrm{D}})}{v_i} \leq t^{\mathrm{O}}_i + \Delta t(v_i, f_i) \tag{15}$$

where $d^{G^{\mathrm{P}}}_{\min}(v^{\mathrm{T}}, v^{\mathrm{D}})$ represents the minimum flight distance from v^{T} to v^{D} in planar graph G^{P} and $\Delta t(v_i, f_i)$ represents the maximum flight time for flight i, given a maximum fuel amount of f_i planned for this airspace and a speed of v_i, where factors affecting fuel consumption such as flight altitude, aircraft weight are omitted for the sake of simplicity.

The ATSU continuity constraint can be expressed as:

$$\mathrm{sub}(e^{\mathrm{S,T}}) \subseteq \left\{ e^{\mathrm{S'},\mathrm{T'}} \mid \mathrm{S'} = \mathrm{T}, \mathrm{T'} \notin j_{e^{\mathrm{S,T}}} \right\} \tag{16}$$

where $e^{\mathrm{S,T}}$ represents the edge with v^{S} and v^{T} respectively as the source and target, $\mathrm{sub}(e^{\mathrm{S,T}})$ represents the set of the subsequent edges after $e^{\mathrm{S,T}}$, and $j_{e^{\mathrm{S,T}}}$ represents the ATSU where $e^{\mathrm{S,T}}$ is located.

The Approaching constraint can be expressed as:

$$d_{\min}(v^{\mathrm{T}}, v^{\mathrm{D}}) < d_{\min}(v^{\mathrm{S}}, v^{\mathrm{D}}) \tag{17}$$

where $d_{\min}(v, v')$ represent the straight-line distance between two vertices.

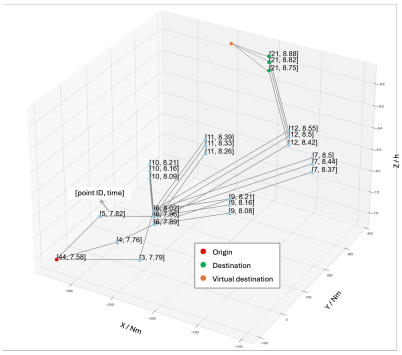


Figure 6. Generated spatio-temporal graph of a flight, where the label of vertexes is in the form of [point ID, time].

Figure 6 shows a flight's generated spatio-temporal graph for an example. If the generated directed graph contains flight i's destination, it indicates a feasible solution. In this case, the flight distance represented by the edges is used for edge weighting. Since there may be several destination vertices (representing different times) included in the graph, a virtual destination is created, which connects with each destination vertices with a weight of 0 so that general path search algorithms can be used on the graph (refer to subsection III-D). If there is no destination vertex in the generated directed graph, an adaptive postponement strategy is employed to expand the search space (refer to subsection III-E).

This graph generation method is highly customisable to accommodate different performance preferences. For example, some $p^{\mathrm{RM}}_{j,k}$ can be set to 0 to prevent flights from entering ATSUs that FMs and/or ATCers do not want them to enter. Additionally, $\Delta t(v_i, f_i)$ can be adjusted to fit the additional flight costs acceptable to airlines. Refer to subsection IV-B for the customisation of ground delay and rerouting strategies.

D. Optimal path search with admissible heuristic function

After obtaining the directed weighted graph (represented by G_i for flight i), we can theoretically apply any graph-based path search algorithm, including the uninformed and informed, such as Dijkstra [13] and A-star [14] algorithms, respectively. For informed algorithms like A-star and its variants, to ensure an optimal solution, an admissible heuristic function must be set. A heuristic function is said to be admissible if it never overestimates the cost of reaching the goal, i.e. the cost it

estimates to achieve the goal is not higher than the lowest possible cost from the current point in the path [15]. With a standard A-star algorithm as an example, the evaluation function consists of actual cost and estimated cost:

$$f^{\mathrm{E}}_{G_i}(v) = f^{\mathrm{A}}_{G_i}(v) + f^{\mathrm{H}}_{G_i}(v) \tag{18}$$

where $f^{\mathrm{E}}_{G_i}(v)$, $f^{\mathrm{A}}_{G_i}(v)$ and $f^{\mathrm{H}}_{G_i}(v)$ are the evaluation function of vertex v, actual cost from the origin to vertex v, and estimated cost from vertex v to the destination in weighted directed graph G_i, respectively. Since directed graphs are weighted by distance, $f^{\mathrm{A}}_{G_i}(v)$ is expressed by:

$$f^{\mathrm{A}}_{G_i}(v) = d^{G_i}_{\min}(v^{\mathrm{O}}, v) \tag{19}$$

where $d^{G_i}_{\min}(v^{\mathrm{O}}, v)$ represents the minimum flight distance from v^{O} to v in graph G_i. $f^{\mathrm{H}}_{G_i}(v)$ can be expressed by:

$$f^{\mathrm{H}}_{G_i}(v) = d^{G^{\mathrm{P}}}_{\min}(v, v^{\mathrm{D}}) \tag{20}$$

where $d^{G^{\mathrm{P}}}_{\min}(v, v^{\mathrm{D}})$ represents the minimum flight distance from v to v^{D} in the planar graph G^{P}.

E. FCFS-based sequential planning algorithm with adaptive postponement

For greater fairness and transparency for community application, the proposed DCB method employs an FCFS strategy. To ensure that the proposed method always yields feasible solutions, this study adopts an adaptive postponement iteration approach [16]. If a directed weighted graph containing the flight destination cannot be generated under the current planned departure time, the planned departure time is postponed by a time ΔT, and the attempt to generate a compliant directed weighted graph is retried. This process is repeated until a directed weighted graph containing the flight destination is obtained. The value of ΔT can be set based on actual requirements to balance optimality and computational speed. Algorithm 1 presents the pseudocode for the proposed DCB method with the FCFS-based sequential planning mechanism.

Algorithm 1 FCFS-based sequential planning algorithm with adaptive postponement

Input: Planned trajectories of flights $i \in I$, airspace structure (planar graph G^{P}), current remaining maximum occupancy probabilities $p^{\mathrm{RM}}_{j,k}, j \in J, k \in K$

for $i \in I$ in chronological order entering airspace **do**
  if flight i's planned trajectory is hotspot-free **then**
    Mark flight i's planned trajectory as hotspot-free
  else
    while flight i' has no hotspot-free trajectory **do**
      if A weighted directed graph including flight i's destination can be generated **then**
        Obtain flight i's hotspot-free trajectory by the optimal path search
      else
        Postpone flight i's departure time by ΔT
      end if
    end while
  end if
  Update remaining max. occupancy probabilities $p^{\mathrm{RM}}_{j,k}$
end for

Output: Hotspot-free trajectories of flights $i \in I$

IV. Simulation Experiments

The experimental environment for this study is Python 3.9, running on macOS with an Apple M1 Pro chip and 16GB of memory.

A. Scenario and parameter setup

The simulation experiments in this study use the validation scenarios from the HYPERSOLVER project, as shown in Figure 5. The nominal capacity of each ATSU is set according to its area. It should be noted that, since this study does not consider changes in flight levels, the capacity set is for a single flight level only. The time window duration is 20 minutes. To conduct pressure testing, 6 traffic density scenarios were established, with 1,500, 1,600, 1,700, 1,800, 1,900, and 2,000 flights per 12 hours, respectively. Each traffic density scenario is randomly generated for 100 instances, resulting in a total of 600 instances. Figure 7 shows examples of the traffic density for each scenario. Overload tolerance ζ is set as 5%.

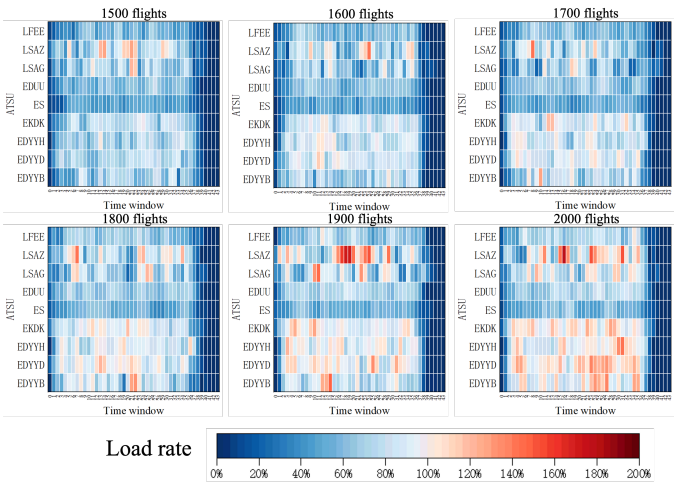


Figure 7. Load rates of experimental scenarios. This figure illustrates one of the 100 random scenarios for each traffic density as an example.

The random generation method for trajectories is as follows:

1) first, it randomly selects two Origin-Destination (OD) points (waypoints on the boundary of the airspace of interest) and generates the shortest path between them,
2) then, randomly generates the departure time (0-12 hours) and flight speed (400-500 knots), and
3) finally, calculates the time at each waypoint.

The model calculating approaching time uncertainty at each waypoint is referred to relevant studies [17].

B. Comparative model and performance indicator

To verify the key features of the proposed method (rerouting strategy, ground delay strategy, and consideration of uncertainty), four comparative models were designed (GRU, GU,

RU, and GR), while a conventional FCFS heuristic and an Integer Linear Programming (ILP) [11] are also used for reference. The features of these models are shown in Table I. Model GRU is this study's proposed method.

TABLE I. FEATURES OF COMPARATIVE MODELS

Model	Feature		
	Reroute	Ground delay	Uncertainty
GRU	✓	✓	✓
GU	-	✓	✓
RU	✓	-	✓
GR	✓	✓	-
FCFS	-	✓	-
ILP	-	✓	-

Table II shows the key performance metrics tested.

TABLE II. PERFORMANCE METRICS

Indicator	Formula	Indicator	Formula
Number of unsolved flights	n_{US}	Rate of unsolved flights	n_{US}/n
Number of changed flights	n_{C}	Rate of changed flights	$n_{\mathrm{C}}/(n-n_{\mathrm{US}})$
Total delay time	t_{D}	Average delay time for delayed flights	$t_{\mathrm{D}}/n_{\mathrm{C}}$
Total additional flight time	$t_{\mathrm{R}}^{\mathrm{AF}}$	Total flight time for rerouted flights (original)	$t_{\mathrm{R}}^{\mathrm{OTF}}$
Number of flights in the scenarios	n	Rate of additional flight time for rerouted flight	$t_{\mathrm{R}}^{\mathrm{AF}}/t_{\mathrm{R}}^{\mathrm{OTF}}$
Total computation time	t_{C}	Average computation time	t_{C}/n

C. Performance test

1) Effectiveness: Figure 8 shows the solvability, where the colour area expresses the standard errors (the same way is used in the following similar figures). Theoretically, as long as the ground delay can be used, the proposed method can always solve the DCB problem. For model RU, even in the highest traffic density scenarios, the rate of unsolved flights is only 2.49%, which indicates, to some extent, that the proposed rerouting method is significant in solving the DCB problem.

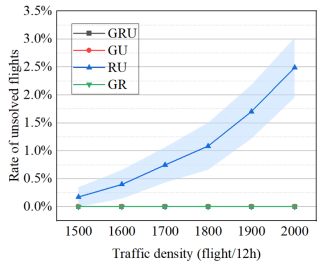
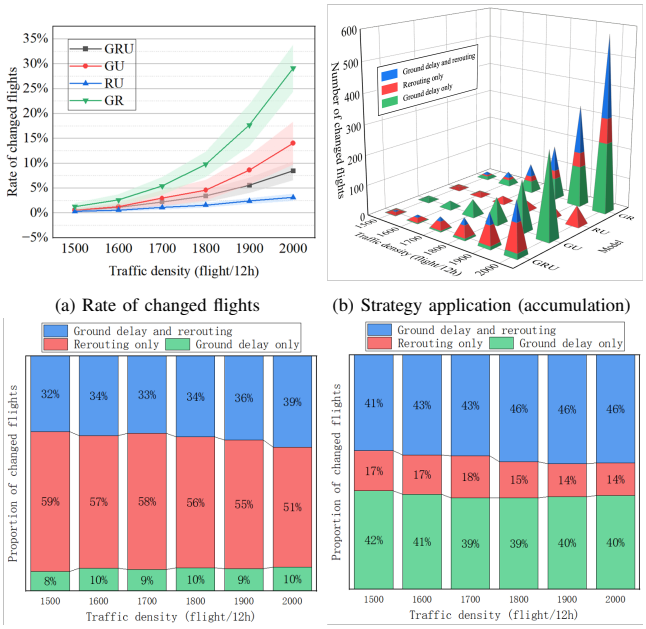


Figure 8. The proportion of cases where the solver fails to generate a hotspot-free trajectory.

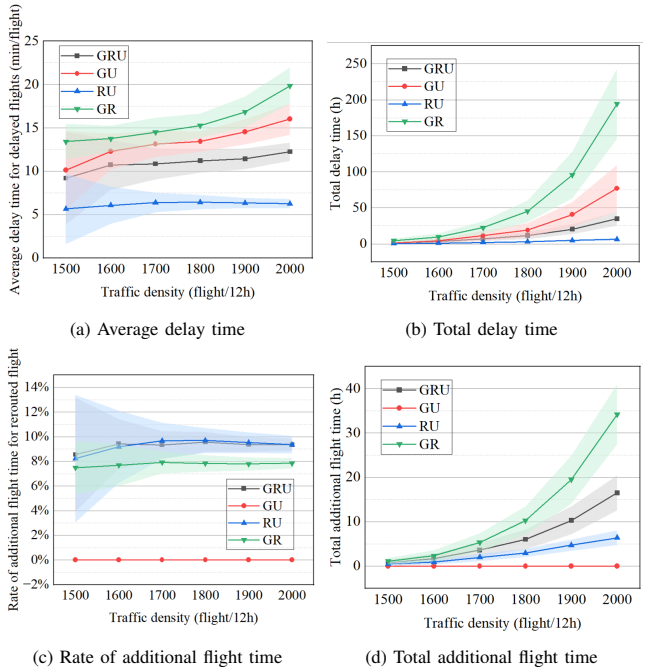
2) Efficiency: Figure 9 and Figure 10 show the strategy application of changed flights and additional cost, respectively. Model RU, which cannot fully solve the DCB problem, is not considered in the efficiency metrics. Among the other models, the proposed model (GRU) can solve the DCB problem by changing a smaller proportion of flights resulting in less delay time and additional flight time. This demonstrates that combining ground delay and rerouting strategies is more efficient than using either strategy alone. In the highest traffic density scenario, the rate of changed flights, the average delay time for delayed flights, and the rate of additional flight time for rerouted flights are only 8.46%, 12.2 minutes, and 9.34%, respectively. Additionally, in model GR, a greater proportion of flights has to use both ground delay and rerouting strategies, and both the delay time and additional flight time are longer, indicating that considering uncertainty can improve the efficiency of solving the DCB problem.



(a) Rate of changed flights (b) Strategy application (accumulation)

(c) GRU strategy application (proportion) (d) GR strategy application (proportion)

Figure 9. Strategy application of changed flights.



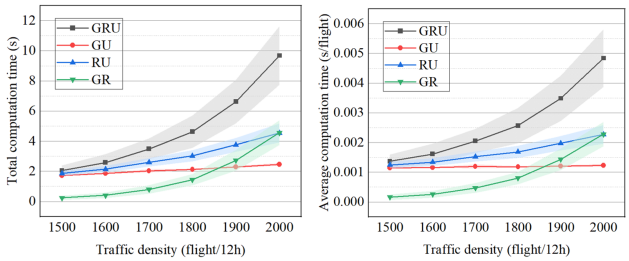
(a) Average delay time (b) Total delay time

(c) Rate of additional flight time (d) Total additional flight time

Figure 10. Hotspot-free trajectories' additional cost compared to planned ones.

3) Timeliness: Figure 11 shows the total and average computation time. Since the proposed method (GRU) contains all the features, its computation time is slightly higher than that of other methods. Nevertheless, the total computation time

of GRU is only 9.68 seconds in the highest traffic density scenario, which meets the performance needs of continuous and fast loop iterations. Moreover, it still has the potential to meet the demand for fast computation while further integrating additional functionalities, such as speed adjustments and rerouting within ATSU.



(a) Total computation time (b) Average computation time

Figure 11. Computation time including data input, modelling and solving.

D. Baseline comparison

The proposed method is compared with the customised FCFS heuristic and an ILP, as a reference. It should be noted that this comparison is not entirely fair, as both the FCFS heuristic and ILP methods can only use the ground delay strategy (the ILP method that considers rerouting cannot solve the large-scale DCB problem in this study within a limited time). Nevertheless, we provide the comparison results for the reader's reference. Table III shows the results of a 1500-flight scenario. The proposed method is significantly more efficient than the other two methods, demonstrating the importance of the rerouting strategy in DCB.

TABLE III. COMPARATIVE RESULTS IN A 1500-FLIGHT SCENARIO

	GRU	ILP	FCFS
Rate of delayed flights	**0.53%**	20.53%	65.07%
Total delay time (min)	**79**	1501	12002
Average delay time for delayed flights (min/flight)	9.88	4.87	12.30
Computation time (s)	2.06	249.6	0.05

V. CONCLUSION

This paper, through simulation experiments, demonstrates that the proposed spatio-temporal graph-based DCB method can quickly solve large-scale DCB problems considering uncertainty at an acceptable additional cost. The main findings of this study are as follows:

1) The ground delay is necessary for solving DCB problems, but rerouting is generally more efficient. Combining both strategies achieves the highest efficiency.
2) Considering uncertainty can improve the efficiency of solving DCB problems.
3) The proposed method shows significant performance advantages over the conventional FCFS heuristic and ground delay-based ILP methods.

In future research, we will further expand its compatibility to enhance its potential for practical application, such as allowing adjustments to flight speed and selecting several intermediate points within ATSUs to modify flight trajectories.

ACKNOWLEDGMENT

The authors want to acknowledge the HYPERSOLVER consortium for their valuable discussions during the project execution and especially for the Concept Description of HYPERSOLVER. HYPERSOLVER consortium is formed by NeoMetSys (NMS: Coordinator), L'Ecole Nationale d'Aviation Civile (ENAC), EUROCONTROL (ECTL), LUFTFARTSVERKET (LFV), the Nanyang Technological University (NTU), the University of Warwick (WAR). The project has received funding from the SESAR Joint Undertaking under Grant Agreement No 101114820 under the European Union's Horizon 2020 research and innovation program for the partners (NMS, ENAC, ECTL, LFV) and National funding for the Associated Partners (NTU and WAR). The authors also want to acknowledge Prof Lei Yang (Nanjing University of Aeronautics and Astronautics) and Prof Yan Xu (Beihang University) for contributing to the DCB method prototype.

REFERENCES

[1] C. Mannino, A. Nakkerud, et al., Air traffic flow management with layered workload constraints, Computers & operations research 127 (2021) 105159.
[2] Y. Chen, M. Hu, et al., Locally generalised multi-agent reinforcement learning for demand and capacity balancing with customised neural networks, Chinese Journal of Aeronautics 36 (4) (2023) 338–353.
[3] ICAO, Global Air Traffic Management Operational Concept, Montréal (2005).
[4] EUROCONTROL, European atm master plan - digitalising europe's aviation infrastructure, Tech. rep., EUROCONTROL (2020).
[5] ICAO, Manual on air traffic management system requirements (doc 9882), Tech. rep., ICAO (2008).
[6] Y. Liu, Y. Liu, et al., Using machine learning to analyze air traffic management actions: Ground delay program case study, Transportation Research Part E: Logistics and Transportation Review 131 (2019) 80–95.
[7] R. Dalmau, Predicting the likelihood of airspace user rerouting to mitigate air traffic flow management delay, Transportation Research Part C: Emerging Technologies 144 (2022) 103869.
[8] Y. Xu, X. Prats, et al., Synchronised demand-capacity balancing in collaborative air traffic flow management, Transportation Research Part C: Emerging Technologies 114 (2020) 359–376.
[9] M. Xiao, K. Cai, et al., Hybridized encoding for evolutionary multi-objective optimization of air traffic network flow: A case study on china, Transportation Research Part E: Logistics and Transportation Review 115 (2018) 35–55.
[10] EUROCONTROL, Atfcm operations manual - network manager, Tech. rep., EUROCONTROL (2021).
[11] Y. Chen, Y. Xu, et al., General multi-agent reinforcement learning integrating heuristic-based delay priority strategy for demand and capacity balancing, Transportation Research Part C: Emerging Technologies 153 (2023) 104218.
[12] L. Xin, Y. Kong, et al., Enable faster and smoother spatio-temporal trajectory planning for autonomous vehicles in constrained dynamic environment, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 235 (4) (2021) 1101–1112.
[13] E. W. Dijkstra, A note on two problems in connexion with graphs, in: Edsger Wybe Dijkstra: his life, work, and legacy, 2022, pp. 287–290.
[14] P. E. Hart, N. J. Nilsson, et al., A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics 4 (2) (1968) 100–107.
[15] P. H. H. Geffner, P. Haslum, Admissible heuristics for optimal planning, in: AI Planning Systems 2000, 2000, pp. 140–149.
[16] Y. Chen, M. Hu, et al., Autonomous planning of optimal four-dimensional trajectory for real-time en-route airspace operation with solution space visualisation, Transportation Research Part C: Emerging Technologies 140 (2022) 103701.
[17] R. A. Paielli, H. Erzberger, Conflict probability estimation for free flight, Journal of Guidance, Control, and Dynamics 20 (3) (1997) 588–596.