Incorporating Optimization in Strategic Conflict Resolution for UAS Traffic Management

Yiwen Tang, *Graduate Student Member, IEEE*, and Yan Xu, *Member, IEEE*

***Abstract*— This study presents an approach to incorporate optimisation in the strategic conflict resolution service for unmanned aircraft systems (UAS) traffic management. A conventional approach in line with the First-Come, First-Served (FCFS) principle is introduced, following the generation of two types of flight plans (i.e., *linear* and *area* operations) with uncertainty buffers further taken into account. This approach is based on iteratively checking the availability of the shared airspace volumes. Next, an optimisation model is formulated, using the same common airspace representation, aiming at minimising the overall delay and deviation to the equivalent FCFS solution (i.e., fairness concern), subject to operational constraints including a time-based separation minima. Some potential implementations are also envisioned for the optimisation model under plausible operational scenarios. Finally, simulation experiments are performed where five case studies are designed, including FCFS and optimisation, as well as their hybrid and batch uses depending on the flight plan submission time. Sensitivity analysis is conducted to assess the impact of some specific model assumptions. Results suggest that, compared to FCFS, a notable delay reduction can be achieved with optimisation incorporated, which is due to the FCFS prioritisation scheme that is often not efficient.**

***Index Terms*— Unmanned aircraft systems, unmanned traffic management, U-space, strategic conflict resolution, first-come, first-served (FCFS), optimization.**

I. Introduction

THE seamless integration of Unmanned Traffic Management (UTM) and Air Traffic Management (ATM) is critical to fully unlock the potential benefits of unmanned aircraft systems (UAS) applications and to contribute to the safety, efficiency, equity and reduced environmental impact of the aviation sector. The current approach for such integration relies on airspace segregation as a first step to ensure that UAS remain well clear of all other traffic and hazard [1]. To integrate UAS operations alongside manned aircraft in non-segregated airspace, it is critical to enhance UAS safety levels to match manned aviation and respond to the challenge from regulatory and technology [2], which in particular builds upon the separation assurance and collision avoidance capabilities.

In conventional ATM, a hierarchical conflict management approach is commonly used, which consists of three layers: strategic conflict management (through airspace organisation and management, demand and capacity balancing, and traffic synchronisation); separation provision; and collision avoidance [3]. These continual layers work as a whole to eventually ensure the safety to a designated level. Similarly, there are also three respective layers of conflict management defined in U-space (which is an enabling framework designated at the European level to support safe, efficient and secure access to airspace for large numbers of UAS). They include strategic (pre-tactical) deconfliction; tactical separation provision; and collision avoidance. Concretely, strategic conflict resolution service is referred to as the initiatives aiming to reduce the need for tactical deconfliction and collision avoidance capabilities. Such initiatives usually occur at the pre-flight phase, which typically involves conflict detection and then resolution. It is linked with the operation planning processing service. As described in [4] and [5], this involves operators sharing the flight plans with relevant parties and reducing any potential loss of separation by planning routes that are unlikely to cause interactions with other airspace users.

Early efforts toward the U-space implementation have demonstrated various initial solutions for the strategic conflict resolution service. For example, DOMUS project [6] explained the central place of operation plan processing within the service. It revealed the interconnection between U-space service providers (USSPs) in such a process. The EuroDrone project developed a strategic deconfliction tool that maps UAS missions and detects potential conflicts with other traffics. Conflicts will be subsequently resolved by means of proposing departure shifts within a pre-defined time interval [7].

Similar initiatives exist in the FAA/NASA UTM programme in the US, with an equivalent service named strategic deconfliction being developed. Initial demonstrations have applied a first-come, first-served (FCFS) approach, with operations required to resolve known conflicts prior to departure [8]. A set of relevant system-level requirements were presented in [9]. Results and analysis focusing on strategic deconfliction performance in UTM demonstration were reported in [10]. With a similar federated architecture applied, UK's Open-Access UTM framework has demonstrated the strategic deconfliction facilitated through the discovery and synchronisation (DSS) service which enables inter USSP communications [11].

Manuscript received 10 September 2022; revised 9 March 2023; accepted 9 June 2023. Date of publication 13 July 2023; date of current version 1 November 2023. This work was supported in part by the Single European Sky ATM Research 3 Joint Undertaking (SESAR JU) through the European Union's Horizon 2020 Research and Innovation Program: Air Mobility Urban-Large Experimental Demonstrations (AMU-LED) under Grant 101017702, in part by the Innovate U.K. through the U.K. Research and Innovation (UKRI) Future Flight Challenge Program: Air Mobility Ecosystem Consortium (AMEC) under Grant 10023201, and in part by the China Scholarship Council under Grant 202108060174. The Associate Editor for this article was J. Blum. *(Corresponding author: Yan Xu.)*

The authors are with the School of Aerospace, Transport and Manufacturing, Cranfield University, MK43 0AL Bedford, U.K. (e-mail: yiwen.tang@cranfield.ac.uk; yanxu@cranfield.ac.uk).

Digital Object Identifier 10.1109/TITS.2023.3290682

© 2023 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

In the past few years, an increasing body of research can be noticed aiming to develop robust, advanced and scalable approaches to support strategic conflict management. A set of autonomous algorithms were developed for flight plan deconfliction in [12], which proves to reduce collisions between UAS in high-traffic density scenarios. Other attempts can be found from airspace management perspective. An innovative lane-based approach was proposed and compared with more conventional volume-based strategic deconfliction, which uses one-way lanes and roundabouts at lane intersections to allow a much more efficient analysis and guarantee of separation safety [13]. An airspace reservation method based pre-flight deconfliction was studied in [14], where the process was considered as a multi-agent path finding problem. In addition, researchers further refined the problem using the decentralised pattern, whereby different USSPs will be able to solve conflicts independently [15]. The adaptive stress testing was applied to an autonomous strategic deconfliction method to seek the potential improvements to this critical UTM service [16]. Numerous simulations were performed to assess how well the requirements for strategic deconfliction developed by standards groups can support end-to-end safety [17].

In addition to efficiently resolving the conflict between UAS flight plans, the fairness concern embedded in this service has been also highlighted to ensure all the entrants should remain equitable in accessing the airspace [18]. Simulations explored how the FCFS scheme performs in terms of fairness to decentralised strategic deconfliction, and results suggest that introducing Required-Time-To-Act (RTTA) concept would be beneficial to constrain the flight plan file-ahead time [19]. Recent research which compared the trade-off between efficiency and fairness in UAS traffic flow management (UTFM) was studied in [20], where a UTFM framework incorporating fairness, user preferences, and dynamic trajectory requests was developed, and two fairness metrics including the number of reversals and time-order deviations were identified with comprehensive analysis from different perspectives.

Although some progresses have been made to develop and implement this service, it is still in its infancy. To be more specific, The commonly used airspace reservation concept limits the dynamic use of airspace, where once a group of airspace volumes are occupied, the access to other flights might be denied for the entire duration of the current flight, which leads to inefficient use of the airspace capacity. Furthermore, as a general consensus for this service, the FCFS principle serves as a fair yet simple way of processing flight plans. However, it lacks the consideration of overall efficiency from the system perspective. Thus, finding methods that balances efficiency and fairness is essential for scaling up to large-scale high-density operations in the future. Another potential weakness is the discussion of how the flight plans could be processed, that is whether flights are dealt with individually or in batch, under centralised or decentralised operational frameworks, which deserves some feasibility study to clearly reveal their effects.

This paper aims to address these research gaps from the following aspects:

- An optimisation model is proposed to be fully incorporated with the existing FCFS approach, where efficiency and fairness can be balanced when providing the strategic conflict resolution service. The integration can be realised in line with different flight plans submission patterns (individual and batch) to meet the operational requirements.
- A grid-based common airspace representation is presented to unify the airspace occupancy and enable the dynamic use of airspace capacity, which also forms the basis for implementing the above integrated approach under both centralised and decentralised settings.
- To fully reveal the effects of proposed approach, including the trade-off between efficiency and fairness, five implementation cases have been designed, as well as a detailed analysis of key impact factors such as optimisation proportion, batch size and separation minima.

The rest of the paper is organised as follows. The flight planning of two types of UAS operation is first introduced, and further expanded with uncertainty buffers to the flight plan. Leveraging the common representation of airspace volumes' spatial-temporal occupancy, the potential conflicts of flight plans are checked and resolved using the FCFS approach as a fairness baseline solution. Then, an optimisation model is presented to minimise the overall costs (including both delay costs and the costs associated with time-order deviation to FCFS solution) whilst resolving all the conflicts. Flight plan processing scenarios are introduced, which discuss the feature of both individual and batch handling. Finally, according to the different operational scenarios, several variants of the model are introduced, leading to five representative case studies. Numerical experiments are thus conducted in which both FCFS and optimisation models are examined in line with the defined case studies, followed by a sensitivity analysis to carefully assess the effects of the proposed approach in real-world applications.

II. First-Come, First-Served (FCFS) Approach

This section introduces a conventional strategic conflict resolution approach that follows the FCFS principle. We first briefly discuss how the flight plans are considered including their expanded uncertainty buffers, which are the key input to the proposed approach. The FCFS algorithm is then presented, based on a common airspace representation that synchronises the flight plan processing from different service providers.

A. Types of Operation

As introduced in our previous work [21], two types of commonly-seen UAS operations, namely *area* and *linear* operation, are considered in this paper. Intuitive examples is shown in Fig. 1. Specifically, for *area* operations, such as scanning and pre-programmed operations, the UAS may revisit certain airspace volumes multiple times, as indicated by the red grids in Fig. 1. In terms of *linear* operations (as indicated by the blue grids in Fig. 1), such as delivery and linear survey missions, the UAS trajectory may pass through each airspace volume only once.

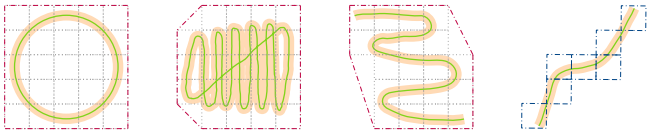


Fig. 1. Schematic of *area* and *linear* UAS operations (as marked in red and blue grids respectively), adapted from [4].

Following such characteristics, this paper further extends the previous definition of operations from 2-Dimensional to 3-Dimensional, adding the altitude layer to the associated trajectory/volume. With regard to the *linear* operation, its trajectory includes both the top of the climb and the top of the descent, accounting for the vertical take-off and landing (VTOL) phases as well as a constant cruise phase to mimic the UAS moving pattern in real operations.

For the *area* operation, its operational area is set with an upper bound of altitude, being the ground level as the lower bound. Specifically, a delta time is computed and attached to each airspace volume traversed along the *linear* trajectory, whereas all the concerned volumes for an *area* operation have the same timestamps which involve only the planned taking off and landing times.

B. Airspace Constraints

Geo-fence is a virtual 3-Dimensional perimeter around a geographic point used as airspace constraints to restrict access to drones for safety, security, privacy or environmental reasons [18]. As it can be predefined or dynamically generated, this measure could be divided into static (long-term) geo-fence, and dynamic (short-term) geo-fence [22]. To be specific, static geo-fence represents physical objects and permanent airspace partitions such as buildings and restricted airspace. Its information is usually provided before flight, which will be used when planning trajectory. On the other hand, the dynamic geo-fence, which is usually defined by the (dynamic) geo-fencing service [4], are geographic boundaries which should be updated during the UAS operations, such as local weather and emergencies. The geo-fences' information will be provided through the geo-awareness service and will have to be shared with the operators as early as possible. In this paper, both static and dynamic geo-fences will be considered in a similar way as in [21], where we consider the tactical geo-fences based on the operation's planned take-off time. Altitude bounds are further added to restrict UAS from entering a specific area within a certain altitude range.

C. Flight Plan Generation

To generate the flight plan for this study, a basic 3-Dimensional airspace model is first built. The airspace is discretised into a set of airspace volumes of the same dimension. For each flight, a volume is connected to its adjacent in 10 directions (8 horizontal and 2 for vertical) except for those crossing the airspace constraints or on the boundary. The central point of each airspace volume is used when generating the flight path. Fig. 2 shows some instances of the generated *linear* flights (blue lines) and *area* flights (green polygons),

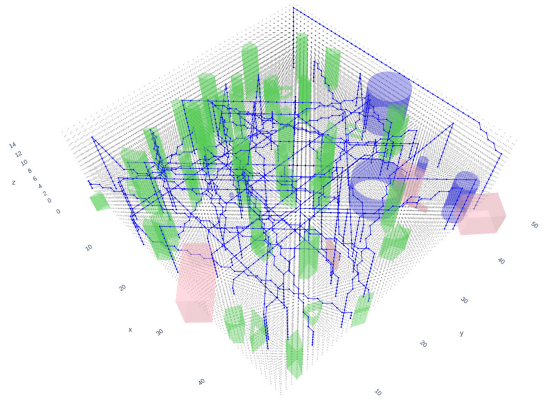


Fig. 2. Generated UAS trajectories in the 3-Dimensional airspace model, including *linear* (blue lines) and *area* flights (green polygons) avoiding airspace constraints (pink cuboids and blue cylinders).

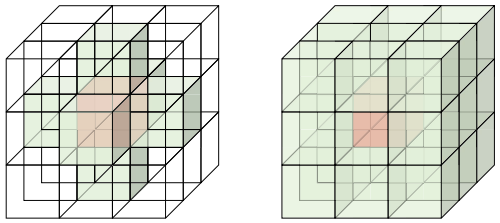
with also static geo-fences (cylinder) and dynamic geo-fences (cuboid), as well as the volume central points (black dots).

For the *linear* flight, the classic A* algorithm is applied to search for the shortest path (bypassing geo-fences). Specifically, recall that UAS follows a VTOL operation pattern, with vertical climbing and descending paths. Once reached the target altitude, UAS will enter the cruise phase, during which the A* algorithm is applied to search the shortest path from the adjacent volumes in 8 horizontal directions, using the UAS as the centre point. Although the A* algorithm applied can search the surrounding directions, for the purposes of this work, a simplified search in 10 directions is used, as path planning is not the primary focus of this study. On top of the generated path, timestamps are attached from the take-off time at its first point and iterated over each trajectory segment at a given speed until the last point for landing time.

Regarding the *area* flight, we consider a polygon which is composed of a random set of vertices to define the boundary of an operation, and append altitude bounds to define the vertical limits. Recall the feature of *area* operation in Sec. II-A, all the airspace volumes within it will be captured as part of the flight plan. Each edge of the operation boundary is similar to the segment of the *linear* trajectory which is subject to all the airspace constraints. The major difference lies in their time stamps. Considering that some airspace volumes might be revisited during the operation, we fix the same operational time for all the involved airspace volumes, rather than specifying unique time stamps for each grid. In short, its take-off and landing time which defines the operation's duration will be set the same across all the concerned volumes.

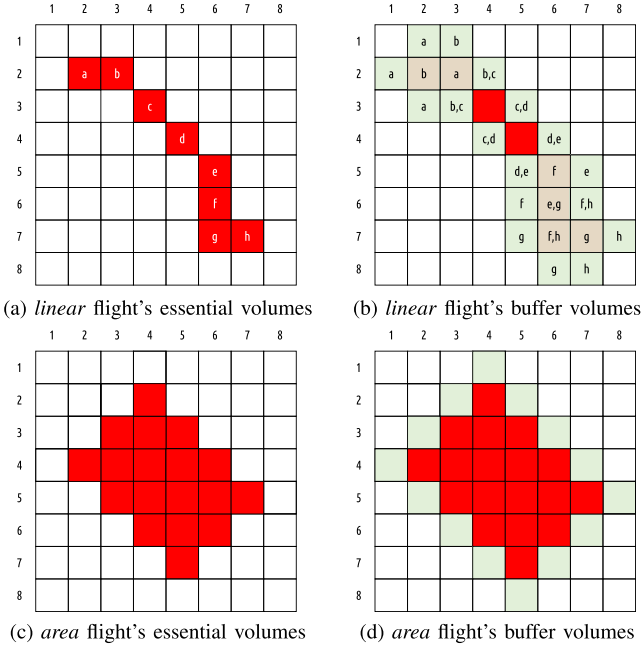
D. Expanded Uncertainty Buffers

The current UAS techniques may involve significant operational uncertainties that could lead to undesirable consequences, e.g., UAS experiencing loss of data link or failure of flight controller, or on the ground, the remote pilot committing a critical human error. As a result, once the nominal trajectory is generated, we further include some safety buffers as part of the flight plan to provide redundant protective space in response to potential uncertainties [23]. Inspired by [4]



(a) 6 volumes (b) 26 volumes

Fig. 3. Typical ways of identifying adjacent volumes as uncertainty buffers.



(a) *linear* flight's essential volumes (b) *linear* flight's buffer volumes

(c) *area* flight's essential volumes (d) *area* flight's buffer volumes

Fig. 4. Examples of relationship between essential and buffer volumes.

and [24], a group of buffer volumes are attached to each airspace volume that is planned to be traversed. Two typical ways of identifying those buffers can be found in the open literature, as shown in Fig. 3, with 6 most adjacent volumes in Fig. 3a and 26 all adjacent in Fig. 3b. Noted that specific uncertainty events are not considered in this paper, and all uncertainties are assumed to be mitigated by the pre-defined buffer space added to the flight plan.

Next is to synchronise the times of the original essential volumes and their associated buffer volumes. An example of the essential volumes in 2-Dimensional for *linear* and *area* flights can be seen in Fig. 4a and Fig. 4c respectively. A specific volume can be an essential volume and a buffer volume at the same time for a *linear* flight. The idea is to enforce that the timestamps of the buffer volumes remain the same as their corresponding essential volumes. Fig. 4b illustrates this situation using the buffer definition of Fig. 3a. For instance, at position (6,6), it is the essential volume f of the original trajectory. With buffers included, it is also the buffer volume corresponding to both essential volumes e and g, which means that three different timestamps are associated with this particular volume within the flight plan. The case for *area* flight is straightforward (see Fig. 4d), as all the volumes (both essential and buffer) are considered to have the same starting and ending times. Thus, the buffer volumes for *area* flights can be simply considered as a set of extended essential volumes.

Algorithm 1 FCFS Pseudo Algorithm

While $f \in F$ **do**
  If f is *linear* flight **then**
    for (e,t), $(b_e,t) \in f_l$ **do** {e/b: essential/buffer}
      If (e,t) or (b_e,t) is blocked **then**
        Delay f_l for 1 time unit
        break
      else
        Insert (e,t) and (b_e,t) to $list_l$
    for (e,t) and $(b_e,t) \in list_l$ **do**
      Occupy (e,t), $(e,t \pm sep)$
      Occupy (b_e,t), $(b_e,t \pm sep)$
  If f is *area* flight **then**
    for $(e/b_e,[t_{ini},t_{end}]) \in f_a$ **do**
      If $(e/b_e,[t_{ini},t_{end}])$ is blocked **then**
        Delay f_a for 1 time unit
        break
      else
        Insert $(e/b_e,[t_{ini},t_{end}])$ to $list_a$
    for $(e/b_e,[t_{ini},t_{end}]) \in list_a$
      Occupy $(e/b_e,[t_{ini} \pm sep, t_{end} \pm sep)$

E. Pseudo Algorithm

Given the above flight plans, we then use the First-Come, First-Served approach to perform strategic conflict resolution, which is widely used in both ATM [25] and UTM [26], [27] field to prevent conflicts between the new flight and the previous flights. Recall that the flight plans are processed according to their request/submission order, rather than their scheduled take-off time, so this approach may be also referred to as First-Submitted/Requested, First-Served. In such a case, the flight plan submitted early will have a high priority to be processed. The pseudo code of this approach is presented in Algorithm 1.

The fundamental of the algorithm is to have a piece of common airspace information to represent the occupancy status of the airspace volumes. Once a group of volumes have been occupied by early-submitted flight plans (with separation minima included), they will not be available for the other subsequently submitted flight plans until the former operation ended. Thus, the conflict flights need to be delayed in time steps to resolve the conflict. With the newly assigned take-off time, the delayed flight will be checked again with the airspace occupancy status and get further delayed if the conflict still exists. Repeat the process until all the concerned volumes are checked to be clear. As shown in Fig. 5, each flight will be checked by common airspace representation based on the order of submission. The graph gives an illustration about the common airspace representation. Contingency events may occur with some pop-up flight requests of higher priority, which may require re-planning the time slots following the updated order. However, this issue is beyond the discussion of this paper.

III. Optimisation Approach

This section introduces an optimisation model for strategic conflict resolution that can be fully incorporated with the FCFS

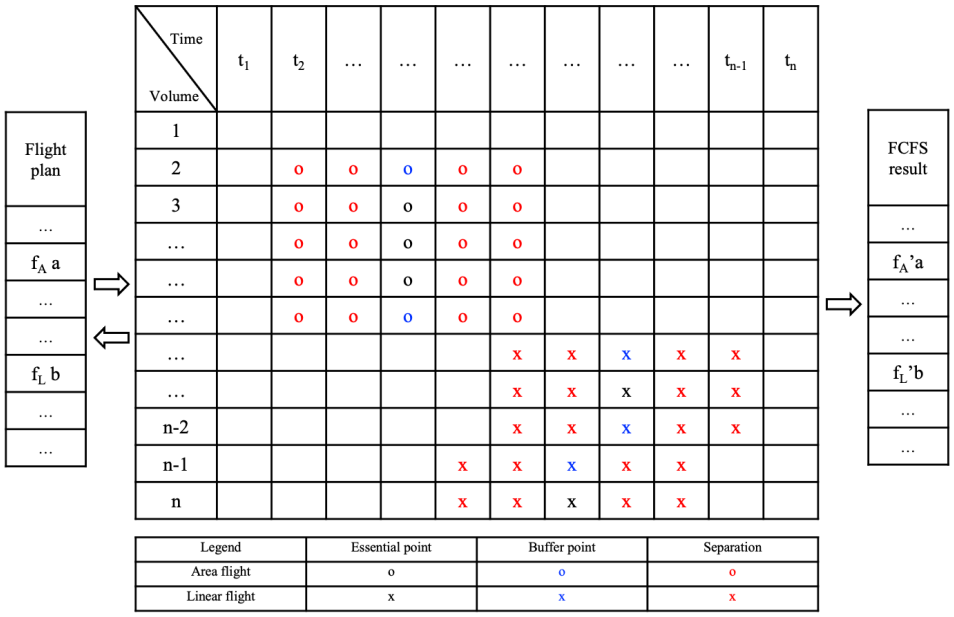


Fig. 5. Schematic of First-Come, First Served approach based on common airspace representation.

approach. Some emerging operational scenarios are envisioned to show the potential of implementing the proposed approach. The mathematical model is formulated, with a special focus on the constraints of time-based separation assurance.

A. Operational Scenarios

While the decentralised or federated architecture for UTM is widely accepted, there are differences in what services are to be centralised, with some functions probably targeting more on centralisation than others. For instance, as an early effort to extend the U-space ecosystem to accommodate Urban Air Mobility, AMU-LED (Air Mobility Urban - Large Experimental Demonstrations) project is expected to demonstrate various U-space services under both centralised and decentralised architectures [28].

As part of the ongoing research work within the AMU-LED project, this paper explores potential operational scenarios of the proposed approach. The detailed architecture, along with specific actors, responsibilities and data flow among them, is beyond the scope of this paper. The reader is directed to the AMU-LED Technical Description Report [29] and the forthcoming deliverables for detailed operational models. In this paper, we will focus on how the flight plans are processed through this service, as illustrated in Fig. 6.

Fig. 6a represents how the flight plan is processed one-by-one following the FCFS rule. Intuitively, each USSP "pulls" the information of airspace occupancy status from the Common Information Service (CIS) provider, and checks if any delay is needed for the particular flight. Once the (potentially delayed) flight plan does not incur any conflicts, it can be "pushed" back through CIS to update the common airspace representation. This can be done in parallel where as many USSPs as possible can be involved in handling their respective flight plans. While this might be an oversimplified description, it could provide us with some general insights into the process. We note that in the real world more complex negotiation schemes and data exchange protocols are being developed and tested.

It can be foreseen that having the flight plans processed one by one would be usually less efficient than processing them as a whole, from the resource allocation point of view (where the airspace occupancy in time and space are the resources). As a result, the proposed optimisation model in this paper aims to deal with a batch of flight plans and thus produces the optimal delay allocation amongst those flights. The optimisation model is also based on the concept of common airspace representation, and thus is fully interoperable with the previous FCFS approach. In addition, there could be different ways to form such batches flexibly, for instance, by periodically handling the submitted flight plans every time when a resolution decision needs to be made, as shown in Figs. 6b and 6c. Specifically, the former takes a decentralised implementation in the same way as FCFS, whereas the latter assumes centralisation where flight plans from various USSPs are processed all together.

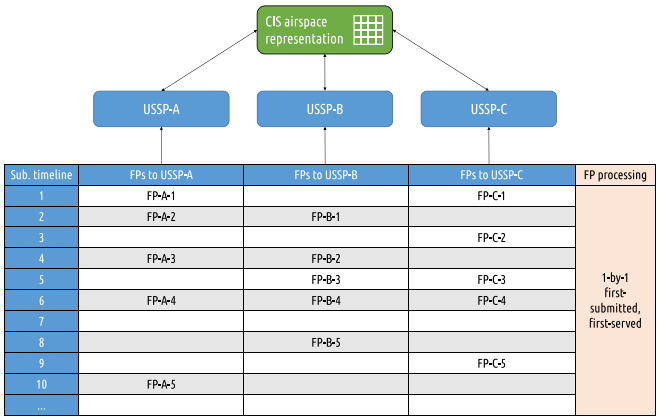
B. Model Formulation

Given the operational scenarios introduced previously about processing flight plans, in this section we present the mathematical model for incorporating optimisation in strategic conflict resolution.

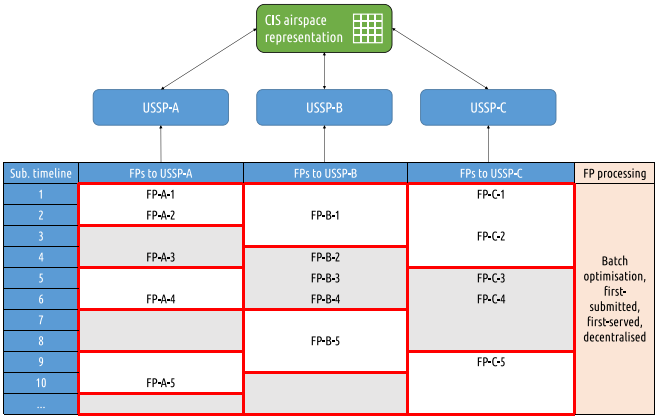
1) Decision Variables: The model is formulated with mixed-integer linear programming and the corresponding decision variables are defined as follows:

$$x_{l,t}^{j} = \begin{cases} 1, & \text{if } \textit{linear} \text{ flight } l \text{ arrives at volume } j \\ & \text{by time } t \\ 0, & \text{otherwise} \end{cases}$$

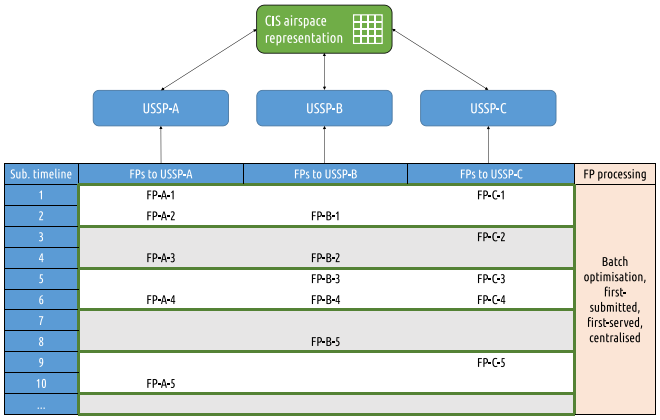
$$m_{a,t}^{j} = \begin{cases} 1, & \text{if } \textit{area} \text{ flight } a \text{ enters in volume } j \\ & \text{by time } t \\ 0, & \text{otherwise} \end{cases}$$



(a) First-Come, First-Served



(b) Decentralised batch optimisation



(c) Centralised batch optimisation

Fig. 6. Potential UTM operational scenarios for processing flight plans through the strategic conflict resolution service.

It should be noted that the "by" time is used, rather than "at" in the decision variables, which would enable a faster solution searching time as introduced in [30], while the "at" time can be derived from $(x^j_{l,t} - x^j_{l,t-1})$. Therefore, the time that a UAS operation arrives at or leaves from a certain airspace volume could be represented, which lays the foundation of the optimisation model. Additionally, if the entrance time for an *area* flight has been determined, the exit time can be known, as the flight duration d_a is provided given the submitted flight plan.

2) Objective Function: We consider an objective function composed of two types of costs including delay and fairness, specifically the overall delay and the time deviation for all *linear* and *area* flights with respect to their FCFS solution.

The total delay costs can be computed by Eq. (1):

$$C_{delay} = \sum_{l\in L}\sum_{j\in J_l^{(1)}}\sum_{t\in T_l^{J_l^{(1)}}}(t - r_l^{J_l^{(1)}})(x^j_{l,t} - x^j_{l,t-1}) + \sum_{a\in A}\sum_{j\in J_a^{(1)}}\sum_{t\in T_a^{J_a^{(1)}}}(t - r_a^{J_a^{(1)}})(m^j_{a,t} - m^j_{a,t-1}) \quad (1)$$

where T^j_l, T^j_a are specific-defined subsets of time moments feasible for delay assignment. The initially scheduled take-off times are depicted by $r_l^{J_l^{(1)}}, r_a^{J_a^{(1)}}$. In the optimisation approach, each flight is assumed to be of the same priority, but the model will allow flight prioritisation by specifying their weighted costs of delay.

The fairness cost is quantified by the time deviation. In this paper, a fair solution is one where the time slot allocation is decided by the flight plan submission sequence. Recall from the FCFS approach that the flight plans are processed according to their submission order. Based on that, we set the FCFS result as the fairness baseline for the time deviation calculation and incorporate it into the objective function of the optimisation approach. In other words, for each flight, the smaller the gap between the time assigned by the optimisation and the FCFS algorithm, the fairer the solution will be regarded. The time deviation metric for *linear* and *area* flights can be computed by Eq. (2):

$$C_{fair} = \sum_{l\in L}\sum_{j\in J_l^{(1)}}\sum_{t\in T_l^{J_l^{(1)}}}|t - s^{J_l^{(1)}}_{l,FCFS}|(x^j_{l,t} - x^j_{l,t-1}) + \sum_{a\in A}\sum_{j\in J_a^{(1)}}\sum_{t\in T_a^{J_a^{(1)}}}|t - s^{J_a^{(1)}}_{a,FCFS}|(m^j_{a,t} - m^j_{a,t-1}) \quad (2)$$

where the baseline time for *linear* and *area* flights are depicted by $s^{J_l^{(1)}}_{l,FCFS}$ and $s^{J_a^{(1)}}_{a,FCFS}$. The time deviation is defined by the

absolute value of the gap between the controlled time and the baseline time.

The two types of costs are combined as a whole, with weighting cost λ set for fairness considered in the model. The equations are reorganised as follows:

$$\min(C_{delay} + \lambda C_{fair}) \tag{3}$$

The above equation shows that adjusting the fairness factor will impact the proportion of fairness concern, which in turn affects the outcome of the delay allocation. Incorporating fairness into the model allows for the consideration of flight submission sequences during the process.

3) Flight Operations Constraints: The constraints are listed below, which can be grouped into flight operations, separation minima, and the binary condition of the decision variables.

The following constraints are associated with the operational limits with regard to each individual *linear* flight.

$$x^j_{l,\underline{T}^j_l-1} = 0,\ x^j_{l,\bar{T}^j_l} = 1 \quad \forall l \in L, \forall j \in J_l \tag{4}$$

$$x^j_{l,t} - x^j_{l,t-1} \geqslant 0, \quad \forall l \in L, \forall j \in J_l, \forall t \in T^j_l \tag{5}$$

$$x^{j'}_{l,t+\hat{t}^{jj'}_l} - x^j_{l,t} = 0, \quad \forall l \in L, \forall t \in T^j_l,\ j = J^{(i)}_l,$$
$$j' = J^{(i+1)}_l : \forall i \in [1, n_l) \tag{6}$$

Constraint (4) states that each *linear* flight should arrive at volume j within a time window, where the upper and lower bound are described by $\bar{T}^j_l$ and $\underline{T}^j_l$ respectively. Constraint (5) enforces the continuity of the timeline by specifying the relationship of the flight arrival status between a certain time and its previous time unit. Constraint (6) stipulates that the controlled time between any segment (j, j') of a *linear* flight remain unchanged than initially scheduled.

Similar to *linear* flight, *area* flight is limited by basic operational constraints, in the meantime, some other constraints are added according to its different characteristics.

$$m^j_{a,\underline{T}^j_a-1} = 0,\ m^j_{a,\bar{T}^j_a} = 1, \quad \forall a \in A, \forall j \in J_a \tag{7}$$

$$m^j_{a,t} - m^j_{a,t-1} \geqslant 0, \quad \forall a \in A, \forall j \in J_a, \forall t \in T^j_a \tag{8}$$

$$m^{j'}_{a,t} - m^j_{a,t} = 0, \quad \forall a \in A, \forall t \in [\underline{T}^j_a, \bar{T}^j_a],\ j = J^{(i)}_a,$$
$$j' = J^{(i+1)}_a : \forall i \in [1, n_a) \tag{9}$$

$$n^j_{a,t} = m^j_{a,t-d_a}, \quad \forall a \in A, \forall j \in J_a, \forall t \in [\underline{T}^j_a + d_a, \bar{T}^j_a + d_a] \tag{10}$$

$$m^j_{a,t} = 1, \quad \forall a \in A, \forall j \in J_a, \forall t \in (\bar{T}^j_a, \bar{T}^j_a + d_a] \tag{11}$$

$$n^j_{a,t} = 0, \quad \forall a \in A, \forall j \in J_a, \forall t \in [\underline{T}^j_a, \underline{T}^j_a - 1 + d_a] \tag{12}$$

$$z^j_{a,t} = m^j_{a,t} - n^j_{a,t}, \quad \forall a \in A, \forall j \in J_a, \forall t \in [\underline{T}^j_a, \bar{T}^j_a + d_a] \tag{13}$$

Constraint (7) - (9) specifies the similar operational constraints as *linear* flight, where the only difference is that Constraint (9) stipulates the same controlled times at all volumes covered by the flight. Constraint (10) states that the duration of any *area* flight will not change from the initial flight plan. Specifically, as the flight duration d_a is fixed, the lower and upper bound of a feasible time window at the "exit" volume can be expressed by $\underline{T}^j_a + d_a$ and $\overline{T}^j_a + d_a$, where $[\underline{T}^j_a, \overline{T}^j_a]$ are the feasible time window defined at the "entrance" volume. Constraint (11) and (12) are linked with Constraint (13), where they enforce the arrival status for *area* flight during the prescribed time duration. Constraint (13) states the arrival status $z^j_{a,t}$ for *area* flight at volume j at time t.

4) Separation Minima Constraints: The overall idea of the separation constraint is that, at each (valid) airspace volume, it can be occupied by only one flight within a sliding time window (length of separation minima), which is realised by the following constraint:

$$\sum_{l \in L^j} c^j_{l,k} + \sum_{a \in A^j} c^j_{a,k} \leqslant 1 \tag{14}$$

where $c^j_{l,k}$ and $c^j_{a,k}$ are the occupancy status of *linear* flight and *area* flight at volume j respectively. k is the sliding time window, where its moving step is the unit time step while the look-ahead horizon equals to the separation minima.

To have a clear view, the principle of applying the separation assurance constraint is as shown in Fig. 7, with a schematic of airspace occupancy status during a certain period of time at volume j. It should be noted that the *linear* flights and the *area* flights have different ways of occupying the airspace volumes. Fig. 7 shows the occupancy by *linear* flights' essential point on the left side, and the occupancy by the buffer point in the middle. On the right side of the figure is the occupancy by *area* flights.

Specifically, for *linear* flights, initially, if an essential point occupies volume j at time t, we count one flight occupying that volume straightforwardly. However, the same volume might be occupied at another time by a buffer point (belonging to another essential point) of the same flight as well (recall Sec. II-D). As both types of occupancy are coming from the same flight, they should be considered as a single occupancy specified by Constraint (14) which counts the total number of flights occupying j. Likewise, for *area* flights, since occupancy status remains the same and unchanged at anytime during a flight (refer to Fig. 4d), the essential point and the buffer point will be regarded as the same. As such, no matter how many slots are occupied from one flight, it will be counted as a single occupancy as well. Recall Fig. 7: within the time window $t1 - t3$, airspace volume j is occupied by 4 flights, namely *linear* flight $l1$ and $l2$, as well as *area* flight $a1$ and $a2$. This means that the separation constraint is not met. Within the time window $t6 - t8$, airspace volume j is only occupied by 1 flight (*linear* flight $l3$), which satisfies the separation constraint.

Considering the above different ways of checking the arrival status of the two operations, Constraint (14) needs to be considered separately. For *linear* flight, the occupancy status at a certain volume is determined by the difference between the controlled times of arrival. As shown in Eq. (15), it states that if more than or equal to one time slot is occupied by a *linear* flight within a time window at volume j (recall that there might be several timestamps associated with one volume for *linear* flights due to the buffer volumes), this volume will be

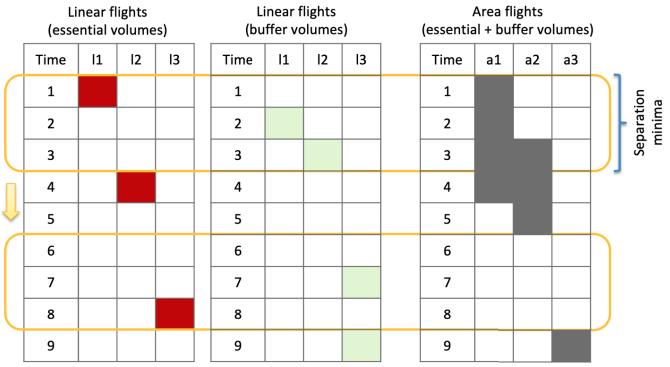


Fig. 7. Principle of applying the separation assurance constraint.

regarded as being occupied, then $c^j_{l,k} = 1$, otherwise $c^j_{l,k} = 0$.

$$c^j_{l,k} = \begin{cases} 1, & \sum_{t}^{t+s} x^j_{l,t} - x^j_{l,t-1} \geqslant 1 \\ 0, & \sum_{t}^{t+s} x^j_{l,t} - x^j_{l,t-1} < 1 \end{cases} \tag{15}$$

Worth noting that Eq. (15) is a piecewise function and should be converted into linear formula, a group of auxiliary variables and constraints are added:

$$\alpha^j_{l,k} = \sum_{j \in J^l} \sum_{t \in [k,k+s) \cap [\underline{T}^j_l, \bar{T}^j_l]} x^j_{l,t} - x^j_{l,t-1}, \quad \forall l \in L, \forall j \in J_l \tag{16}$$

$$c^j_{l,k} = \lambda 1^j_{l,k} + \lambda 2^j_{l,k}, \quad \forall l \in L, \forall j \in J_l \tag{17}$$

$$\beta 1^j_{l,k} + \beta 2^j_{l,k} = 1, \quad \forall l \in L, \forall j \in J_l \tag{18}$$

$$\alpha^j_{l,k} = \lambda 1^j_{l,k} + M \lambda 2^j_{l,k}, \quad \forall l \in L, \forall j \in J_l \tag{19}$$

$$\lambda 0^j_{l,k} + \lambda 1^j_{l,k} + \lambda 2^j_{l,k} = 1, \quad \forall l \in L, \forall j \in J_l \tag{20}$$

$$\beta 1^j_{l,k} \leqslant \lambda 0^j_{l,k} + \lambda 1^j_{l,k}, \quad \forall l \in L, \forall j \in J_l \tag{21}$$

$$\beta 2^j_{l,k} \leqslant \lambda 1^j_{l,k} + \lambda 2^j_{l,k}, \quad \forall l \in L, \forall j \in J_l \tag{22}$$

where $\lambda 0^j_{l,k}, \lambda 1^j_{l,k}, \lambda 2^j_{l,k}, \alpha^j_{l,k} \in \mathbb{R}^+$, $\beta 1^j_{l,k}, \beta 2^j_{l,k}, c^j_{l,k} \in \{0, 1\}$. J_l represents the collection of volumes including both essential volume and its related buffer volumes, which is because airspace can be regarded as occupied by either essential or buffer volume. Constraints (16) - (22) convert the piecewise function to a group of linear formulas and calculates the number of *linear* flights arriving at volume j, then the arrival status $c^j_{l,k}$ is determined.

Similarly, the arrival status for *area* flight can be calculated by Constraint (23). The sum of arrival status $\sum_t^{t+s} z^t_{a,j}$ determines whether the volume is occupied. In other words, if one or more than one time slot at volume j is occupied, the volume will be regarded as being occupied, then $c^j_{a,k} = 1$, otherwise, $c^j_{a,k} = 0$.

$$c^j_{a,k} = \begin{cases} 1, & \sum_{t}^{t+s} z^t_{a,j} \geqslant 1 \\ 0, & \sum_{t}^{t+s} z^t_{a,j} < 1 \end{cases} \tag{23}$$

We then convert the piecewise function Eq. (23) into linear formula by adding a group of auxiliary variables and constraints:

$$\alpha^j_{a,k} = \sum_{t \in [k,k+s) \cap [\underline{T}^j_a, \bar{T}^j_a + d_a]} z^j_{a,t}, \quad \forall a \in A, \forall j \in J_a \tag{24}$$

$$c^j_{a,k} = \lambda 1^j_{a,k} + \lambda 2^j_{a,k}, \quad \forall a \in A, \forall j \in J_a \tag{25}$$

$$\beta 1^j_{a,k} + \beta 2^j_{a,k} = 1, \quad \forall a \in A, \forall j \in J_a \tag{26}$$

$$\alpha^j_{a,k} = \lambda 1^j_{a,k} + s \lambda 2^j_{a,k}, \quad \forall a \in A, \forall j \in J_a \tag{27}$$

$$\lambda 0^j_{a,k} + \lambda 1^j_{a,k} + \lambda 2^j_{a,k} = 1, \quad \forall a \in A, \forall j \in J_a \tag{28}$$

$$\beta 1^j_{a,k} \leqslant \lambda 0^j_{a,k} + \lambda 1^j_{a,k}, \quad \forall a \in A, \forall j \in J_a \tag{29}$$

$$\beta 2^j_{a,k} \leqslant \lambda 1^j_{a,k} + \lambda 2^j_{a,k}, \quad \forall a \in A, \forall j \in J_a \tag{30}$$

where $\lambda 0^j_{a,k}, \lambda 1^j_{a,k}, \lambda 2^j_{a,k}, \alpha^j_{a,k} \in \mathbb{R}^+$, $\beta 1^j_{a,k}, \beta 2^j_{a,k}, c^j_{a,k} \in \{0, 1\}$. Constraints (24) - (30) convert the piecewise function to linear formulas and calculate the number of occupied time slot at volume j, then the value of arrival status $c^j_{a,k}$ can be determined.

5) Decision Variable Conditions:

$$x^j_{l,t} \in \{0, 1\}, \quad \forall l \in L, \forall j \in J_l, \forall t \in T^j_l \tag{31}$$

$$m^j_{a,t} \in \{0, 1\}, \quad \forall a \in A, \forall j \in J_a, \forall t \in T^j_a \tag{32}$$

$$n^j_{a,t} \in \{0, 1\}, \quad \forall a \in A, \forall j \in J_a, \forall t \in T^j_a \tag{33}$$

$$z^j_{a,t} \in \{0, 1\}, \quad \forall a \in A, \forall j \in J_a, \forall t \in [\underline{T}^j_a, \bar{T}^j_a + d_a] \tag{34}$$

Finally, Constraints (31) - (34) state the binary constraints and domains of the primary decision variables used in the model. In addition, $n^j_{a,t}$ and $z^j_{a,t}$ are auxiliary variables associated with *area* flights.

IV. Experiments and Discussions

This section presents numerical experiments with an illustrative scenario. We look into airspace that covers a 3-Dimensional space of 2.5*2.5 km^2 with a maximum altitude of 150 m (corresponds to the upper limit of Very Low-Level (VLL) airspace [31]), divided into 50*50*5 identical airspace volumes, leading to each volume of 50*50*30 m^3 size. Five case studies have been conducted, applying both FCFS and optimisation approaches, as well as their combined variants.

A. Experimental Setup

The traffic sample includes 200 flights (100 *linear* and 100 *area*) through the 24 hours' overall duration. There are 30 airspace constraints (15 static and 15 dynamic geo-fences) across this airspace. Trajectory buffers are considered for each *linear* flight in a way that each essential volume is associated with 6 surrounding buffer volumes. The generated flight plans and airspace constraints are as shown in Fig. 8.

Some experimental assumptions have been made: (1) the unit time step is set as 5 min; (2) the speed of *linear* operations are randomly set between 10-15 m/s; (3) the duration of *area* operations are randomly set between 5-30 minutes; and (4) a time-based separation minima is assumed to be 15 min at every airspace volume.

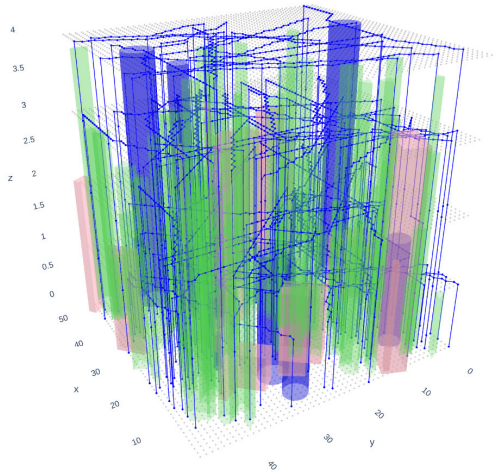


Fig. 8. An illustrative example of flight plans and airspace constraints of this study (Blue lines and green polygons are *linear* flight and *area* flight respectively).

In the experiments, Python has been used to develop the FCFS algorithm. For the optimisation approach, GAMS software suite has been used as the modelling tool and Gurobi v.7.5 optimiser as the solver. The experiments have been run on a 64 bit Intel®Core™i5-9500 CPU @ 3.00GHz 6 Cores computer with 16 GB of RAM and Linux OS.

B. Case Studies

The generated 200 flight plans are first sorted based on their submission order (which is defined arbitrarily). Depending on whether the FCFS/optimisation approach is selected to tackle a particular group of flights, five different case studies are considered to gather a set of comparable solutions:

- **Case-1**: Full FCFS
- **Case-2**: Full optimisation
- **Case-3**: Full optimisation (with fairness baseline)
- **Case-4**: Hybrid (optimisation followed by FCFS)
- **Case-5**: Batch optimisation

Concretely, flight plans are solely processed by FCFS and optimisation approach in Case-1 and Case-2 respectively. They are executed only once with the assumption that all the flight plans are available at the execution time. We note that this may not be the case in the real world, where some on-demand flight plans may not be available at an early stage. These two cases are set to evaluate the effectiveness of the two approaches, which can be also used to reveal the upper and lower bounds across all the case studies. The fairness factor (i.e., deviation to the FCFS solution) is considered in Case-3 to reveal how the fairness concern would impact the optimisation results, where $\lambda = 0.3$ is set as default.

Case-4 adopts a hybrid manner where we concatenate two approaches by applying optimisation to the first half of the flight plans and then using FCFS for the rest taking into account the previously occupied airspace volumes. The rationale of Case-4 is that some operations may be scheduled well in advance, thus grouped and handled as a whole by the optimisation approach, whereas other operations may be submitted shortly before taking off, thus individually processed by the FCFS approach. In Case-5, the flight plans are divided into five smaller batches which are resolved one group by the other, using the optimisation approach, in such a way to mimic the periodic processing of flight plans in the real world. In both cases, we aim to explore the flexible application and combination of these two methods to minimise delays while ensuring fairness.

C. Result Comparison

After assessing the entire set of the submitted flight plans, we have identified that there are 57 *linear* and 22 *area* operations in conflict in the initial experiment if without any resolution. The total number of conflicts is 655 (out of which there are 513 and 142 for *linear* and *area* flights respectively). All the conflicts can be effectively resolved by either the FCFS or the optimisation approach. An overall comparison of the solutions derived from the above five case studies can be found in Table I, summarised in a few key indicators.

We can observe from the table that Case-1 requires the highest amount of delay (890 min + 325 min), whereas the delay in Case-2 (550 min + 185 min) reaches the lowest. Although the number of affected flights is high (48 flights delayed in total), the total delay in Case-2 is almost only half of the delay that is needed in Case-1. The maximum delay in Case-2 also proves the efficiency of using the optimisation approach to minimise the delay. The total delay in Case-3 is slightly higher than that in Case-2, while the number of the affected flights drops to 43 (the least among all the cases). It can be noticed that adding the fairness factor into account leads to a small increase in total delay and also affects the number of delayed flights. The effects of the fairness factor will be further discussed in Sec. IV-D.

In Case-4, it can be seen that this case combines the benefits of the two approaches, reducing delay and improving the flexibility of flight plan processing. Case-5 applies the optimisation approach with a set of flight plans in batches (i.e., around 40 flights). Obviously, if the batch size reduces to only one flight, then the solution should be the same as with Case-1. Thus, the results suggest that potential benefits (e.g., 9% delay reduction) can be still achieved even by coupling a small number of flights for optimisation, which tends to be also easy for implementation in reality.

In addition, two representative metrics, namely the time-order deviation and the number of deviated flights, are calculated in each case study to reflect the fairness level. Case-1 serves as the baseline, and the respective deviation values from Case-2 to Case-5 show a decreasing trend as the weight of the FCFS approach increases in these cases. Specifically, as the entire set of flights is divided into small groups in Case-5, the extent to which flights (within each group) can be optimised is quite limited. The flights are processed almost in the same order as they are submitted, thus the deviation to the FCFS solution is the lowest.

D. Sensitivity Analysis

To explore how some of the key concerns, including the fairness factor, optimisation proportion, batch size and separation

TABLE I
RESULT COMPARISONS ACROSS FIVE CASES OF THE STUDY

KPI	**Case-1**	**Case-2**	**Case-3**	**Case-4**	**Case-5**
Delay (min) - *linear*	890	550	605	650	880
Delayed flights (a/c) - *linear*	35	37	33	32	35
Max delay (min) - *linear*	80	55	50	80	80
Delay (min) - *area*	325	185	260	350	270
Delayed flights (a/c) - *area*	11	11	10	13	9
Max delay (min) - *area*	75	45	45	75	75
Deviated time to FCFS (min)	0	1,310	820	725	185
Deviated flights to FCFS (a/c)	0	59	39	32	9
Total delay (min)	1,215	735	865	1,010	1,150
Total delayed flights (a/c)	46	48	43	45	44
Total solution time (sec)	15	1,320	1,440	310	180

TABLE II
EXPERIMENTAL SETTINGS IN SENSITIVITY ANALYSIS

Run No.	Case study	Sensitivity analysis		
		Fairness factor	Parameter	
1	Hybrid implementation	$\lambda = 0$	Optimisation proportion (%)	10%
2				20%
3				30%
...		$\lambda = 0.3$		40%
				50%
				60%
25				70%
26		$\lambda = 0.5$		80%
27				90%
28	Batch optimisation	$\lambda = 0$	Batch size (min)	120
29				144
...		$\lambda = 0.3$		240
				360
44		$\lambda = 0.5$		480
45				720
46	Full optimisation	$\lambda = 0.05$	-	
47		$\lambda = 0.2$		
48		$\lambda = 0.3$		
49		$\lambda = 0.4$		
50		$\lambda = 0.5$		
51		$\lambda = 0.6$		
52		$\lambda = 0.7$		
53		$\lambda = 0.8$		
54		$\lambda = 0.9$		
55	All	-	Separation minima (min)	5
56				10
57				20
58				25

minima, would affect the effects of applying the approaches in the above case studies, we have performed several groups of sensitivity experiments, where the detailed settings can be found in Table II.

1) Fairness Factor: Recall from Section III that the total delay cost of the optimisation approach contains parameter λ which depicts the weight of the time deviation with regard to the baseline solution derived from the FCFS approach. As λ increases, the deconfliction effects tend to be closer to the FCFS solution, as the time assigned by the FCFS is preferred over the optimisation. This leads to an increase in the total amount of delay because the flight plans processed by the FCFS only depend on the submission order instead of minimising the overall delay.

Fig. 9 shows the variation of delay for different values of the fairness factor (λ) specifically in Case-3. It is clear to see that the total delay increases as the fairness factor becomes higher. When λ equals 1 or 0.05, it yields almost the same results as with Case-1 or Case-2. Worth noting that, the trade-off between efficiency (minimising delay) and fairness (with respect to the first-come, first-served principle) is not linear. When λ is below a certain value (e.g., $\lambda = 0.5$), it will notably affect the efficiency, whereas when λ is greater than 0.8, the efficiency only increases slightly.

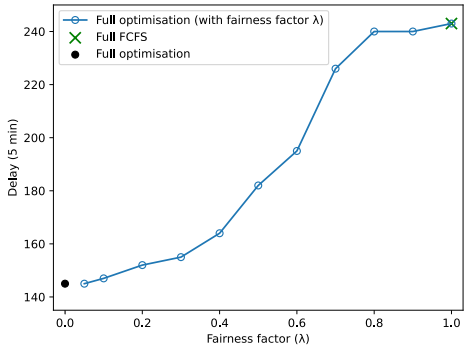


Fig. 9. Impact of the fairness factor to optimisation approach.

2) Optimisation Proportion: In line with Case-4, the hybrid implementation, the entire set of flight plans is divided into two parts, where the earlier submitted flight plans will be handled by the optimisation approach first. The optimised solution will be used to fill the common airspace occupancy status. Then, the rest of the flight plans will be processed through the FCFS approach based on their submission order.

This raises the question of how the ratio between the two parts of the flights would affect the conflict resolution efficiency and whether the embedded fairness factor will have a significant impact. It is expected that, as this ratio increases, the result will be closer to the full optimisation approach which means a decrease in total delay required.

We assume that the fairness factors are at 0, 0.3 and 0.5 respectively, producing three sets of results as presented in Fig. 10. With a low optimisation proportion (e.g., lower than 30%), it is obvious that the deconfliction efficiency is relatively low as the delay is quite high at all fairness factors. The value and trend of delay are the same when λ is 0.3 and 0.5, which indicates that the change of fairness factor only affects delay slightly at this stage. In those cases when the optimisation proportion is ranging from 30% to 70%, the total delay decreases gradually when the proportion increases. As expected, the required delay is higher when λ equals 0.5. From 70% to 90%, the delay for three fairness parameters

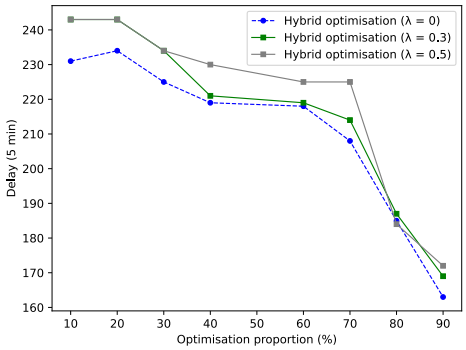


Fig. 10. Impact of the optimisation proportion to hybrid approach.

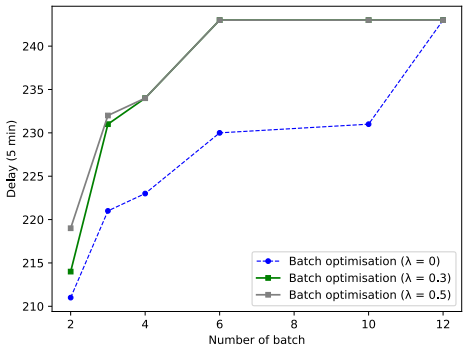


Fig. 11. Impact of the batch size to batch optimisation.

rapidly drops to some similar value. Results suggest that the optimisation proportion has a great impact on conflict resolution efficiency especially when it is at a high level.

3) Batch Size: Considering the use of optimisation approach in real-world operations, we envision that in most cases the flight plans might be processed in a number of batches, instead of being taken as a whole. The main idea of the batch optimisation is that, with flight plans divided into batches, the one containing earlier submitted flight plans will be processed first and the optimised results will be fixed. Then, flights from the next batch will be processed based on such common airspace occupancy, and so forth for the rest of the batches.

We first look into the impact of the varying batch size in the baseline Case-2 (without fairness factor), and then show the effects of fairness on the efficiency of conflict resolution. As the batch size grows (i.e., the time window for flight plan submission set for each batch increases), more flight plans will be included in every batch.

Fig. 11 shows how delay varies for different batch sizes, along with the effects of applying different fairness factors. The total delay gradually grows when the number of batches increases. It is similar and remains at a low level in all fairness settings when there are only 2 batches, which indicates that reducing the number of batches would improve the efficiency. Then the delay becomes stable when the batch size equals to or is larger than 4, and it reaches the maximum value after the batch size reaches 6, which means that there is no more optimisation space. Compared with the optimisation case ($\lambda = 0$), we can notice that incorporating the fairness leads to

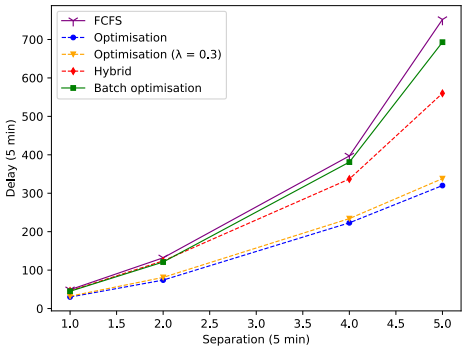


Fig. 12. Impact of separation minima to all the cases of study.

an increase in total delay. However, the three sets of results converge to the same point when n for excellence in applied research that has real-world impact and reflects our values of ambition and ithe batch size reaches a large number, which is because the batch optimisation (e.g., with a batch size of 12 in this example) would be extremely similar to FCFS itself.

4) Separation Minima: The effects of varying the time-based separation minima in each case are discussed. The default separation minima is set as 15-minute, where this value is initially set to take into account the uncertainty of UAS performance, UTM services and other aspects. The fairness factor (λ) is set as 0.3 and the optimisation proportion is 50% in the hybrid case, and the batch size is set to 5.

Results of the relationship between separation minima and total delay for the five cases of study are presented in Fig. 12. In general, delay rises with separation minima, where the most noticeable change can be found in the full FCFS case, while the least change is in the full optimisation ($\lambda = 0$) case. The total delay jumps from less than 500 minutes to more than 3,500 minutes in the FCFS case, with respect to the minimum and the maximum separation minima, which accounts for a nearly 3 folds increase. It can be seen that the full FCFS approach is not efficient in terms of deconfliction with a large separation minima. Worth noting that incorporating fairness does not show an obvious impact in this regard. The delay values of both hybrid and batch optimisation are in between the other cases.

E. Discussion

Through the above experiments and analysis, the approach proposed in this paper proves effective in strategic conflict resolution. However, there are still some concerns that may limit its performance, or they are presently based on some assumptions. In this section, we discuss the limitations and the envisioned solutions for future improvements.

One of the limitations is the computing performance of the optimisation model. The major affecting factors are the size of the airspace to be managed, the number of buffer volumes and separation minima. As mentioned above, the operating space is represented by the common airspace representation, and conflict detection and resolution are realised by searching all the volumes within it. As a result, the scale of the problem will increase with the size of the airspace. To address this problem, only the conflict volumes will be modelled without

losing the generosity of the approach. In such a case, the size of the problem will only depend on the number of conflict volumes. With the increasing of the airspace scale, generating trajectories can also become a bottleneck. Thus, other path planning algorithms such as Rapidly-exploring Random Trees (RRT), D*, and Probabilistic Roadmaps (PRM) will be investigated with a view to improving the searching efficiency.

While the uncertainty buffer used in this study is represented by adjacent buffer volumes, it may not be an accurate reflection of the uncertainty encountered by UAS during real operations. In particular, this simplified approach may not fully mitigate the uncertainty caused by unexpected events. To this end, a probability-based uncertainty representation could be used as a more precise approach, where the uncertainty can be modelled such as in normal distribution, with different eigenvectors in the along-track and cross-track directions.

V. Conclusion and Future Work

In this paper, we presented an optimisation model for the strategic conflict resolution service with a view to addressing the inefficiency of the conventional First-Come, First-Served (FCFS) approach. The model incorporates fairness based on time deviation, allowing the submission sequence to be considered during the delay allocation process. Both approaches were examined and compared through an illustrative example where five case studies were performed, including both FCFS and optimisation, along with their implementation variants depending on how the flight plans are grouped and processed. Experimental results show that both approaches can effectively resolve the conflicts, with optimisation producing much less delay than what is required by FCFS. Given the availability of flight plans at the time when decisions need to be made, the optimisation approach can be applied jointly with the FCFS approach, or it can be realised through batch computations as a practical solution. Flexible combination of the two methods can minimise delays while ensuring fairness.

Some questions still remain open, such as the prioritisation scheme of the optimisation model and how it can be designed to be compatible with the way that the fairness concern is currently addressed. As opposed to the exact optimisation approach used in this paper, heuristic algorithms such as Genetic Algorithm and Particle Swarming Optimisation, might be able to seek the solution within a reasonably short time. Comparing with those non-exact optimisation method will provide some added value in understanding the trade-off between solution quality and speed, which is useful for implementation and will be explored in our future work. Also, a specific mechanism could be further devised encompassing the RTTA concept proposed in U-space. In addition, it might be beneficial to better address the uncertainty factors associated with the flight plans, in such a way that potential overlap between the airspace volume occupancy (i.e., broken of the separation constraint) could be allowed to improve the efficiency of conflict resolution at the pre-fight phase.

Nomenclature

$l \in L$	set of *linear* flights
$a \in A$	set of *area* flights
$j \in J$	set of airspace volumes
$t \in T$	set of time moments
k	sliding time segment
s	safety time separation
L^j	subset of *linear* flight l traversing airspace volume j
A^j	subset of *area* flight a traversing airspace volume j
J_l	subset of airspace volumes that *linear* flight l is planned to traverse
J_a	subset of airspace volumes that *area* flight a is planned to traverse
T_l^j	subset of feasible time moments for *linear* flight l to enter airspace volume j
T_a^j	subset of feasible time moments for *area* flight a to enter airspace volume j
$J_l^{(i)}$	i^{th} $(1 \dots n)$ airspace volume of *linear* flight l
$J_a^{(i)}$	i^{th} $(1 \dots n)$ airspace volume of *area* flight l
$\overline{T}_l^j, \underline{T}_l^j$	upper and lower bound of feasible time moments T_l^j
$\overline{T}_a^j, \underline{T}_a^j$	upper and lower bound of feasible time moments T_a^j
r_l^j	scheduled time of *linear* flight l to enter airspace volume j
r_a^j	scheduled time of *area* flight a to enter airspace volume j
d_a	scheduled flight duration of *area* flight a
$\hat{t}_l^{jj'}$	scheduled flight time between segment jj' of *linear* flight l

Acknowledgment

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising.

References

[1] M. Graydon, N. A. Neogi, and K. Wasson, "Guidance for designing safety into urban air mobility: Hazard analysis techniques," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 2099.

[2] N. Neogi and A. Sen, *Integrating UAS Into the NAS—Regulatory, Technical, and Research Challenges*. Cambridge, U.K.: Cambridge Univ. Press, 2017, pp. 120–159.

[3] *Doc 9854 AN/458 Global Air Traffic Management Operational Concept*, Int. Civil Aviation Org., Montreal, QC, Canada, 2005.

[4] *U-Space Concept of Operations*, 3rd ed., CORUS, SESAR JU, Toronto, ON, Canada, 2019.

[5] *UTM Concept of Operations Version 2.0*, Federal Aviation Admin., Washington, DC, USA, 2020.

[6] *D5.2 Final Study Report*, DOMUS, SESAR JU, Chennai, India, 2020.

[7] V. Lappas et al., "EuroDRONE, a European UTM testbed for U-space," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 1766–1774.

[8] J. Rios, D. Mulfinger, J. Homola, and P. Venkatesan, "NASA UAS traffic management national campaign: Operations across six UAS test sites," in *Proc. IEEE/AIAA 35th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2016, pp. 1–6.

[9] J. Rios, "Strategic deconfliction: System requirements," NASA, UAS Traffic Manag. (UTM) Project, Washington, DC, USA, Final Rep., 2018. [Online]. Available: https://www.researchgate.net/profile/Joseph-Rios-3/publication/332107751_UAS_Traffic_Management_UTM_Project_Strategic_Deconfliction_System_Requirements_Final_Report/links/5ca10ced299bf11169547807/UAS-Traffic-Management-UTM-Project-Strategic-Deconfliction-System-Requirements-Final-Report.pdf

[10] J. L. Rios, J. Homola, N. Craven, P. Verma, and V. Baskaran, "Strategic deconfliction performance: Results and analysis from the NASA UTM technical capability level 4 demonstration," NASA, Washington, DC, USA, Tech. Rep., 2020.

[11] *Implementing an Open-Access UTM Framework for the U.K.*, Connected Places Catapult, London, U.K., 2020.

[12] L. Watkins, N. Sarfaraz, S. Zanlongo, J. Silbermann, T. Young, and R. Sleight, "An investigative study into an autonomous UAS traffic management system for congested airspace safety," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6.

[13] D. Sacharny, T. C. Henderson, M. Cline, B. Russon, and E. Guo, "FAA-NASA vs. lane-based strategic deconfliction," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2020, pp. 13–18.

[14] F. Ho et al., "Pre-flight conflict detection and resolution for UAV integration in shared airspace: Sendai 2030 model case," *IEEE Access*, vol. 7, pp. 170226–170237, 2019.

[15] F. Ho et al., "Decentralized multi-agent path finding for UAV traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 997–1008, Feb. 2022.

[16] X. Yang, P. Wei, M. Egorov, S. Munn, and A. Evans, "Stress testing of UAS traffic management decision making systems," in *Proc. AIAA Aviation Forum*, 2020, p. 2868.

[17] M. Egorov, A. Evans, S. Campbell, S. Zanlongo, and T. Young, "Evaluation of UTM strategic deconfliction through end-to-end simulation," in *Proc. 14th USA/Eur. ATM R&D Seminar*, 2021, pp. 20–23.

[18] *Unmanned Aircraft Systems Traffic Management (UTM)—A Common Framework With Core Principles for Global Harmonization*, ICAO, Montreal, QC, Canada, 2020.

[19] A. D. Evans, M. Egorov, and S. Munn, "Fairness in decentralized strategic deconfliction in UTM," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 2203.

[20] C. Chin, K. Gopalakrishnan, M. Egorov, A. Evans, and H. Balakrishnan, "Efficiency and fairness in unmanned air traffic flow management," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5939–5951, Sep. 2021.

[21] Y. Tang, Y. Xu, and G. Inalhan, "An integrated approach for on-demand dynamic capacity management service in U-space," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4180–4195, Oct. 2022.

[22] M. Stevens and E. Atkins, "Geofence definition and deconfliction for UAS traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5880–5889, Sep. 2021.

[23] *UAS Traffic Management Conflict Management Model*, FAA-NASA UTM Res. Transition Team, FAA, NASA, Washington, DC, USA, 2010.

[24] *Standard Specification for UAS Traffic Management (UTM) UAS Service Supplier (USS) Interoperability*, ASTM Int., West Conshohocken, PA, USA, 2022.

[25] S. Torres and K. L. Delpome, "An integrated approach to air traffic management to achieve trajectory based operations," in *Proc. IEEE/AIAA 31st Digit. Avionics Syst. Conf. (DASC)*, Oct. 2012, pp. 3E6-1–3E6-16.

[26] V. Battiste, A.-Q.-V. Dao, T. Z. Strybel, A. Boudreau, and Y. K. Wong, "Function allocation strategies for the unmanned aircraft system traffic management (UTM) system, and their impact on skills and training requirements for UTM operators," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 42–47, 2016.

[27] J. Lundberg, K. L. Palmerius, and B. Josefsson, "Urban air traffic management (UTM) implementation in cities—Sampled side-effects," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2018, pp. 1–7.

[28] *D2.2.010 High Level Conops—Initial*, AMU-LED, SESAR JU, Brussels, Belgium, 2021.

[29] *D4.1 Technical Description Report*, SESAR JU, Brussels, Belgium, 2021.

[30] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Oper. Res.*, vol. 46, no. 3, pp. 406–422, Jun. 1998.

[31] *UAS ATM Common Altitude Reference System*, EUROCONTROL, Brussels, Belgium, 2018.



Yiwen Tang (Graduate Student Member, IEEE) received the B.Eng. degree in traffic engineering from the Nanjing University of Aeronautics and Astronautics in 2016, the M.Sc. degree in air transport management from Cranfield University in 2019, and the M.Sc. degree in traffic engineering from the Nanjing University of Aeronautics and Astronautics in 2020. She is currently pursuing the Ph.D. degree in aerospace theme with Cranfield University.

Her current research interests include ATM and UTM.



Yan Xu (Member, IEEE) received the B.Eng. and M.Sc. degrees in traffic engineering from the Nanjing University of Aeronautics and Astronautics, and the Ph.D. degree in aerospace science and technology from the Technical University of Catalonia.

He is currently a Lecturer of ATM/CNS with the Centre for Autonomous and Cyber-Physical Systems, School of Aerospace, Transport and Manufacturing, Cranfield University. His current research interests include air traffic flow and capacity management, ATM/UTM, and urban air mobility.



https://dspace.org/

Incorporating optimization in strategic conflict resolution for UAS traffic management

Tang, Yiwen; Xu, Yan

2023-11-01
Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/TITS.2023.3290682
Tang Y, Xu Y. (2023) Incorporating optimization in strategic conflict resolution for UAS traffic management. IEEE Transactions on Intelligent Transportation Systems, Volume 24, Issue 11, November 2023, pp. 12393-12405

Downloaded from a DSpace repository. https://dspace.org/