

Enhancing U-Space Efficiency: An Advanced Optimization Framework for Strategic Deconfliction and Rescheduling

Zhiqiang Liu¹, Zihang Zhang, Juan José Ramos Gonzalez², and Jose Luis Munoz-Gamara³

Abstract—To guarantee a safe, efficient and resilient very low level airspace for diverse unmanned aerial vehicles, the U-space framework schedules trajectories in four dimensions - latitude, longitude, altitude and time - so that conflicts are avoided and capacity is maximized. In earlier work on the U-space service provider platform, conflicts were mitigated solely by time-shifting: requested flights were either authorized with a specific take-off slot inside the drone operator's requested launch window or rejected when no conflict-free slot could be found. This paper proposes an enhanced mitigation scheme that widens the range of options available to drone operators whose flight plans would otherwise be rejected. Instead of a single shift, the new scheme searches for alternative requested launch windows whose positions minimize both delay and the probability of generating new conflicts. Three optimization models for locating these alternative windows are formulated and embedded in strategic deconfliction service. Their effectiveness is assessed through Monte-Carlo simulations that replay real flight track data over Castelldefels (Barcelona). Comparative metrics - including acceptance ratio, average delay and computational cost - demonstrate the trade-offs between the models and confirm that the proposed strategy significantly increases flight plans acceptance while preserving safety.

Index Terms—U-space, strategic deconfliction, optimization and control, air transportation, UTM, constraint programming.

I. INTRODUCTION

THE rapid integration of Unmanned Aerial Vehicles (UAVs) into urban and suburban environments is poised to revolutionize industries ranging from logistics to emergency services, creating a new paradigm for intelligent transportation systems. This surge in autonomous air traffic, however, presents significant challenges to airspace safety, security, and operational efficiency [1], [2], [3], [4].

To manage this complexity, the European Commission mandated the development of a new generation traffic management system, giving rise to the U-space framework [5].

Received 1 September 2025; revised 10 February 2026; accepted 12 August 2026. This work was supported in part by the National Spanish Project "Improving the Planning and Management of Flight Contingencies in the Scalable Deployment of Drone Operations" funded by Ministry of Science and Innovation (MCIN)/State Research Agency (AEI)/10.13039/501100011033 under Grant PID2023-152053OB-C22. The Associate Editor for this article was X. Sun. (Corresponding author: Jose Luis Munoz-Gamara.)

The authors are with the Technical Innovation Cluster on Aeronautical Management, Universitat Autònoma de Barcelona, Sabadell, 08202 Barcelona, Spain (e-mail: Zhiqiang.Liu@uab.cat; Zihang.Zhang@student.univaq.it; JuanJose.Ramos@uab.es; JoseLuis.Munoz.Gamara@uab.cat).

Digital Object Identifier 10.1109/TITS.2026.3725514

The fundamental objective of U-space is to establish a safe, efficient, and secure Very Low Level (VLL) airspace capable of accommodating a wide variety of aircraft. This framework is actualized through a suite of services provided by certified U-space Service Providers (USSPs), who are tasked with managing shared airspace and ensuring that all operations adhere to the highest safety standards [3], [6]. According to current legislation, a scenario where multiple USSPs provide services in the same volume, coordinated by a Common Information Service Provider (CISP) [7] is possible.

A core responsibility of any USSP is to provide the conflict resolution service between Flight Plans (FPs, same as the U-plan defined in [6]). A conflict is formally defined as a loss of specified separation minima - the minimum required distances that aircraft must maintain to ensure safety [8], [9]. The process of detecting and eliminating these conflicts is known as deconfliction. The system must account for both actual and potential conflicts, which are situations that could escalate into a loss of separation minima if adjustments are not made to FPs [6]. The primary method for strategic deconfliction is to modify the take-off times of conflicting FPs, which is preferred as it avoids altering the geometric flight path requested by the Drone Operator (DO) [7], [10].

The Airspace Manager (AM) is the operational entity formally designated by the competent authority for each U-space volume [11], [12]. Using the data streams disseminated by the CISP, the AM engages the strategic deconfliction service provided by certified USSPs to verify each newly filed FP against all previously approved FPs. Even in scenarios where a single USSP operates exclusively within an airspace volume, the CISP remains involved to provide the common information services, while the USSP may additionally assume the AM's responsibilities for executing the strategic deconfliction process [6], [11], [12].

The operational procedure begins when a DO submits a FP containing a four-dimensional trajectory and a Requested Launch Window (RLW) to the AM. The AM consults the latest airspace structure, capacity constraints and according to the predefined strategic deconfliction time window parameters via the Common Information Service (CIS) provided by the CISP and runs strategic deconfliction algorithms to detect potential conflicts. If a conflict-free take-off slot exists within the RLW, AM approves the FP with a predefined Authorized Launch

TABLE I
ABBREVIATIONS

Abbrev.	Definition
1D	One-dimensional
4D	Four-dimensional
AGL	Above Ground Level
ALW	Authorized Launch Window
AM	Airspace Manager
ARLW	Alternative Requested Launch Window
ATM	Air Traffic Management
ATOT	Actual Take-Off Time
CD&R	Conflict Detection and Resolution
CDT	Confirmation Due Time
CIS	Common Information Service
CISP	Common Information Service Provider
CP	Constraint Programming
CP-SAT	Constraint Programming–SAT solver
CW	Confirmation Window
DCB	Dynamic Capacity Balancing
DO	Drone Operator
EASA	European Union Aviation Safety Agency
FCFS	First-Come First-Served
FP	Flight Plan
GA	Genetic Algorithm
JSSP	Job-Shop Scheduling Problem
LoWC	Loss of Well Clear
LNS	Large Neighbourhood Search
MAC	Mid-Air Collision
MILP	Mixed-Integer Linear Programming
MIP	Mixed-Integer Programming
NASA	National Aeronautics and Space Administration
NMAC	Near Mid-Air Collision
P-coins	Priority Coins
POD	Point of Departure
RLW	Requested Launch Window
RTL	Return-To-Launch
RTOT	Requested Take-Off Time
SESAR	Single European Sky ATM Research
ST	Solution Table
SW	Submission Window
TCL-4	Technical Capability Level 4
TLS	Target Level of Safety
UAM	Urban Air Mobility
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
USSP	U-space Service Provider
UTM	Unmanned Aircraft System Traffic Management
VLL	Very-Low-Level
VRP	Vehicle Routing Problem

Window (ALW). However, if no such slot can be identified, the FP is simply rejected [13], [14].

This binary “approve/reject” outcome exposes a significant limitation in current deconfliction services. Even when employing advanced mitigation policies like Batch processing [14] - which considers all request FPs simultaneously to maximize approvals - inevitable rejections occur in dense airspace scenarios. These rejections are problematic as they result in the underutilization of available airspace and offer no constructive feedback to the DO, forcing an inefficient, manual process of trial-and-error resubmission. This identifies a critical research gap: the lack of an automated, intelligent mitigation strategy that extends beyond rejection to proactively propose viable alternatives.

Strategic deconfliction for dense drone traffic has roots in both Air Traffic Management (ATM) and operations research scheduling. Early ATM studies [15] on free-flight airspaces treated conflict detection and resolution (CD&R) as a pairwise geometric problem, but subsequent work shifted toward system-level optimization to cope with high demand [16]. In the U-space and Unmanned Aircraft System Traffic Management (UTM) context, [17] presented a review of multiple CD&R methods for both manned and unmanned aviation. Work [18] evaluates how strategic and tactical CD&R modules perform and interact in VLL airspace across varying traffic densities and under uncertainties such as wind and departure delays. Work [14] and [19] demonstrated that heterogeneous USSP performance can severely curtail shared-capacity utilization, thereby motivating globally coordinated algorithms rather than purely competitive ones. Work [20] and [21] introduced a grid-to-interval transformation that converts four-dimensional (4D) trajectory conflicts into one-dimensional (1D) forbidden intervals, enabling the use of standard scheduling solvers and forming the algorithmic backbone for several SESAR-JU prototypes [10], [22]. Work [13], [14], [23], and [24] explored the impact of First-Come First-Served (FCFS) and Batch submission methods on airspace capacity. The results show that there is still significant room for optimization of VLL airspace capacity.

Multiple research groups have framed deconfliction as a variant of the Job-Shop Scheduling Problem (JSSP), where aircraft correspond to jobs and airspace cells to machines [10], [25], [26], [27]. Comparative studies reveal that Constraint Programming (CP) [28] provides superior scaling relative to Mixed-Integer Linear Programming (MILP) on large instances, especially when temporal granularity falls below 10s [26], [29]. To gain additional speed, heuristic meta-schedulers such as Large Neighbourhood Search (LNS) [30] and Genetic Algorithms (GA) [31] have been coupled with CP back-ends [32], but at the expense of optimality guarantees. Meanwhile, robust optimization frameworks have emerged that incorporate wind and positioning uncertainty, yet these methods still assume that any FP failing the primary optimization is irrevocably discarded [20].

Beyond pure scheduling, negotiation-based mechanisms have also been explored. The SESAR CORUS-XUAM very-large-scale demonstrations introduced bilateral slot-trading between co-located USSPs to enhance fairness and overall capacity utilization [3], [6]. In the United States, NASA’s UTM program has prototyped an air-traffic marketplace in which operators bid for launch slots through a combinatorial-auction mechanism that was assessed during the Technical Capability Level 4 (TCL-4) field campaign [33]. Although such market approaches raise economic efficiency, they still offer no deterministic guarantee that a FP rejected by the primary optimizer will obtain a viable alternative, and they often place additional decision-making burden on DOs.

Recent work has also investigated synergistic strategic deconfliction actions, combining departure delays with speed control and rerouting for 4D route optimization [34]. In addition, continuous time-window formulations with standardized slots/windows have been studied to better utilize airspace

resources within a fixed planning horizon [23]. The present work is positioned differently: it deliberately restricts mitigation to temporal adjustments that preserve the submitted trajectory geometry (Section II-B) and extends the temporal search span beyond the original RLW to provide feasible alternative launch-time windows for initially rejected FPs.

Despite these advances, three persistent gaps can be identified:

- 1) **Post-rejection recovery.** Existing prototypes either drop rejected FPs or leave recovery to manual resubmission cycles, leading to under utilized airspace and operator frustration.
- 2) **Flexible solution output.** Most solvers return a single point estimate (one take-off time). Continuous or multi-interval windows - essential for robust timetabling and downstream negotiation - are rarely produced.
- 3) **Quantified trade-offs.** Acceptance ratio improvements are typically reported, but few works rigorously analyze how flexibility and computational effort co-vary across alternative modelling choices.

Addressing these gaps requires an optimization framework that searches beyond the RLW, delivers continuous or disjoint windows when multiple feasible regions exist, and exposes the trade-off surface among acceptance, flexibility, and computation time. The models developed in this study are designed to satisfy these criteria and thereby convert a “hard” rejection into a constructive, data-driven rescheduling opportunity.

To address this deficiency, this paper proposes and evaluates a set of advanced optimization models designed to automatically generate feasible and flexible alternative take-off time windows for FPs that are initially rejected. The approach is to extend the search for a solution beyond the original RLW into a larger, predefined Alternative Requested Launch Window (ARLW) [35]. Three distinct models are developed to increasing complexity:

- A single-adjustment model that seeks a discrete alternative take-off time.
- A single-interval model that provides a continuous time interval for greater flexibility.
- A multiple-interval model that identifies all possible non-contiguous take-off windows, representing the most comprehensive solution set.

These models are implemented using state-of-the-art CP solvers and are rigorously evaluated using real-world drone flight data to analyze the trade-offs between solution flexibility, acceptance rate, and computational feasibility. The ultimate goal is to provide AM with a powerful tool that not only enhances airspace occupancy but also improves service quality by transforming a rejection into a constructive, flexible rescheduling opportunity.

The remainder of this paper is organized as follows: Section II details the strategic deconfliction service, including the problem background, trajectory representation, spatio-temporal conflict representation and optimization objective. Section III introduce the methodology that converts grid-based conflict detection into one-dimensional forbidden-interval constraints for the optimization stage. Section IV presents the

formal definitions and mathematical representation of three proposed models. Section V describes the experimental setup and presents a comparative analysis of the simulation results. Finally, Section VI concludes the paper with a summary of findings and an outline for future research.

II. STRATEGIC DECONFLICTION SERVICE

In the context of Unmanned Aircraft Systems (UAS) flight authorization, the strategic deconfliction service constitutes a critical component within the U-space regulatory framework established by the European Union Aviation Safety Agency (EASA) [7]. During the pre-flight phase, the AM is responsible for the strategic resolution of potential conflicts. The issuance of flight authorization is contingent upon the successful strategic deconfliction of the proposed FP against existing approved operations.

The deconfliction process involves a predictive analysis of potential conflicts by comparing newly submitted FPs with those previously authorized. Should the predicted trajectories indicate a probability of violating separation minima that exceeds a predetermined threshold, the strategic deconfliction service is initiated. This service then generates mitigation strategies, which may include modifications to the planned trajectory or adjustments to the FP’s temporal parameters, to ensure adherence to the required separation standards.

This research employs a time-based separation methodology to mitigate conflicts by adjusting the take-off times of conflicting UAS FPs. The approach is designed to resolve operational conflicts while preserving the originally planned flight trajectories of the DOs. The DronAs utilizes a suite of optimization algorithms to coordinate take-off time windows, thereby effectively resolving conflicts among multiple FPs. This methodology was initially developed for ATM within the PARTAKE project [10], [20], [21].

Figure 1 delineates the operational timeline, progressing from the initial planning phase to the tactical phase, which commences with the departure of the approved FPs. The DO is required to submit a FP that specifies the flight trajectory and a RLW. The RLW is a time interval that includes the Requested Take-off Time (RTOT), with the start time of the RLW serving as its temporal reference. The planning phase encompasses the following sequential steps:

Initially, DOs must submit their FPs to AM, including a RTOT inside RLW, within a designated Submission Window (SW, the time interval prior to take-off during which the submission of FPs is permitted). Subsequently, the AM employs strategic deconfliction algorithms to identify and resolve potential conflicts among the submitted FPs. A given FP is either approved, signifying the existence of at least one ALW, or it is rejected if strategic deconfliction is not feasible. The ALW is designed to be a sub-interval of the RLW, thereby accommodating the DO’s operational preferences. Throughout the Confirmation Window (CW, the time interval between the end of the SW and the start of the RLW), the strategic deconfliction algorithm retains the capability to modify feasible ALWs to accommodate new FP approval requests. The final ALW for approved missions is communicated to the DOs

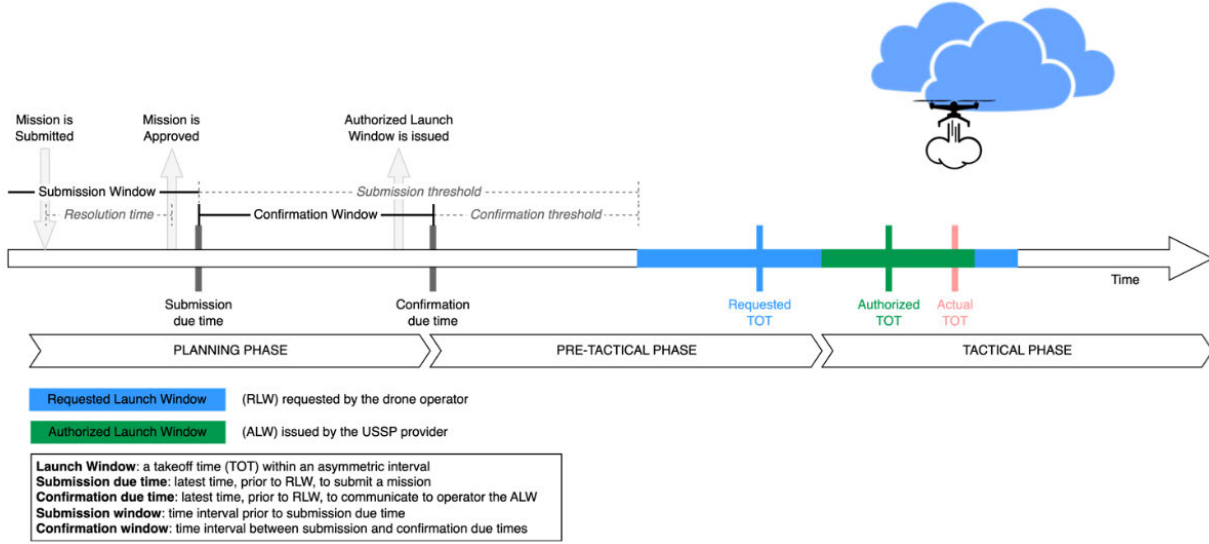


Fig. 1. Strategic deconfliction service. A post-rejection recovery stage is executed after each scheduling cycle (Section II-C, Fig. 4); during this stage, new FP submissions are paused to prevent overlap with the subsequent scheduling cycle.

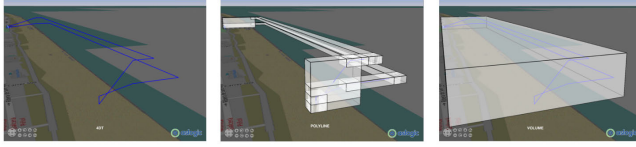


Fig. 2. Three flight trajectory representations.

no later than the Confirmation Due Time (CDT) which is the deadline of the CW [13], [14], [36].

Upon issuance of the ALW to the DO, the pre-tactical phase commences in preparation for the flight. The initiation of the ALW signifies the beginning of the tactical phase. The DO is obligated to operate in accordance with the established regulations and to initiate take-off within the confines of the allocated ALW. This adherence ensures the absence of strategic conflicts, defined as a loss of separation minima, with other planned FPs.

For clarity on the interaction between scheduling rounds, the post-rejection recovery is treated as a separate stage executed after a scheduling cycle (see Section II-C and Fig. 4). During this short recovery stage, new FP submissions are temporarily paused, so that recovery decisions are not made in parallel with the acceptance and authorization of newly filed FPs.

A. Trajectory Representation

A foundational step in transforming operational flight missions into a tractable mathematical model is to define a data-based representation of each flight trajectory. This representation provides the basic information unit for strategic airspace management, and its granularity affects the level of functionality that USSPs can provide and, ultimately, airspace utilization.

A flight trajectory can be interpreted as the spatio-temporal volumes within which the UAS operator intends to conduct

operations under nominal conditions. The associated operational volume is characterized in four dimensions (latitude, longitude, altitude, and time) and includes uncertainty buffers to account for navigation performance, flight technical error, path definition error, and system latencies.

As illustrated in Fig. 2 [14], three principal trajectory representations are commonly considered. The 4D format (Fig. 2 left) represents the mission as a sequence of 4D waypoints, where each waypoint specifies latitude, longitude, altitude, and time, and is augmented with temporal and positional uncertainty. The polyline format (Fig. 2 middle), as specified in [11], models the trajectory as a sequence of one or more 4D volumes described by lateral and vertical boundaries together with discrete entry and exit times; uncertainty buffers are incorporated in each dimension, and the level of detail of these polyline volumes can vary, which affects the precision of the FP description. The volume format (Fig. 2 right) specifies a single airspace volume that contains the FP for its entire duration without further trajectory detail, reserving the entire volume for the expected flight time.

The proposed workflow requires a time-ordered sequence of 4D samples to perform conflict computation and to build the corresponding constraint. Hence, 4D-format flight plans can be used directly. In contrast, both polyline-format and volume-format submissions are handled conservatively because the intended trajectory cannot be uniquely inferred from them. For polyline-format submissions, each polyline segment is treated as an individual volume whose activation time window is determined by the temporal dimension associated with that segment. This results in multiple smaller time-bounded volumes rather than a single large continuous one. When only a single volume-format submission is provided, the entire declared volume is treated as occupied for the full duration of the operation. In both cases, these conservative assumptions yield more restrictive constraints and may reduce FP acceptance due to increased reserved airspace.

The choice of trajectory representation influences subsequent analysis not only in terms of data structure but also in terms of implied airspace occupancy. Compared with the polyline and volume formats, the 4D format represents the mission with higher trajectory granularity and typically results in lower reserved airspace, which can increase FP acceptance and improve airspace utilization [14], [19].

B. Spatio-Temporal Conflict Representation

To maintain a secure, flexible, and efficient airspace, the primary objective of USSPs is the detection and mitigation of conflicts while simultaneously maximizing airspace utilization. The foundational concept of separation minima, originally established for manned aviation and subsequently adapted for UAS operations, defines the minimum required displacement between aircraft to ensure the achievement of an agreed-upon Target Level of Safety (TLS). Separation minima are categorized as: lateral separation, the minimum horizontal distance; vertical separation, the minimum altitude difference; and longitudinal separation, the minimum distance for aircraft following co-linear or identical flight paths [6], [11].

In the U-space and CORUS-XUAM context, a conflict refers to a predicted breach of the specified separation minima between airborne flights within controlled airspace. An actual breach is classified as a Loss of Well Clear (LoWC), while more severe occurrences are designated as Near Mid-Air Collision (NMAC) or Mid-Air Collision (MAC) [6], [11]. Such situations pose a significant threat to airspace safety and are addressed through a process termed deconfliction.

Furthermore, it is critical to address the concept of potential conflicts. A potential conflict is a latent condition wherein two or more FPs, not presently in conflict, would result in a separation minima violation if specific modifications, such as adjustments to take-off time, speed, or route, were not applied to either plan.

In the flight planning phase, this process is initiated when DOs submit their FPs, these submissions are systematically analyzed by the AM to detect both actual and potential conflicts over a specified temporal horizon. Following detection, the AM executes a mitigation process involving the approval, adjustment, or rejection of FPs to ensure a conflict-free state. The outcomes of this process, including any required modifications to the submitted FPs, are then formally communicated back to the respective DOs.

Spatial rerouting is not considered in this work. Under the U-space framework, flight authorization is plan-based and requires that an accepted operation is free of intersection in space and time with other notified flight authorizations [11]. In addition, when an operational risk assessment is required, it must identify risks on the ground and in the air and describe the intended operation's operational environment and geographical area [12]. Therefore, conflict mitigation is restricted to temporal adjustments that preserve the submitted trajectory geometry.

C. Optimization Objective

Conflict mitigation is the process of resolving detected issues by applying adjustments to FPs. Within the scope of

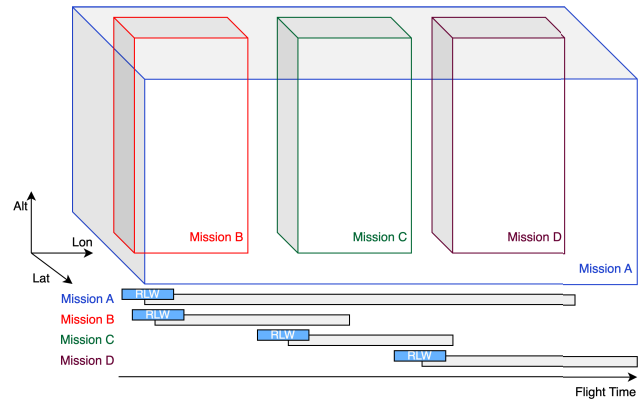


Fig. 3. Spatiotemporal interactions between four missions.

this paper, mitigation is achieved exclusively by modifying the take-off time window of conflicting FPs.

During this process, some FPs may be rejected to allow for the temporal adjustment of others. The criteria for these decisions are defined by a mitigation policy. A conventional approach is the FCFS policy, which prioritizes missions based on their submission order. In contrast, a Batch mitigation policy disregards submission order and processes a set of missions concurrently to maximize the total number of approvals, thereby enhancing airspace usage efficiency. This may, however, lead to the rejection of an earlier submitted FP if it is incompatible with a more optimal solution for the entire Batch.

Under FCFS policy, FPs are processed sequentially according to their submission order. Once an FP is accepted in a scheduling round, it is treated as fixed, and a newly arriving FP is rejected if no conflict-free ALW can be found within its RLW against the already accepted set. Hence, when two FPs conflict, the earlier-arriving FP is retained, while the later-arriving FP is rejected if it cannot be accommodated by temporal adjustment.

Under Batch policy, a set of FPs is optimized jointly with the objective of maximizing the number of accepted missions. Therefore, the rejected FPs are those not selected in the returned acceptance-maximizing subset. When multiple acceptance-maximizing subsets exist, a random tie-breaking mechanism is applied among equivalent optima to avoid deterministic bias induced by a fixed ordering.

Maximizing accepted FPs is an efficiency-driven objective and may yield multiple optimal solutions with different distributions of rejections and delays across operators. This reflects the efficiency-fairness trade-off discussed in UTM allocation studies [37]. In this work, FCFS provides procedural transparency (fairness-by-order), whereas the Batch formulation is identity-symmetric (no operator-specific preference is encoded). To avoid systematic disadvantages when multiple optima exist, random tie-breaking among equivalent acceptance-maximizing solutions is used.

Equity-aware criteria can be incorporated as secondary objectives/constraints after achieving maximum acceptance, consistent with equity-related discussions in shared U-space performance assessments [14].

The FCFS policy's inherent lack of foresight regarding later submitted FPs typically results in a lower mission acceptance

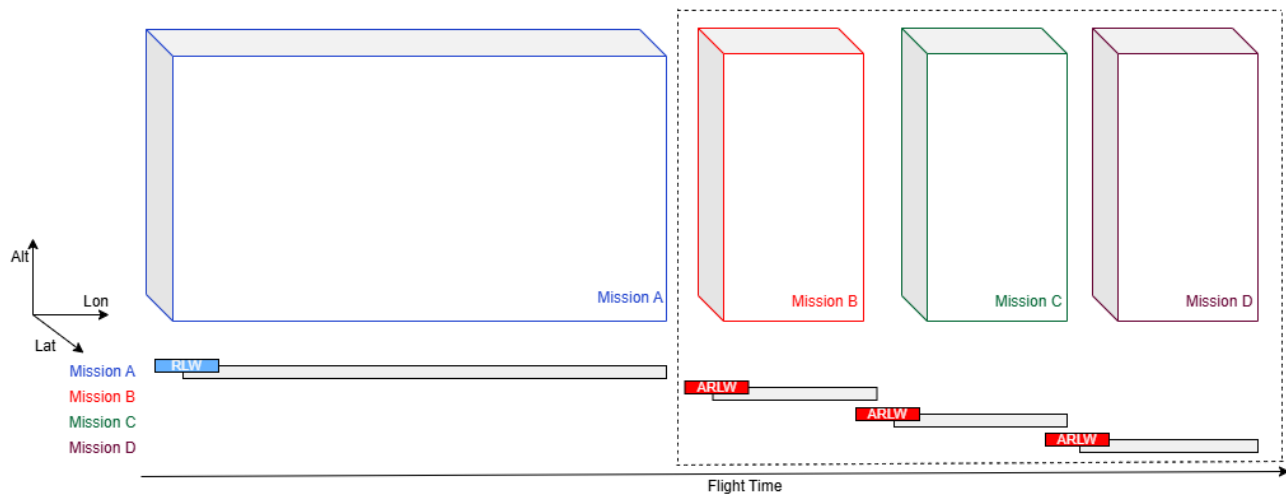


Fig. 4. Mitigation for finding alternatives.

rate compared to the Batch approach. This is a consequence of the immutability of an approved mission; it cannot be rescinded to accommodate a new request.

A spatiotemporal interaction occurs when two or more trajectories have a non-empty geometric intersection and their use of that intersection overlaps in time. As shown in Figure 3, such intersections exist between Missions A and Mission B, C, D with overlapping usage times based on declared flight durations. Missions B, C, and D have overlapping time intervals but no geometric intersections, so no spatiotemporal interaction exists among them.

In the Figure 3 scenario, if Mission A's request arrives first, it is approved, and the remaining three are rejected, as no shift of the ALW within Mission A's RLW can avoid conflicts. This yields a 25% acceptance ratio, regardless of the SW or CW. Conversely, if any of B, C, or D arrives first, all except A are approved, giving a 75% ratio. This shows arrival order has far more impact on efficiency than SW or CW.

Under a Batch policy, requests are evaluated together, and if the goal is to maximize approvals, this example also yields a 75% acceptance ratio. A prior simulation study, wherein authorization requests were submitted in Batches, reported an average acceptance rate of 61% under a Batch policy, a 35% improvement over the traditional FCFS policy [13], [14].

However, even a Batch policy will unavoidably reject some FPs. It is therefore logical to propose a subsequent process that offers alternative take-off time options outside the initially RLW. This mechanism provides a secondary opportunity for operators to reschedule FPs by extending the RLW for rejected FPs and identifying feasible slots within it.

Figure 4 presents an example with alternative requested launch window. In the initial Batch mitigation, Missions A is approved, while Missions B, C and D are rejected. By extending the initial RLW, as represented in the dashed line square, ARLWs could be identified for the previously rejected Missions B, C and D. It is important to note that even with an extended RLW, feasibility is not guaranteed for all rejected FPs.

The methodology advanced in this study focuses on FPs rejected during an initial mitigation phase. The proposed concept involves processing new requests for a set period, after which new submissions are paused. The previously rejected FPs are then aggregated into a Batch, and ARLWs are systematically identified to accommodate as many as possible. Following this, the standard mitigation policy resumes. This cyclical approach increases the overall FP acceptance rate by re-evaluating and optimizing rejected requests. This study will assume a Batch policy is used for both the initial mitigation and the alternative-search phase to ensure consistency and maximize airspace usage.

Limited research has addressed the provision of alternative take-off time windows in dynamic scheduling, the primary objective of this research is to develop and implement optimization models capable of generating feasible ARLWs for FPs that were initially rejected. These models aim to automate the search for alternatives, providing DOs with optimal suggestions in real-time to reduce manual workload. The key goals are to maximize the aggregate efficiency of airspace, minimize the systemic impact of schedule adjustments, and ensure the proposed solutions are robust.

III. METHODOLOGY

As previously established, a submitted FP is analyzed by a strategic conflict detection algorithm to identify any potential or existing conflicts with previously approved operations. The core methodology employed in this work follows a two-step process: first, the 4D trajectory of each operation is discretized into a voxel-based spatial grid, enabling the detection of spatio-temporal interactions through the identification of overlapping occupied volumes within their respective ALWs. Second, the conflict mitigation phase addresses these interactions by applying a time-shifting strategy to the involved trajectories, effectively transforming the resolution problem into a temporal allocation task while preserving the originally requested flight paths.

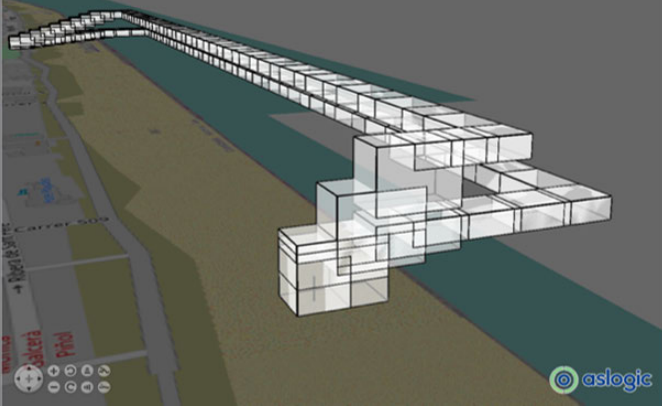


Fig. 5. Illustration of the airspace digitalization.

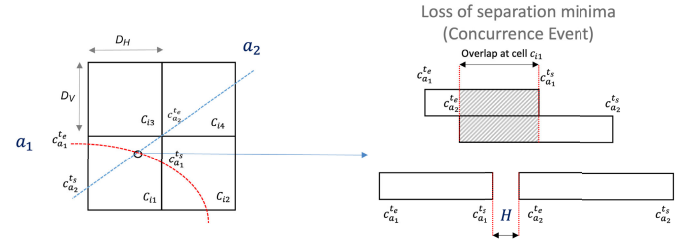
To facilitate conflict detection, the airspace is discretized into a 3D grid of voxels, with horizontal and vertical dimensions defined by the applicable separation minima and adjusted for navigation performance. Using a spatial projection, each continuous 4D trajectory is segmented at the points where it intersects voxel boundaries (Fig. 5), and the resulting segments are stored as sequences of 4D points (latitude, longitude, altitude, and time), with the first and last points denoting cell entry and exit events, respectively [20]. Each voxel is considered occupied for the duration of the ALW. This ALW-long occupancy assumption is conservative and ensures conflict-freeness for any admissible take-off time within the ALW. In practice, voxel entry/exit times can be variable or uncertain across executions. This can be captured by interval-based robustification (enlarging the protected time intervals) or by probabilistic occupancy modeling with chance constraints to maintain a prescribed conflict-risk level while reducing conservatism [38]. A spatio-temporal conflict is detected when voxels from different trajectories overlap in both space and time. In such cases, the mitigation algorithm resolves conflicts by time-shifting the ALWs of the affected flights. Under the FCFS policy adopted in these experiments, strategic deconfliction is performed upon each new flight authorization request, which is approved only if no conflict exists or if any detected conflict can be resolved through ALW adjustment.

Consider two flights a_1 and a_2 and a discretized cell c . Let the nominal entry and exit times of a_k in c be $c_{a_k}^{te}$ and $c_{a_k}^{ts}$ ($k \in \{1, 2\}$). Given the U-space separation minima (horizontal D_H , vertical D_V), a concurrence event occurs in c if the time intervals $[c_{a_1}^{te}, c_{a_1}^{ts}]$ and $[c_{a_2}^{te}, c_{a_2}^{ts}]$ overlap, which is equivalent to a conflict inside cell c . The conflict interval is:

$$[\max(c_{a_1}^{te}, c_{a_2}^{te}), \min(c_{a_1}^{ts}, c_{a_2}^{ts})] \quad (1)$$

In this paper, D_H and D_V (hence the cell size) are set to 30 meters and 15 meters respectively.

Following the airspace discretization process, the detection of spatio-temporal interdependencies analyzes the temporal clearance between pairs of aircraft segments occupying the same cell. Based on the above definition (Fig. 6 [20]), the algorithm performs a neighborhood analysis to capture concurrence events across cell boundaries: for each occupied cell j , it systematically inspects all surrounding cells to detect any

Fig. 6. Conflict in cell c of two flights.

potential conflict with aircraft operating in adjacent airspace volumes.

Throughout the paper, the RLW/ALW are represented as time-shift intervals with respect to the RTOT. Without loss of generality, the ALW is normalized to the form $[0, u]$ (of length u). Hence, the Actual Take-Off Time (ATOT) is:

$$\text{ATOT}_i = \text{RTOT}_i + d_i, \quad d_i \in [0, u] \quad (2)$$

This normalization is also reflected in the figures. The more general case $[\ell, u]$ can be obtained by a constant time translation and is omitted to avoid unnecessary notational burden.

As established in Section II, the flight authorization process introduces temporal uncertainty. A DO's FP submission includes a RLW, and an approved plan is returned with an ALW, which is a sub-interval of the RLW. The FP is permitted to take off at any time within this ALW. Consequently, it is imperative to consider all potential conflicts that could arise from any valid take-off time shift within the permitted launch windows. This temporal flexibility increases the allowable time variation and necessitates a method to manage the resulting conditional concurrence relationships. To store this information, the concept of the forbidden interval is introduced.

A forbidden interval is defined as the range of relative time adjustments between two FPs that would result in a conflict. Consider two FPs, a_1 and a_2 , with respective requested take-off times t_1 and t_2 . Let d_1 and d_2 be the variable time adjustments, representing the deviation of the actual take-off time from the requested time for each flight. The actual take-off times, $t_1 + d_1$ and $t_2 + d_2$, must remain within their respective RLWs (RLW 1 and RLW 2):

$$t_1 + d_1 \in \text{RLW}_1 \quad (3)$$

$$t_2 + d_2 \in \text{RLW}_2 \quad (4)$$

If a spatio-temporal conflict between these two missions occurs when $d_1 - d_2$ lies within the interval $[s, e]$, then to ensure deconfliction the following condition must hold:

$$d_1 - d_2 \notin [s, e]. \quad (5)$$

Figure 7 illustrates the concept for two flights a_1 and a_2 traversing the same cell c . Let $(c_{a_k}^{te}, c_{a_k}^{ts})$ be the nominal (entry, exit) times of flight a_k in c . Model the RLW and the ALW of a_k as time-shift intervals with respect to its requested schedule, and let $d_k \in \text{ALW}_k \subseteq \text{RLW}_k$ be the applied time shift. The actual entry/exit times are:

$$t_{e,\text{act}}^{a_k} = c_{a_k}^{te} + d_k, \quad t_{s,\text{act}}^{a_k} = c_{a_k}^{ts} + d_k \quad (6)$$

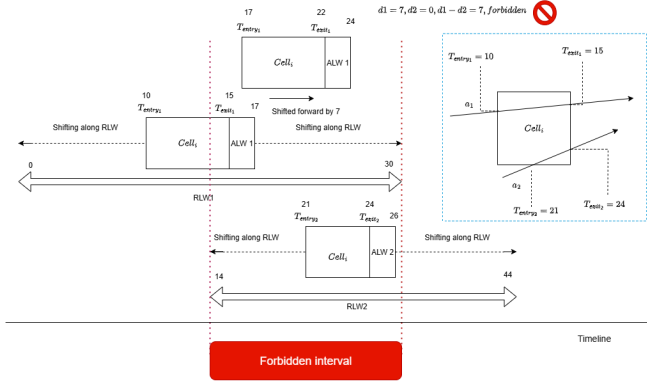


Fig. 7. Forbidden interval.

A conflict in cell c occurs if the adjusted intervals overlap,

$$[c_{a_1}^{t_e} + d_1, c_{a_1}^{t_s} + d_1] \cap [c_{a_2}^{t_e} + d_2, c_{a_2}^{t_s} + d_2] \neq \emptyset \quad (7)$$

which is equivalent to the relative shift $\Delta = d_1 - d_2$ lying within the forbidden interval:

$$[s, e] = [c_{a_2}^{t_e} - c_{a_1}^{t_s}, c_{a_2}^{t_s} - c_{a_1}^{t_e}] \quad (8)$$

Therefore, deconfliction requires:

$$d_1 \in \text{ALW}_1, \quad d_2 \in \text{ALW}_2, \quad \text{and} \quad d_1 - d_2 \notin [s, e] \quad (9)$$

The detection algorithm systematically computes these forbidden intervals. It analyzes all potential conflicts by considering any allowable take-off time within the RLWs, incorporating an additional time buffer for cell entry and exit times to account for the ALW. The algorithm then detects and aggregates the conflict intervals from all relevant cells for each pair of flights. This process yields a set of forbidden intervals specific to each flight pair, which may contain zero, one, or multiple distinct intervals depending on the conflict geometry.

In this manner, the detection algorithm transforms the complex 4D spatio-temporal interdependencies between flights into a set of 1D forbidden intervals. This simplified 1D representation is then utilized as a primary input for the subsequent conflict mitigation process.

A. Mitigation Model

The primary objective of the mitigation process is to analyze the identified conflicts and provide DOs with deconflicted take-off time authorizations that maximize overall airspace utilization. This work employs a time-based separation strategy, which resolves conflicts exclusively by adjusting the take-off times of the conflicting FPs. The principal advantage of this approach is that it preserves the originally requested flight trajectory, offering a minimally invasive solution for the DO.

Leveraging the concept of forbidden intervals, the mitigation process under a Batch policy which - seeks to maximize the number of approved missions - can be formulated as a mathematical optimization problem. The model is presented below:

1) Entities:

$$\text{Missions} : a_1, \dots, a_n \quad (10)$$

$$\text{Conflicts} : c_1, \dots, c_m \quad (11)$$

- n : number of missions.
- m : number of conflicts.
- Each conflict c involves missions a_i and a_j , forbidden interval $[s, e]$, represented by $(a_i, a_j, [s, e])$.

2) Decision Variables:

$$dvar_i = (d_i, p_i), \quad \text{for } i \in \{1, \dots, n\} \quad (12)$$

- d_i : time adjustment for mission i , domain: d_i in RLW
- p_i : presence (boolean) for mission, domain: $\{0, 1\}$

3) Constraints:

$$\forall c : (a_i, a_j, [s, e]) \\ ((p_i = 1 \wedge p_j = 1) \Rightarrow (e_i - s_j < s \vee s_i - e_j > e)) \quad (13)$$

4) Optimization Goal:

$$\max \sum_i p_i \quad (14)$$

In this model, decision variables are the time adjustments of missions, which indicate their ALWs, and their presence states. $p_i = 0$ means flight mission a_i is rejected. The constraint is that for each conflict detected $(a_i, a_j, [s, e])$, the time adjustment difference of the two missions involved (a_i, a_j) , cannot fall into the forbidden interval $([s, e])$. In the end, optimization goal is to maximize the number of approved missions.

B. Optimization Tools

Google OR-Tools [39] is an open-source software suite created by Google to solve a variety of optimization problems and other research groups have conducted the investigation on CP [40], MLIP [41], and Vehicle Routing Problem (VRP) [42] with it. The core components of OR-Tools are written in C++, with wrappers available for Java, .NET, and Python. It offers several technological advantages over other solutions, including:

- A thriving community that ensures robust stability and ongoing development. This results in regular updates, bug fixes, and the introduction of new features.
- Consistent high performance, powered by multithreaded algorithms designed for complex computations, allowing for faster results without additional hardware or development costs.
- Flexible constraint management and infrastructure selection, enabling quick and efficient deployment while minimizing infrastructure expenses.

Given the advantages highlighted above, OR-Tools has been selected as the optimization tool for this study. Additionally, Python has been chosen as the programming language due to its compatibility with OR-Tools and its seamless integration with other components of the USSP platform, DronAS. The problem under investigation can be formulated as a MIP problem. OR-tools offers primarily two methods to address this type of problem:

- MP solver: A wrapper for several third-party MIP solvers, which use standard branch-and-bound techniques.
- CP-SAT solver: A constraint programming solver that uses SAT (satisfiability) methods.

As a general rule, MIP solvers are more effective for problems that can be formulated as standard linear programs with additional arbitrary integer variables, whereas the CP-SAT solver is often preferable for problems where the majority of variables are Boolean. In the alternative-searching model, similar to the mitigation model discussed in Section III-A, the decision variables primarily include two types: time adjustments (integer) and mission presence (Boolean). Additionally, as stated in [43], “for typical MIP problems that involve both integer and Boolean variables, there is often no significant speed difference between the two solvers, making either option viable”. Moreover, the OR-tools documentation provides an example of a job shop scheduling problem solved using the CP-SAT solver [32]. Considering this, the CP-SAT solver is selected as the method for solving the alternative-searching problem.

IV. ALTERNATIVE SOLVING MODELS

A. Single-Adjustment Model

A foundational approach for accommodating previously rejected FPs is the Single-Adjustment model. The core principle of this model is to extend the original RLW into a larger temporal timeframe, termed the ARLW, and to search for a deconflicted ALWs for each rejected FP within this expanded window.

This model is a direct extension of the initial mitigation framework. It can be conceptualized as a re-application of the initial mitigation algorithm under modified conditions. Specifically, the set of already approved FPs is treated as a fixed and unmodifiable regulated traffic set. The optimization is then performed solely on the rejected FPs, utilizing their expanded ARLWs as the new temporal boundaries for the search.

1) Entities:

$$\text{Missions} : a_1, a_2, \dots, a_n \quad (15)$$

$$\text{Conflicts} : c_1, c_2, \dots, c_m \quad (16)$$

$$\text{Regulated Missions} : R \quad (17)$$

- n : number of missions.
- m : number of conflicts.
- $R : \forall i \in R, a_i$ has been approved, $i \in \{1, \dots, n\}$.
- Each conflict c involves missions a_i, a_j , and a forbidden interval $[s, e]$, represented by $(a_i, a_j, [s, e])$.

2) Decision Variables:

$$\text{dvar}_i = (s_i, p_i), \quad i \in \{1, \dots, n\} \quad (18)$$

- s_i : alternative adjustment of mission i , domain: s_i in ARLW
- p_i : presence (boolean) for mission i , domain: $\{0, 1\}$

3) Constraints:

$$i \in R \Rightarrow (p_i = 1 \wedge s_i = d_i) \quad (19)$$

$$\forall c : (a_i, a_j, [s, e])$$

$$((p_i = 1 \wedge p_j = 1) \Rightarrow (s_i - s_j < s \vee s_i - s_j > e)) \quad (20)$$

4) Optimization Goal:

$$\max \sum_i p_i \quad (21)$$

The model entities are first defined. The set of missions is partitioned into two subsets: the Regulated set, comprising missions that were approved during the initial mitigation phase and whose ALWs are now considered fixed parameters; and the set of candidate missions for which a new take-off time is sought.

Conflicts are defined as tuples, each containing a pair of missions and their corresponding forbidden interval. The set of all conflicts is a function of the temporal window under consideration; a longer window increases the number of potential conflicts. Consequently, the conflict set for this model must encompass all potential conflicts that could arise within the expanded ARLW.

The decision variables for each candidate mission i are: a continuous variable representing its temporal adjustment within the ARLW, and a binary variable representing its presence (i.e., approval) in the solution.

The model is subject to the following constraints:

- Regulated Missions (Equation 19): The first constraint enforces the immutability of the approved missions. For any mission in the Regulated set, its presence variable is fixed to 1, and its adjustment variable is fixed to its previously determined value.
- Conflict Resolution (Equation 20): The second constraint ensures deconfliction. For each identified potential conflict between a pair of missions i and j , if both missions are selected for approval, the difference in their temporal adjustments ($s_i - s_j$) is constrained to lie outside the corresponding forbidden interval $([s, e])$.

Finally, consistent with a Batch mitigation policy, the objective function is to maximize the total number of approved missions. It is important to note that due to practical constraints on solver execution time, the implemented algorithm may not converge to the theoretical global optimum. Instead, it is configured to return the best feasible solution found within the allocated computation time, yielding what is often referred to as a quasi-optimal solution.

B. Single-Interval Model

While the Single-Adjustment model provides a valid solution for generating alternative take-off times, it exhibits certain limitations in its operational flexibility. The model yields a single point-in-time adjustment, s_i , for each rejected mission i . Given a standard ALW of length l , this solution corresponds to a fixed take-off interval of $[s_i, s_i + l]$. This approach is inherently restrictive, as it explores only a fraction of the potentially available deconflicted solution space.

To address this limitation and enhance the flexibility offered to DOs, an improved formulation is proposed. The central innovation of this model is the generation of an interval of adjustments for each mission, defined by a start time s_i and an end time e_i . This interval $[s_i, e_i]$ itself constitutes a new time window rather than a single point from which a fixed-length window is derived. The valid take-off time shifting is constrained to the decision interval:

$$\text{ATOT}_i = \text{RTOT}_i + d_i, \quad d_i \in [s_i, e_i] \subseteq \text{RLW}_i \quad (22)$$

The fundamental distinction in the mathematical formulation of Single-Interval model lies in the deconfliction constraint. Instead of ensuring that the difference between two scalar adjustment values lies outside a forbidden interval, the new constraint must guarantee that the entire range of possible differences between the two adjustment intervals does not overlap with the forbidden interval. The mathematical presentation of this model is as follows:

1) *Entities:*

$$\text{Missions} : a_1, a_2, \dots, a_n \quad (23)$$

$$\text{Conflicts} : c_1, c_2, \dots, c_m \quad (24)$$

$$\text{Regulated Missions} : R \quad (25)$$

- n : number of missions.
- m : number of conflicts.
- $R : \forall i \in R, a_i$ has been approved, $i \in \{1, \dots, n\}$.
- Each conflict c involves missions a_i, a_j , and a forbidden interval $[s, e]$, represented by $(a_i, a_j, [s, e])$.

2) *Decision Variables:*

$$\text{dvar}_i = (s_i, e_i, p_i), \quad i \in \{1, \dots, n\} \quad (26)$$

- s_i : start of adjustment interval of mission i , domain: s_i in ARLW
- e_i : end of adjustment interval of mission i , domain: e_i in ARLW
- p_i : presence (boolean) for mission i , domain: $\{0, 1\}$

3) *Constraints:*

$$\forall i \in \{1, \dots, n\} : (e_i \geq s_i) \quad (27)$$

$$i \in R \Rightarrow (p_i = 1 \wedge s_i = e_i = d_i) \quad (28)$$

$$\forall c : (a_i, a_j, [s, e]), \quad ((p_i = 1 \wedge p_j = 1) \Rightarrow (e_i - s_j < s \vee s_i - e_j > e)) \quad (29)$$

$$\sum p_i \geq P \quad (30)$$

4) *Optimization Goal:*

$$\max \sum_i (e_i - s_i) \quad (31)$$

The entities of Single-Interval model are identical to those in Single-Adjustment model. However, the decision variables for each candidate mission are expanded from a two-element tuple to a three-element tuple, comprising a binary presence variable and two continuous variables, s_i and e_i , which represent the start and end of the adjustment interval, respectively.

The constraints are formulated as follows:

- **Interval Validity** (Equation 27): The first constraint ensures that for each generated adjustment interval, the end time is greater than or equal to the start time $e_i \geq s_i$.
- **Regulated Missions** (Equation 28): The second constraint fixes the adjustment intervals for the previously approved missions, enforcing the immutability of their schedules.
- **Interval Deconfliction** (Equation 29): The third constraint introduces the crucial distinction of this model. Given an adjustment interval $[s_i, e_i]$ for mission i and $[s_j, e_j]$ for mission j , the complete range of possible temporal differences between them is the interval $[s_i - e_j, e_i - s_j]$. This constraint mandates that this derived interval of differences must have no intersection with the predefined forbidden interval for the conflicting pair (i, j) .
- **Acceptance Rate Consistency** (Equation 30): To ensure that the goal of maximizing interval length does not compromise the number of successful missions, a fourth constraint is introduced. This constraint fixes the total number of approved missions to equal P , where P is the maximum number of missions that could be approved, as determined by the prior optimization stage based on the Single-Adjustment Model.

As finding minimally short intervals would be of limited operational utility, the objective of this model is to maximize the total length of the generated adjustment intervals for all approved missions.

C. Multiple-Interval Model

While Single-Interval model enhances flexibility by generating a single ARLW, its solution represents a convex approximation of what may be a non-convex or disjoint feasible solution space. A single ARLW may not capture the complete set of all possible deconflicted take-off opportunities.

To overcome this limitation and provide a more comprehensive set of solutions, a further extension, designated as Multi-Interval model, is proposed. This model is designed to identify multiple, disjoint alternative intervals for each mission, thereby offering a more accurate and exhaustive representation of the available launch windows to DOs.

It is noted that generating multiple disjoint ARLWs does not imply that all such intervals are simultaneously reserved. The model characterizes an admissible (conflict-free) set under take-off uncertainty within the ALW; operationally, only one executable ARLW is ultimately selected, while the remaining intervals represent additional feasible options rather than consumed capacity.

The formulation of this Multi-Interval model necessitates significant modifications. A set of decision variables representing multiple start and end points must be defined for each mission. Consequently, the deconfliction constraints must be expanded to consider all combinatorial pairings of intervals between any two conflicting missions, ensuring that each pair of intervals remains compliant with the corresponding forbidden interval constraints.

The theoretical formulation of this model is presented as follows:

1) Entities:

$$\text{Missions} : a_1, a_2, \dots, a_n \quad (32)$$

$$\text{Conflicts} : c_1, c_2, \dots, c_m \quad (33)$$

$$\text{Regulated Missions} : R \quad (34)$$

- n : number of missions.
- m : number of conflicts.
- R : $\forall i \in R, a_i$ has been approved, $i \in \{1, \dots, n\}$.
- Each conflict c involves missions a_i, a_j , and a forbidden interval $[s, e]$, represented by $(a_i, a_j, [s, e])$.

2) Decision Variables:

$$dvar_i = (s_{ik}, e_{ik}, p_i), \quad i \in \{1, \dots, n\}, k \in \{1, \dots, q_i\} \quad (35)$$

- s_{ik} : start time of the k -th feasible interval for mission i within its ARLW; domain: $s_{ik} \in \text{ARLW}_i$.
- e_{ik} : end time of the k -th feasible interval for mission i within its ARLW; domain: $e_{ik} \in \text{ARLW}_i$.
- p_i : presence indicator (boolean) for mission i ; domain: $\{0, 1\}$.
- q_i : number of feasible intervals considered for mission i .

3) Constraints:

$$\forall i \in \{1, \dots, n\} :$$

$$i \in R \Rightarrow (p_i = 1 \wedge \forall k \in \{1, \dots, q_i\} : s_{ik} = e_{ik} = d_i) \quad (36)$$

$$p_i = 0 \Rightarrow (\forall k \in \{1, \dots, q_i\} : s_{ik} = e_{ik} = 0) \quad (37)$$

$$\forall k \in \{1, \dots, q_i\} : s_{ik} \leq e_{ik} \quad (38)$$

$$\forall k \in \{1, \dots, q_i - 1\} : e_{ik} \leq s_{i,k+1} \quad (39)$$

$$\forall c : (a_i, a_j, [s, e]) : (p_i = 1 \wedge p_j = 1) \Rightarrow$$

$$\left(\forall k \in \{1, \dots, q_i\}, \forall l \in \{1, \dots, q_j\} : e_{ik} - s_{jl} < s \right. \\ \left. \vee s_{ik} - e_{jl} > e \right) \quad (40)$$

$$\sum_{i=1}^n p_i \geq P \quad (41)$$

4) Optimization Goal:

$$\max \sum_{i=1}^n \sum_{k=1}^{q_i} (e_{ik} - s_{ik}) \quad (42)$$

In the formulation, the decision variable q_i denotes the maximum number of admissible intervals for mission i . This bound is given by the number of pairwise conflicts involving mission i plus one, since each conflict can partition its ARLW into at most two disjoint sub-intervals. In the worst-case scenario of mutually non-overlapping conflicts, k conflicts yield at most $k + 1$ intervals.

The constraints are organized into three parts:

- Generic interval rules and provisions for pre-classified missions:
 - Regulated missions (Equation 36): For missions contained in the regulated set, the corresponding decision variables are fixed at the prescribed values.
 - Rejected missions (Equation 37): For missions that have been excluded, every interval variable is set to zero, preventing the optimiser from allocating capacity to them.

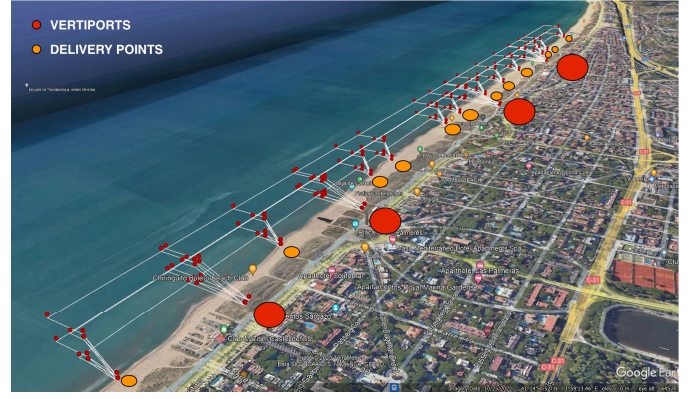


Fig. 8. Operational scenario in Castelldefels.

– Interval directionality (Equation 38): The start time of each interval is constrained to be no greater than its end time.

– Interval ordering (Equation 39): Intervals are indexed so that the end of interval k does not exceed the start of interval $k + 1$.

- Conflict-avoidance with forbidden intervals (Equation 40): Ensures that, for any two conflicting missions, the difference in their selected adjustment times never lies within the corresponding forbidden interval $[s, e]$. Every possible pair of mission intervals is inspected so that no overlap with the forbidden regions can occur.
- Presence maximization (Equation 41): Preserves the global acceptance ratio by maximizing the number of accepted missions.

After optimization, contiguous intervals associated with the same mission are merged, when feasible, to produce a more compact and operationally convenient schedule.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section reports representative numerical results obtained with the three optimization models introduced in Section IV. The sensitivity of each model to key preset parameters—most notably the window length—is quantitatively assessed, and a comparative evaluation of their relative performance is presented.

A. Simulation Setup

The simulation campaign reported herein is grounded in the Spanish CORUS-XUAM demonstration undertaken in Castelldefels (Barcelona). That exercise proved the ability of the U-space ecosystem to supervise logistics-UAS operations in medium-scale urban and suburban environments flying in controlled airspace. A mesh of vertiports - designated Points of Departure (PODs) - was emulated, from which four independent drone operators scheduled departures and arrivals to execute last-mile-delivery missions (Figure 8 [14]).

All flight operations follow the same sortie structure: a UAV departs from its operator's POD, flies to a delivery point, and returns to the same POD for turnaround, reflecting continuous last-mile delivery rotations. Moreover, the number of available

UAVs is assumed to be sufficient for each operator such that fleet availability does not constrain the submitted demand in the reported experiments.

Each POD acted as a logistics hub: consignments arriving by conventional modes were transferred to multirotor aircraft that subsequently served 12 delivery points distributed across a 3 km radius [3], [6].

Traffic was channelled through a predefined airspace structure consisting of four parallel corridors that supported continuous delivery rotations. Two corridors carried westbound flows and two carried eastbound flows. Vertical separation was enforced by assigning discrete flight levels: 30 m and 70 m Above Ground Level (AGL) for westbound traffic, and 40 m and 80 m AGL for eastbound traffic. Combined with a deliberate lateral offset between counter-flow corridors, this stratification enabled safe crossings between PODs and delivery points. The offset also provided a buffer for unplanned climbs - e.g., loss-link Return-To-Launch (RTL) procedures - without compromising vehicles operating at the upper levels.

The reference scenario is used here to synthesize traffic sets in which each operator submits FPs with randomly generated RTOT. All simulation results presented in the sequel are based on Monte-Carlo samples drawn from that dataset. For each simulation run, a sample of 50 missions is randomly selected. The RTOT for these missions are uniformly distributed within a 30-minute window. Each mission is assigned a RLW with a duration of 20 minutes (1200 seconds), commencing at its respective RTOT. For missions that are rejected during the initial mitigation phase, a search for a deconflicted schedule is performed within an ARLW. The ARLW is defined as a 50-minute window that also commences at the mission's originally RTOT.

B. Result Of Single-Adjustment Model

The Single-Adjustment model operates in two stages. First, an initial Batch mitigation is performed on all submitted FPs ($a_1, \dots, a_n$) to determine a set of adjustments ($d_1, \dots, d_n$) within their respective RLWs, with the objective of maximizing the number of approved missions. In the second stage, the missions rejected in the first stage are re-evaluated. This stage aims to find a single, deconflicted adjustment value for each rejected mission within an expanded ARLW.

Figure 9 illustrates the results of this process. The mission labels are color-coded: blue for missions approved in the first stage; green for missions initially rejected but approved in the second stage; and red for missions that remain rejected. The approved adjustments for the initial set are represented by blue dots, which fall within the RLW (ending at the 1200-second purple line). The alternative adjustments found for the secondarily approved missions are represented by green dots, which lie within the ARLW but outside the original 20 mins RLW.

The red dashed segments mark candidate time slots for missions that remained unapproved after the initial pass, assuming conflicts only against the initially approved (blue) set. They are not feasible solutions overall because they ignore

conflicts involving the secondarily approved (green) missions and any conflicts among the red missions themselves.

In this simulation with 50 missions, 32 were approved in the first (batch) mitigation and 18 were rejected. During the alternative-search over the extended window, the algorithm recovered 11 of those 18, leaving 7 finally rejected. For instance, JUNO-23 and JUNO-25 still fail because they conflict with secondarily approved (green) missions, even though they have conflict-free windows relative to the blue set alone.

In terms of performance, this model increased the airspace utilization from 32/50 (64%) to 43/50 (86%), a relative increase of 22% in the number of accommodated missions. This result represents the maximum achievable acceptance rate for this scenario. However, the flexibility for the 11 newly approved missions is minimal, as each is assigned only a single point-in-time for take-off. A trade-off exists between maximizing the acceptance rate and providing greater operational flexibility. Offering alternative intervals instead of single adjustments could increase flexibility, potentially at the cost of the overall acceptance rate. This limitation motivates the development of the subsequent models.

C. Result Of Single-Interval Model

In the Single-Interval model, the objective is to provide an alternative take-off time adjustment interval, $([s_i, e_i])$, for missions that were initially rejected, in contrast to the single adjustment value (d_i), from the previous model. In cases where the solution yields $s_i = e_i$, the interval degenerates to a single point in time. This section presents a simulation result for this model and provides an interpretation of its performance.

For this experiment, the minimum number of secondarily approved missions (from the set of initially rejected missions) was constrained to 11 (According to Eq. (30)). As depicted in Figure 10, the results show that some alternatives are represented by an interval between two green dots, rather than a single point. Consistent with the Single-Adjustment model, 32 out of the 50 missions were approved in the initial mitigation phase. Of the 18 rejected missions, 11 were subsequently approved in the alternative-search phase, satisfying the specified constraint.

A key finding is that 4 of these 11 newly approved missions (JUNO-13, OMAHA-19, SWORD-24, and UTAH-46) were assigned ARLWs with a length greater than zero. The total sum of the lengths of all generated intervals for the 11 approved missions was 920 seconds.

The increase in operational flexibility can be quantified by comparing the take-off choice available to the recovered missions. In the baseline Single-Adjustment model, each recovered mission is assigned a fixed launch time; operationally, this corresponds to an ALW of 120 seconds that is anchored to that single adjusted time, i.e., the operator has effectively no freedom to reposition the window in time. Summing the fixed ALW duration over the 11 secondarily approved missions yields $11 \times 120\text{ s} = 1320\text{ s}$, which is a constant quantity by construction and should be interpreted as the total authorized window duration, rather than an indicator of scheduling flexibility. In contrast, the Single-Interval model outputs, for each recovered mission, a feasible placement range

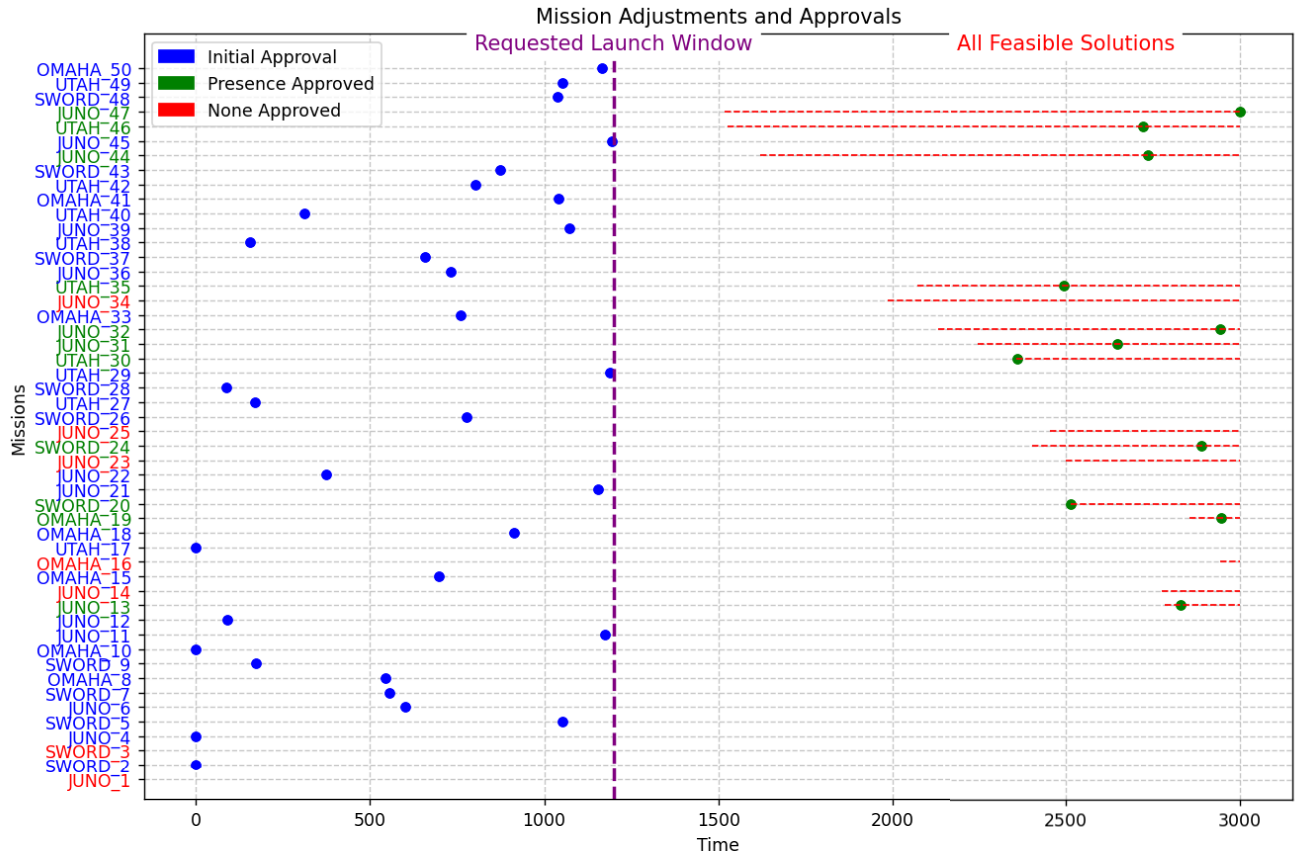


Fig. 9. A visualized result from the Single-Adjustment model.

within which the same fixed-length ALW can be positioned without introducing conflicts. The reported 920 seconds is computed as the sum of the lengths of these feasible placement ranges across all 11 recovered missions, and it represents the additional temporal slack provided by the Single-Interval formulation beyond a single anchored launch time. Therefore, for this specific scenario, the Single-Interval model provides an additional flexibility margin equal to 69.7% (920/1320) of the total fixed ALW duration, while the Single-Adjustment baseline provides (approximately) zero such slack because the launch times are fixed.

To support the above Single-Interval model illustration with statistical evidence, the resulting flexibility indicators are summarized in Fig. 11. The left panel reports the distribution of the total length of generated ARLWs normalized by the number of recovered missions, which represents the average temporal slack provided per recovered FP. The right panel reports the proportion of degenerate solutions ($s_i = e_i$). While degenerate intervals occur under highly constrained conflict geometries or due to local optimality of the search-based solver, a substantial fraction of recovered missions still receive non-degenerate intervals, confirming that the proposed formulation provides tangible operational flexibility beyond a fixed, anchored take-off assignment.

D. Result Of Multiple-Interval Model

The Multiple-Interval model is designed to identify more than one ARLW for each mission. This approach, in theory,

can generate the complete set of feasible take-off times, thereby maximizing operational flexibility for DOs. For this experiment, the minimum number of recovered (secondarily approved) missions is controlled by the parameter P in Eq. (41). The value $P = 11$ reported in the Single-Interval results corresponds to the maximum number of recoverable missions obtained in that baseline Single-Adjustment model recovery stage for the same instance, and it is used as a reference target. However, it is not guaranteed to remain feasible in the Multiple-Interval model. This is because the Multiple-Interval model constraints impose a strictly stronger feasibility requirement: conflict-freeness must hold simultaneously across all authorized sub-interval combinations, rather than for a single selected interval per mission. As a consequence, enforcing $P = 11$ can render the Multiple-Interval instance infeasible. In the reported case, the solver did not return a feasible schedule under $P = 11$ within the fixed computation budget; therefore, P was set to the largest feasible recovery bound found for this instance, namely $P = 9$, and this value is used for the Multiple-Interval performance evaluation presented below.

Figure 12 shows that 9 missions were secondarily approved; however, the specific set of approved missions differed from that of the Single-Interval model. Notably, the model successfully generated multiple, disjoint alternative windows for some missions (e.g., UTAH-46 was assigned the intervals [1527, 1574] and [2995, 3000]).

Fig. 13 provides distributional evidence of the practical gain delivered by the Multiple-Interval formulation. The left panel shows that the average interval lengths obtained by the two

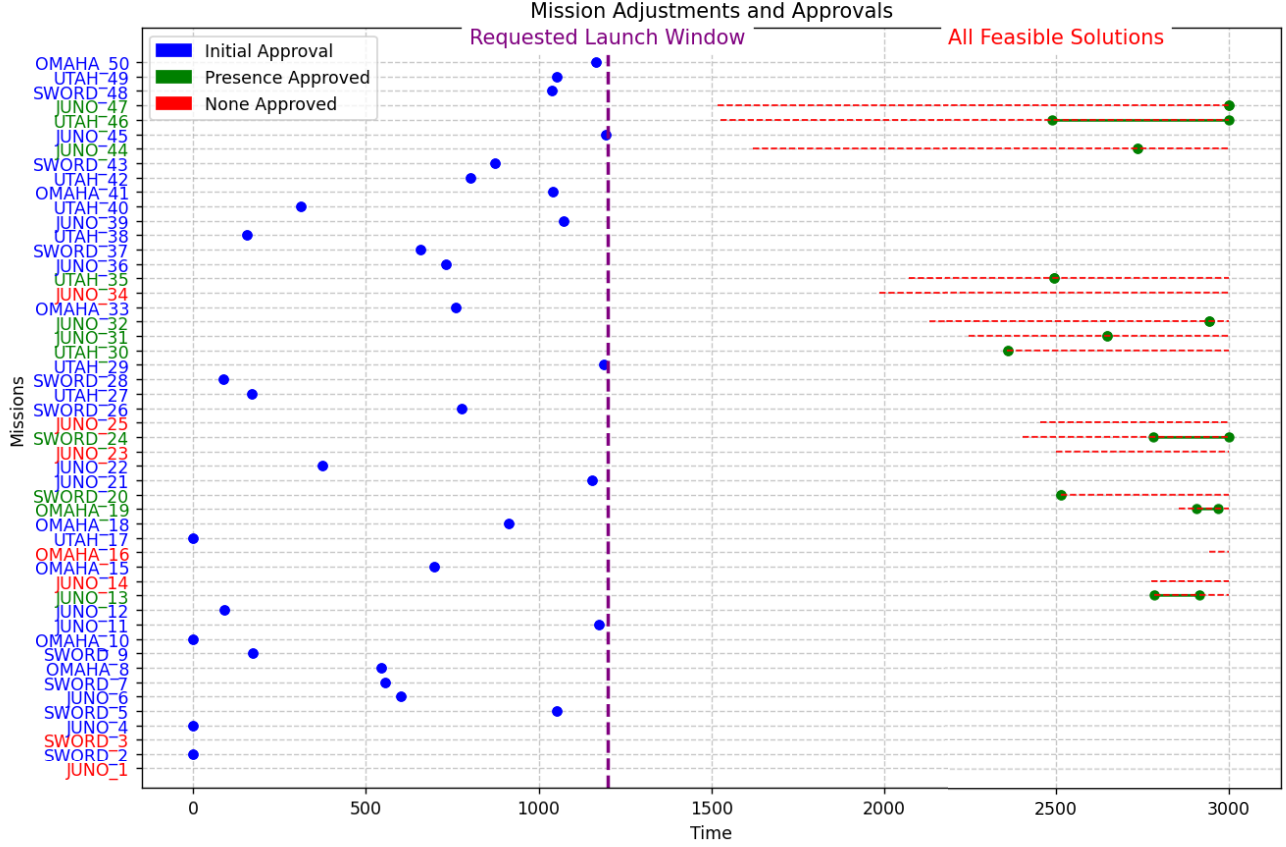


Fig. 10. A visualized result from the Single-Interval model.

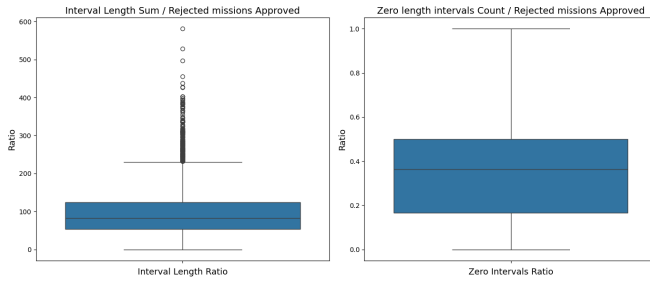


Fig. 11. Flexibility distribution of the Single-Interval model over Monte-Carlo runs. Left: ratio of the total generated interval length to the number of recovered (secondarily approved) missions. Right: ratio of zero-length intervals among recovered missions.

models are nearly identical in distribution, indicating only marginal improvement in solution quality. The right panel shows that the occurrence of multiple disjoint alternatives is rare for the tested traffic settings: in most runs the ratio is close to zero, with only occasional cases exhibiting a noticeable presence of multiple intervals. This empirical pattern explains why the theoretical completeness of the Multiple-Interval model translates into limited operational benefit in the evaluated corridor-based scenario.

The total length of all generated alternative intervals was 1266 seconds, exceeding the 920 seconds achieved by the simpler Single-Interval model. This aligns with theory, since the

feasible set of the Multiple-Interval model contains that of the Single-Interval model. Nevertheless, the search space grows roughly quadratically with the number of candidate intervals and the solver operated under a finite time budget; accordingly, the 1266-second outcome should be viewed as a quasi-optimal solution that could likely improve with additional compute.

E. Results Analysis

To gain insight into the performance of the proposed models, a series of comprehensive simulation experiments were conducted. These experiments were designed to evaluate the relationship between the length of the ARLW and the mission acceptance rate, quantify the operational flexibility provided by the interval-generating models, and compare the performance of the Single-Adjustment, Single-Interval and Multiple-Interval approaches. Three mitigation schemes were evaluated:

- **Impact of ARLW Length on Acceptance Ratio.** A study was conducted to analyze how the length of the ARLW affects the secondary acceptance rate of the Single-Interval model. Over 1000 simulation runs, each with a unique, randomly sampled traffic set of 50 missions, the ARLW length was varied. Figure 14 shows that enlarging the ARLW raises mission acceptance. The x-axis is the ARLW-to-RLW length ratio (e.g., 1.25 means a 25 % extension); the y-axis is the share of

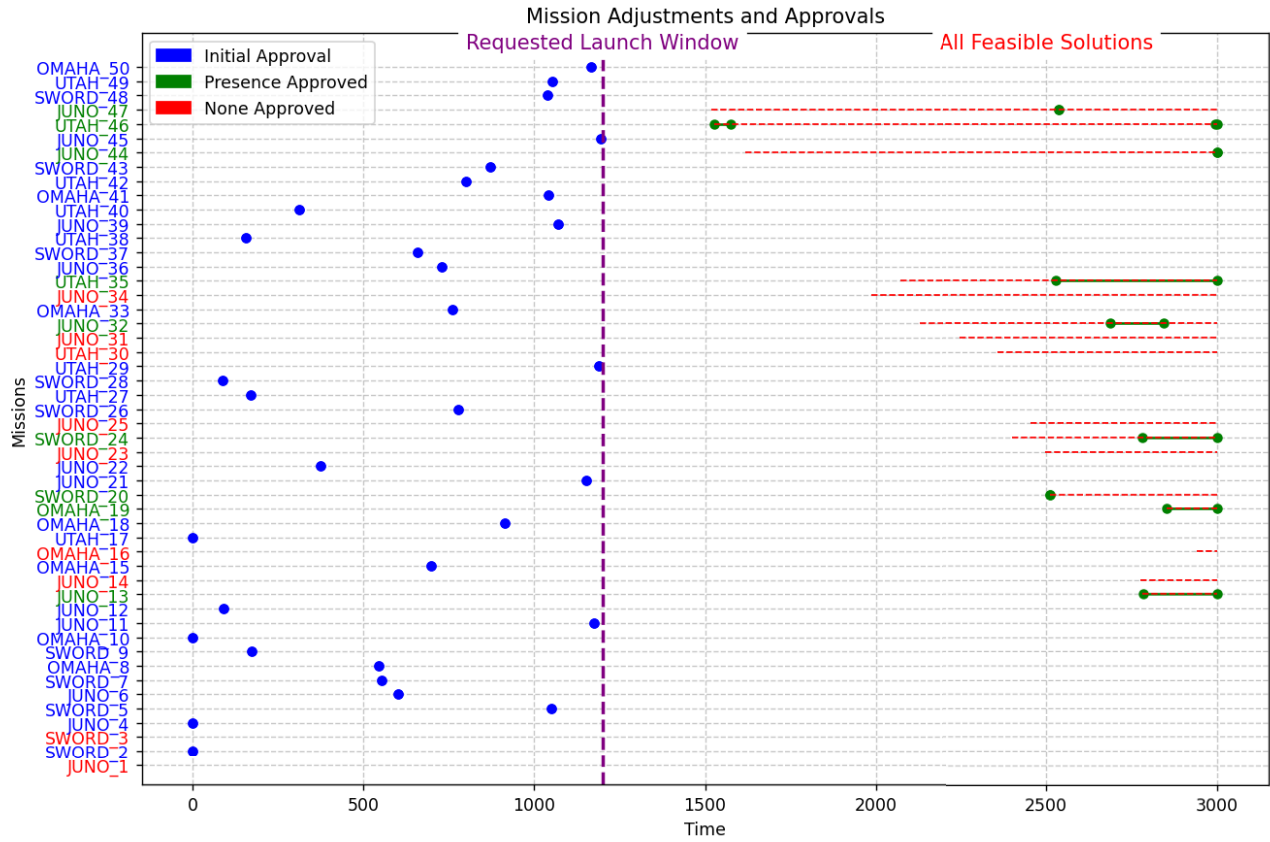


Fig. 12. A visualized result from the Multiple-Interval model.

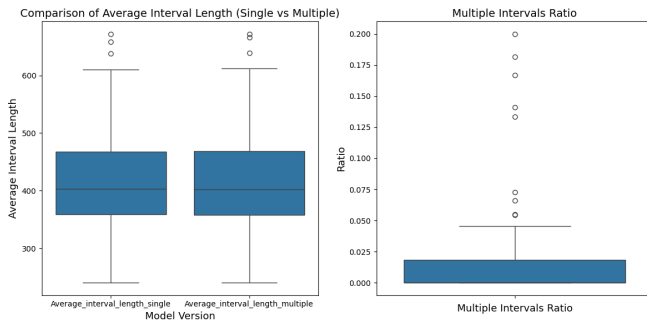


Fig. 13. Comparison between Single-Interval and Multiple-Interval models. Left: distribution of the average interval length per recovered mission. Right: ratio of solutions in which more than one disjoint interval is found for a mission (Multiple-Interval ratio).

newly approved missions relative to the initial approvals (e.g., 0.2 means 6 extra approvals out of 30). A linear regression analysis confirmed relationship between the ARLW length and the number of newly approved missions, yielding an R-squared value of 0.876 ($p < 0.001$). The model's slope coefficient of 0.2731 indicates that for every 100% increase in the ARLW length relative to the original RLW, the secondary acceptance ratio increases by approximately 27.3%. This finding quantitatively shows that providing a larger temporal search space directly improves the system's capacity to accommodate previously rejected missions.

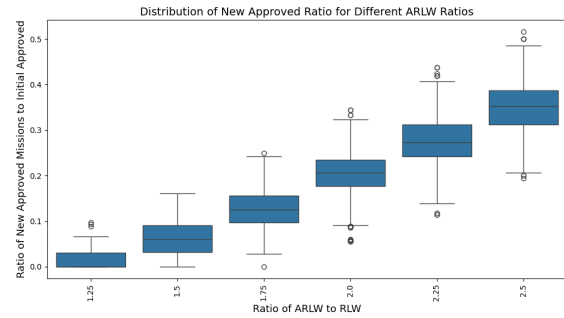


Fig. 14. Relations between ARLW and acceptance ratio.

- **Flexibility and Solution Characteristics of the Single-Interval Model.** The flexibility provided by the Single-Interval model was quantified by analyzing the total length of the generated ARLWs. Compared to a baseline Single-Adjustment model limited to a fixed 120-second ALW, the Single-Interval model demonstrated an average flexibility increase of 81.1%. An analysis of the solution characteristics revealed that, on average, 32.8% of the generated ARLWs were “zero-length” intervals, effectively single points in time. This phenomenon can be attributed to highly constrained conflict geometries or to the optimization solver converging on a local optimum. Nonetheless, a significant majority (67.2%) of the solutions were non-degenerate intervals, providing tangible operational flexibility to operators. This outcome

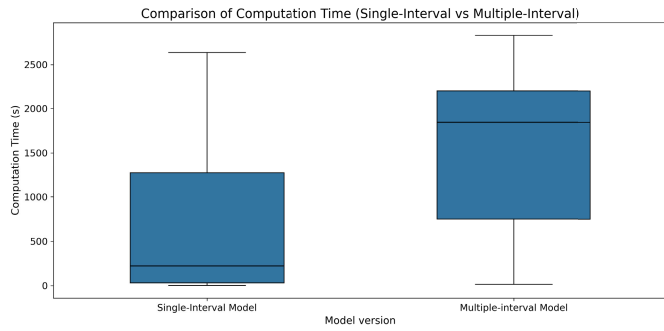


Fig. 15. Computation time comparison between the Single-Interval and Multiple-Interval models.

is noteworthy given the high computational efficiency of the model, which averaged just 0.97 seconds per mitigation instance.

- **Comparative Performance of Single- and Multiple-Interval models.** A direct comparison between the Single-Interval and Multiple-Interval models was performed over 100 simulation runs, with a maximum solver time of 200 seconds per mitigation instance. The objective was to assess the trade-off between the theoretical completeness of the Multiple-Interval model and its computational cost. The results showed that the Multiple-Interval model provided only a marginal improvement in solution quality, with the average total interval length being negligibly higher than that of the Single-Interval model (416.54s vs. 415.21s). Furthermore, the model found more than one interval for a mission in only 1.7% of cases on average. In stark contrast, the computational cost is substantially higher for the Multiple-Interval model. As shown in Fig. 15, the median computation time increases from 219.38 s (Single-Interval) to 1844.86 s (Multiple-Interval), corresponding to a 740.95% increase (median slow-down ratio of 6.94 $\times$). This indicates that the additional theoretical completeness of the Multiple-Interval formulation comes at a prohibitive computational burden in the tested high-density setting.

These findings highlight a clear trade-off between solution optimality and computational efficiency. The substantial increase in computational burden for the Multiple-Interval model is not justified by the minimal gains in flexibility. Therefore, the Single-Interval model represents a more practical and effective approach for real-time operational scenarios, offering a robust balance of increased acceptance rates, enhanced flexibility, and computational tractability.

Interpretation and scope. The reported evaluation is conducted in a corridor-based scenario with limited lateral route alternatives, under a trajectory-preserving mitigation policy implemented through temporal allocation. Consequently, the observed performance should be interpreted as deconfliction capability within a primarily scheduling-driven decision space. In airspace networks offering multiple route alternatives, extending the framework to a joint routing-and-scheduling formulation could further enlarge the feasible set and may yield additional system-level improvements; such extensions are beyond the present scope.

VI. CONCLUSION

This research set out to answer a very practical question that arises in U-space operations: What is the most effective way to rescue FPs that are initially rejected by the strategic deconfliction service, while keeping both computational cost and operator freedom in check? To this end, three optimization models were designed, implemented and systematically compared inside the DronAS platform.

A. Model Performance and Trade-Offs

A comparison of the models reveals a critical trade-off between operational flexibility and computational complexity. The Single-Interval model demonstrates a clear advantage over the baseline Single-Adjustment model, providing a 69.7% increase in the total duration of available take-off time and thereby offering significantly more flexibility to DOs.

In contrast, the Multiple-Interval model, while theoretically capable of providing the most flexibility by identifying multiple disjoint launch windows per mission, is severely hampered by practical performance limitations. The model's complexity increases quadratically $O(n^2)$ with the number of potential intervals, leading to a combinatorial explosion in decision variables and constraints. For example, a scenario with 20 rejected missions and 1000 conflicts would require 9000 constraints for the Multiple-Interval model (with q_i equal to 3 intervals) versus 1000 for the Single-Interval model. In simulations, this resulted in computation times that were often orders of magnitude higher, frequently failing to produce a solution within practical time limits.

Ultimately, the Single-Interval model emerges as the most effective and practical solution. It provides a substantial, measurable increase in both mission acceptance rates and operator flexibility at a low computational cost, with an average solution time of under one second in tested scenarios.

B. Connection to Multi-Agent Negotiation Framework

The introduction of a multi-agent negotiation framework is motivated by the structural heterogeneity of DOs and the coordination demands arising in dense VLL airspace. DOs differ in service-level commitments, operating costs, and risk attitudes; modeling them as heterogeneous agents with tailored reward functions provides a principled vehicle to encode these divergent preferences and strategic behaviors. The Solution Table (ST) constructed from the Single-Interval model exposes a compact, verifiable action space for negotiation: each candidate ARLW is conflict-free by construction, enabling preference aggregation over validated options; auction-based clearing can then reconcile individual valuations with safety and capacity constraints to yield explainable, auditable allocations that better utilize shared airspace while respecting regulatory provisions and DO diversity [35].

The structured ST compacts model outputs into decision-ready options. For every previously rejected FP, the ST records a conflict-free ARLW together with rejected FP and DO identifiers. Rows enumerate globally feasible solutions that jointly satisfy forbidden-interval constraints, while columns

associate entries to the corresponding DO; blank cells indicate the absence of a valid ARLW for that rejected FP in a given solution. The AM publishes a filtered ST to each DO that contains only its admissible options embedded within globally ST, thereby preserving commercial confidentiality and traceability.

On top of ST, a multi-round sealed-bid auction is executed in a competitive environment with heterogeneous DO agents. Each agent selects a preferred solution from its view and submits a bid limited by virtual Priority Coins (P-coins). P-coins are debited only when a DO wins at the last round, which supports exploration across rounds and mitigates early budget exhaustion. The AM clears one executable solution per episode, broadcasts outcomes as feedback, and the process iterates until a round cap or natural convergence (e.g., a single active bidder).

Executing the multi-agent negotiation framework converts rejections into re-accommodations without altering trajectories, raising acceptance ratio and temporal flexibility while maintaining safety guarantees. Because every selected option is drawn from the pre-validated ST, the resulting allocation is explainable and auditable, aligning DO diversity with regulatory provisions. Empirical results show that the Single-Interval backbone offers a practical balance - low computational cost with measurable gains in utilization - while the negotiation layer improves re-accommodation rates and reconciles competition with network efficiency in dense VLL airspace.

C. Conclusion and Future Work

This study successfully achieved its core objectives of enhancing airspace utilization, automating the search for alternative requested launch windows, and providing greater flexibility for FP planning. The Single-Interval model, in particular, proved to be a robust and efficient tool, making it the recommended approach for most real-time operational scenarios.

Future research will proceed along three primary vectors. First, efforts will be made to optimize the Multiple-Interval model by reducing its complexity, for instance, by leveraging more powerful computational hardware. Second, work will continue on the seamless integration of the solution table with the multi-agent negotiation framework, refining the data exchange to support a cohesive decision-making ecosystem. Finally, the developed models will be fully encapsulated within the operational U-space platform, DronAS. This involves not only embedding the algorithms but also expanding the platform's capabilities, to ensure the system is robust and adaptable for real-world deployment.

ACKNOWLEDGMENT

Opinions expressed in this article reflect only the author's views.

REFERENCES

- [1] International Civil Aviation Organization. (2022). *UAS Toolkit—Advanced UAS Operations in Urban Environments*. [Online]. Available: <https://www.icao.int/safety/UA/Pages/UA-Toolkit.aspx>
- [2] Federal Aviation Administration. (2021). *Urban Air Mobility Concept of Operations V1.0*. [Online]. Available: <https://nari.arc.nasa.gov/sites/default/files/attachments/UAMConOpsv1.0.pdf>
- [3] CORUS-XUAM. (2020). *Concept of Operations for European U-Space Services—Extension for Urban Air Mobility*. [Online]. Available: <https://doi.org/10.3030/101017682>
- [4] D. P. Thippavong et al. (2018). *Urban Air Mobility Airspace Integration Concepts and Considerations*. NASA. [Online]. Available: <https://ntrs.nasa.gov/api/citations/20180005218/downloads/20180005218.pdf>
- [5] SESAR Joint Undertaking. (2017). *U-Space Blueprint*. [Online]. Available: <https://www.sesarju.eu/u-space-blueprint>
- [6] Single European Sky ATM Research 3 Joint Undertaking. (2023). *U-Space Concept of Operations (ConOps)—Fourth Edition*. SESAR 3 Joint Undertaking, Luxembourg. [Online]. Available: <https://data.europa.eu/doi/10.2829/207917>
- [7] EASA. (2020). *EASA Opinion, no. 01/2020: High-Level Regulatory Framework for the U-Space*. [Online]. Available: <https://www.easa.europa.eu/en/documentlibrary/opinions/opinion-012020>
- [8] European Commission. (2022). *BUBBLES: Defining the Building Basic BLocks for a U-Space SEparation Management Service*. Accessed: Jul. 16, 2025. [Online]. Available: <https://cordis.europa.eu/project/id/893206>
- [9] C. C. Puchol, N. V. Vélaz, J. V. B. Tejedor, and J. A. V. Carbó, "BUBBLES: A new concept of operations for separation management in the U-space," *J. Phys., Conf. Ser.*, vol. 2526, no. 1, Jun. 2023, Art. no. 012092, doi: [10.1088/1742-6596/2526/1/012092](https://doi.org/10.1088/1742-6596/2526/1/012092).
- [10] European Commission. (2018). *PARTAKE: Cooperative DepArture Res for a Competitive ATM Network Service*. Accessed: Jul. 16, 2025. [Online]. Available: <https://cordis.europa.eu/project/id/699307>
- [11] European Union. (Apr. 2021). *Commission Implementing Regulation (EU) 2021/664 of 22 April 2021 on a Regulatory Framework for the U-Space*. [Online]. Available: <https://eur-lex.europa.eu/eli/regimpl/2021/664/oj>
- [12] European Union Aviation Safety Agency. (2024). *Easy Access Rules for Unmanned Aircraft Systems (Regulations (EU) 2019/947 and 2019/945) Including U-Space*. [Online]. Available: <https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-unmanned-aircraft-systems-regulations-eu>
- [13] Z. Liu, J. L. Munoz-Gamarra, and J. J. Ramos Gonzalez, "Characterization of strategic deconflicting service impact on very low-level airspace capacity," *Drones*, vol. 8, no. 9, p. 426, Aug. 2024. [Online]. Available: <https://www.mdpi.com/2504-446X/8/9/426>
- [14] J. J. R. González, Z. Liu, J. L. Muñoz-Gamarra, E. Pastor, and C. Barrado, "Impact assessment of the strategic planning performance in shared U-space volumes," *Eur. J. Transp. Infrastruct. Res.*, vol. 25, no. 1, pp. 67–85, Jan. 2025. [Online]. Available: <https://journals.open.tudelft.nl/ejtir/article/view/7472>
- [15] A. Geser and C. Munoz, "A geometric approach to strategic conflict detection and resolution [ATC]," in *Proc. 21st Digit. Avionics Syst. Conf.*, vol. 1, 2002, pp. 6B1-1–6B1-11.
- [16] 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, doi: [10.1287/opre.46.3.406](https://doi.org/10.1287/opre.46.3.406).
- [17] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Review of conflict resolution methods for manned and unmanned aviation," *Aerospace*, vol. 7, no. 6, p. 79, Jun. 2020. [Online]. Available: <https://www.mdpi.com/2226-4310/7/6/79>
- [18] C. A. Badea, A. Vidosavljević, J. Ellerbroek, and J. Hoekstra, "Evaluating the synergy of conflict detection and resolution services for constrained urban airspace," *IEEE Open J. Intell. Transp. Syst.*, vol. 6, pp. 24–36, 2025.
- [19] J. Munoz-Gamarra, J. Ramos, Z. Liu, and A. Sobejano, *Impact of USSPs Performance in Shared U-Space Volumes (SESAR Innovation Days)*, SESAR Joint Undertaking, Brussels, Belgium, 2023.
- [20] N. Scheffers, J. J. R. González, P. Folch, and J. L. Munoz-Gamarra, "A constraint programming model with time uncertainty for cooperative flight departures," *Transp. Res. C, Emerg. Technol.*, vol. 96, pp. 170–191, Nov. 2018.
- [21] N. Scheffers, M. A. A. Carmona, J. J. R. González, F. S. Nieto, P. Folch, and J. L. Munoz-Gamarra, "STAM-based methodology to prevent concurrence events in a multi-airport system (MAS)," *Transp. Res. C, Emerg. Technol.*, vol. 110, pp. 186–208, Jan. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X18313408>
- [22] DACUS. (2020). *Demand and Capacity Optimisation in U-Space*. [Online]. Available: <https://doi.org/10.3030/893864>
- [23] Y. Tang and Y. Xu, "Incorporating optimization in strategic conflict resolution for UAS traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 12393–12405, Nov. 2023.

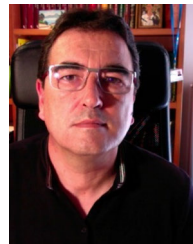
- [24] F. Ho et al., "Pre-flight conflict detection and resolution for UAV integration in shared airspace: Sendai 2030 model case," *IEEE Access*, vol. 7, pp. 170226–170237, 2019.
- [25] C. Cebi, E. Atac, and O. K. Sahingoz, "Job shop scheduling problem and solution algorithms: A review," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–7.
- [26] R. M. E. S. de Oliveira and M. S. F. O. D. C. Ribeiro, "Comparing mixed & integer programming vs. Constraint programming by solving job-shop scheduling problems," *Independ. J. Manage. Prod.*, vol. 6, no. 1, pp. 211–238, Mar. 2015.
- [27] A. Fekih et al., "Mixed-integer programming and constraint programming models for the flexible job shop scheduling problem," in *Artificial Intelligence and Industrial Applications*, vol. 771. Cham, Switzerland: Springer, 2023, pp. 110–122.
- [28] J. N. Hooker, "Chapter 15—Operations research methods in constraint programming," in *Handbook of Constraint Programming* (Foundations of Artificial Intelligence), vol. 2, F. Rossi, P. van Beek, and T. Walsh, Eds., Amsterdam, The Netherlands: Elsevier, 2006, ch. vol. 15, pp. 527–570. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1574652606800192>
- [29] W. T. Lunardi, E. G. Birgin, P. Laborie, D. P. Ronconi, and H. Voos, "Mixed integer linear programming and constraint programming models for the Online printing shop scheduling problem," *Comput. Oper. Res.*, vol. 123, Nov. 2020, Art. no. 105020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054820301374>
- [30] N. Nikolic, I. Grujicic, and N. Mladenovic, "A large neighbourhood search heuristic for covering designs," *IMA J. Manage. Math.*, vol. 27, no. 1, pp. 89–106, Jan. 2016.
- [31] F. Zhang, X. Cao, and D. Yang, "Intelligent scheduling of public traffic vehicles based on a hybrid genetic algorithm," *Tsinghua Sci. Technol.*, vol. 13, no. 5, pp. 625–631, Oct. 2008.
- [32] Google Developers. (2024). *Job Shop Scheduling Problem*. Accessed: Aug. 14, 2024. [Online]. Available: <https://developers.google.com/optimization/scheduling/jobshop>
- [33] C. Leet and R. A. Morris. (2514). *Combinatorial Auction-Based Strategic Deconfliction of Federated UTM Airspace*. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2024-4454>
- [34] B. Pang, K. H. Low, and C. Lv, "Adaptive conflict resolution for multi-UAV 4D routes optimization using stochastic fractal search algorithm," *Transp. Res. C, Emerg. Technol.*, vol. 139, Jun. 2022, Art. no. 103666.
- [35] Z. Liu, J. J. R. Gonzalez, J. Luis, and M. Gamarra, "A multi-agent negotiation framework for planning conflict-free U-space scenarios," in *Proc. ICRAT*, 2024. [Online]. Available: <https://www.icrat.org/previous-conferences/11th-international-conference/papers/>
- [36] J. L. Munoz-Gamarra, J. J. Ramos, and Z. Liu, "U-space contingency management based on enhanced mission description," *Aerospace*, vol. 11, no. 11, p. 876, Oct. 2024. [Online]. Available: <https://www.mdpi.com/2226-4310/11/11/876>
- [37] 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.
- [38] B. Pang, K. H. Low, and V. N. Duong, "Chance-constrained UAM traffic flow optimization with fast disruption recovery under uncertain waypoint occupancy time," *Transp. Res. C, Emerg. Technol.*, vol. 161, Apr. 2024, Art. no. 104547. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X24000688>
- [39] Keleno. (2020). *OR-Tools: A Simple Guide to Google Optimization Tools*. [Online]. Available: <https://medium.com/keleno/or-tools-fc7f9a6f59b0>
- [40] W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible job-shop scheduling via graph neural network and deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1600–1610, Feb. 2023.
- [41] T. O. Pocola et al., "Optimal sizing and control of a grid-connected battery in a stacked revenue model including an energy community," *Appl. Energy*, vol. 397, Nov. 2025, Art. no. 126122.
- [42] Y. Xu, M. Fang, L. Chen, G. Xu, Y. Du, and C. Zhang, "Reinforcement learning with multiple relational attention for solving vehicle routing problems," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 11107–11120, Oct. 2022.
- [43] Google. (2024). *Google Optimization: Mixed Integer Programming (MIP)*. Accessed: Jul. 23, 2025. [Online]. Available: <https://developers.google.com/optimization/mip>



Zhiqiang Liu is currently pursuing the Ph.D. degree in electronic and telecommunication engineering with the Universitat Autònoma de Barcelona (UAB), Spain. His research focuses on U-space/UTM, multi-agent negotiation, deep reinforcement learning, and very-low-level (VLL) airspace capacity optimization. His recent work develops and evaluates a deep reinforcement learning-based multi-agent negotiation framework that re-accommodates previously rejected flight plans via iterative sealed-bid auctions. His research interests include learning-enabled auction mechanisms, robust optimization, and intelligent air transportation systems, aiming to deliver scalable, regulation-aligned decision support for conflict-free U-space operations.



Zihang Zhang received the B.Sc. degree in computational mathematics from Zhejiang University, China, in 2020, and the joint M.Sc. degree in interdisciplinary mathematics from the InterMaths Erasmus Mundus Program in 2024. As part of the program, he studied at the University of L'Aquila, Italy; TU Wien, Austria; and the Universitat Autònoma de Barcelona, Spain, focusing on logistics and decision-making. His master's research involved the development of multi-model optimization approaches for strategic planning in low-altitude airspace, emphasizing trajectory deconfliction and scheduling. His research interests include optimization, applied machine learning, and intelligent decision systems.



Juan José Ramos Gonzalez received the Ph.D. degree from the Universitat Autònoma de Barcelona (UAB) in 2003. He is currently a Professor with the Department of Telecommunication and System Engineering, UAB. He is also the Program Director of European Master in Logistics and Supply Chain Management, UAB. He is also a Researcher, focusing on modeling, simulation, and optimization of dynamic systems, especially in the field of logistics. He is an Expert in air transportation, production technologies/logistics, intelligent transport systems, industrial collaboration, and technology transfer. Over the past ten years (2014–2024), he has actively participated in several national and European research initiatives: four projects funded by Spanish Ministry of Science and Innovation (MICINN), serving as a Principal Investigator (PI) in the two most recent, and eight European projects under the H2020 and Horizon Europe (HE) frameworks, serving as a Principal Investigator (PI) in the two most recent.



Jose Luis Munoz-Gamarra received the Ph.D. degree in electronic engineering from the Autonomous University of Barcelona in 2014. He is currently a Researcher with the Department of Telecommunication and Systems, Autonomous University of Barcelona. He is a member of the UAV Systems Laboratory, where he specializes in the development of advanced mission planning services, with a particular focus on dynamic capacity balancing (DCB), strategic deconfliction, and contingency planning. His research lies at the intersection of airspace management, optimization, and autonomy, aiming to support scalable and resilient drone operations. He is also actively involved in the design of novel airspace structures and in defining the regulatory and operational requirements necessary for the full-scale deployment of urban air mobility (UAM) systems.