From Rejection to Re-Accommodation: A Generate-and-Negotiate Framework for Conflict-Free U-space

Zhiqiang Liu, Juan José Ramos Gonzalez, Jose Luis Munoz-Gamarra*
Technical Innovation Cluster on Aeronautical Management
Universitat Autonoma de Barcelona
Sabadell, Barcelona, Spain
Corresponding Email: *JoseLuis.Munoz.Gamarra@uab.cat

***Abstract*—High-density very low level operations frequently lead to flight plan rejections under strategic deconfliction services that preserve trajectories and adjust the take-off time window. This study addresses the problem through alternative generate based on a Single-Interval alternative solving model: for each previously rejected flight plan, a continuous conflict-free alternative requested launch window is computed within a bounded range, allowing drone operators to select feasible departures without trajectory changes while maintain the safety requirements. The method relies on constraint-based scheduling with separation, capacity, and constraints, and optimizes either acceptance or temporal flexibility subject to operational limits. Results on representative urban-corridor scenarios indicate substantial reductions in rejections and large gains in available flexibility at low computational cost. The outputs are packaged into a global solution table that aggregates globally feasible combinations of the alternative requested launch window to ensure that subsequent selections remain conflict-free. A multi-agent negotiation framework can consume the global solution table to clear among solutions using sealed-bid. Overall, Single-Interval alternative solving model coupled with deep reinforcement learning based multi-agent negotiation framework converts rejections into explainable, auditable re-accommodations while remaining compatible with current strategic deconfliction workflows.**

***Keywords*—U-space, Airspace Capacity Optimization, Strategic Deconfliction, Constraint Programming, UTM.**

I. Introduction

The European U-space framework establishes the regulatory and service ecosystem that enables safe, efficient, and secure Very Low Level (VLL) operations for diverse Unmanned Aerial Vehicles (UAVs) across urban and suburban environments [1], [2]. It is implemented by certified U-space Service Providers (USSPs) delivering pre-flight and in-flight services under the oversight of competent authorities, with information exchange coordinated via a Common Information Service Provider (CISP); in a given U-space volume, when there is only one USSP existing, it may be designated to orchestrate strategic planning and deconfliction in accordance with the service set defined in the latest concept of operations and EU regulation [3]–[5]. Collectively, these roles define the baseline; building on it, this study proposes that the USSP actively manage VLL airspace capacity by computing and generating conflict-free alternatives and facilitating operator negotiation to resolve conflicts and re-accommodate demand [5], [6].

To compute these alternatives, strategic deconfliction techniques rooted in Air Traffic Management (ATM) and operations-research scheduling especially Constraint Programming (CP) are employed. Early work on free-flight airspaces formulated Conflict Detection and Resolution (CD&R) as a pairwise geometric task [7], before shifting toward system-level optimization to address high demand [7], [8]. In the U-space setting, a key algorithmic advance is the grid-to-interval reduction that transforms the four dimensional (4D: latitude, longitude, altitude and time) trajectory conflicts into one dimensional (1D) forbidden intervals, enabling standard scheduling solvers and powering several SESAR-JU prototypes [9]–[11]. Finally, analyses comparing First-Come-First-Served (FCFS) and Batch submissions underscore sizable, unexploited headroom in VLL capacity [12], [13]. These methods yield feasible options; allocating them among heterogeneous operators is, however, a negotiation problem.

Multi-Agent Systems (MAS) address the allocation problem via decentralized coordination: autonomous agents negotiate over scarce airspace resources, often through auctions, with demonstrated benefits in air traffic flow and related mobility domains [10], [14]. In parallel, Deep Reinforcement Learning (DRL) has become state of the art for adaptive decision-making, with landmark results driving applications across robotics, finance, and transportation [15], [16]. For strategic deconfliction, policy-gradient DRL - especially PPO's [17] actor-critic [18] with a clipped objective - offers stable learning under competitive interactions.

In dense VLL airspace, even Batch policy mitigations that evaluate groups of requests inevitably reject many Flight Plans (FPs), leaving capacity on the table and forcing Drone Operators (DOs) into trial-and-error resubmissions [12], [13]. This paper addresses the problem with an alternative-generate stage that follows the baseline planner: rather than dropping a rejected FP, compute and generate conflict-free alternatives that can be offered back to the relavent DO [19].

The paper makes three contributions. First, the Single-Interval alternative solving model is introduced which maxi-

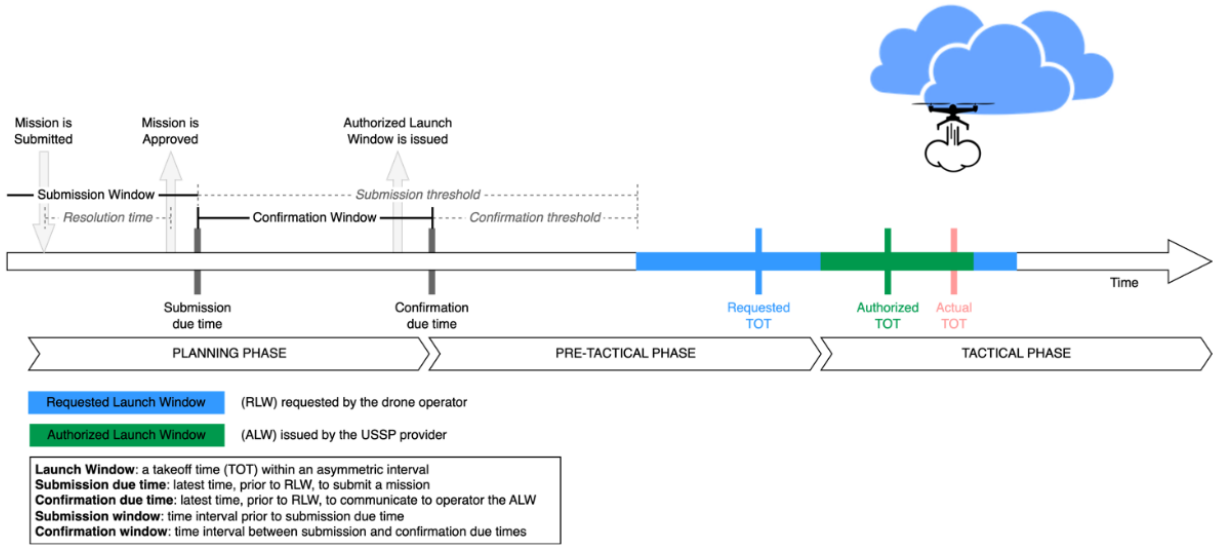


Figure 1. Strategic deconfliction service.

mizes temporal flexibility by producing a continuous Alternative Requested Launch Window (ARLW) for each rejected FP, thereby improving service quality and operational robustness with sub-second solves. Second, all ARLWs is packaged into a global solution table that the USSP can issue per DO. The global solution table lists, for each previously rejected FP, one or more conflict-free ARLW and aggregates them into globally feasible "bundles" for Batch execution. Third, a deep reinforcement learning based multi-agent negotiation framework is initiated and validated to respect the DO's opinion to the executed solution through a sealed bidding process.

The remainder of this paper is organized as follows: Section II details the strategic deconfliction service. Section III introduce the methodology for the advanced optimization for strategic deconfliction and rescheduling. Section IV presents the formal definitions of the proposed multi-agent negotiation framework. Finally, Section V concludes the paper with a summary of findings and an outline for future research.

II. Strategic Deconfliction Service

Within the U-space regulatory framework, the Strategic Deconfliction Service (SDS) is the pre-flight function that authorizes a candidate FP only if its planned operation can safely coexist with all previously approved FPs in the same airspace volume. The USSP - certified by the competent authority - uses common information disseminated by the CISP to compare the candidate FP against the accepted FP set and predict any loss of separation minima. When the estimated probability of infringement exceeds a predefined threshold, the SDS is triggered to generate mitigation actions [6].

The SDS adopts a time-based separation policy as the primary mitigation mechanism: geometric trajectories requested by DOs are preserved while conflicts are removed by shifting take-off times. This preserves operator intent and reduces trajectory complexity risks. In practice, the DronAS platform (a USSP platform capable of running simulations, varying parameters, and assessing the impact of parameter changes on strategic deconfliction performance) coordinates mutually compatible take-off time windows using optimization algorithms originally developed in the ATM domain [9]–[11], resolving multi-FP interactions while maintaining safety targets.

Figure 1 delineates the operational timeline, each FP submitted to the USSP contains a 4D trajectory and a Requested Launch Window (RLW), a time interval centered around the Requested Take-Off Time (RTOT). If the SDS finds a conflict-free slot within the RLW, the USSP issues an Authorized Launch Window (ALW), defined as a sub-interval of the RLW. Otherwise, the FP is rejected. The submission must occur within the Submission Window (SW), and the USSP may refine feasible ALWs during the Confirmation Window (CW) - the interval between the end of SW and the start of the RLW - before communicating the final ALW by the Confirmation Due Time (CDT). Once an ALW is issued, the pre-tactical phase begins; the tactical phase commences at ALW start and the DO must depart within the ALW to guarantee strategic conflict-free execution.

Operationally, different admission policies can underpin the SDS. A FCFS policy authorizes FPs in arrival order, providing transparency but yielding limited capacity in dense demand. Batch policy considers sets of FPs simultaneously and typically assigns more ALWs within the same RLW portfolio. Empirical results in representative scenarios show average acceptance ratios around 41.5% for FCFS and 49.8% for Batch, illustrating the capacity gains of simultaneous optimization while preserving safety constraints [13], [19].

Despite these improvements, a residual approve/reject binary outcome remains a limitation in high-density VLL operations: some FPs are inevitably rejected, which can underutilize available airspace and provides little guidance to DOs.

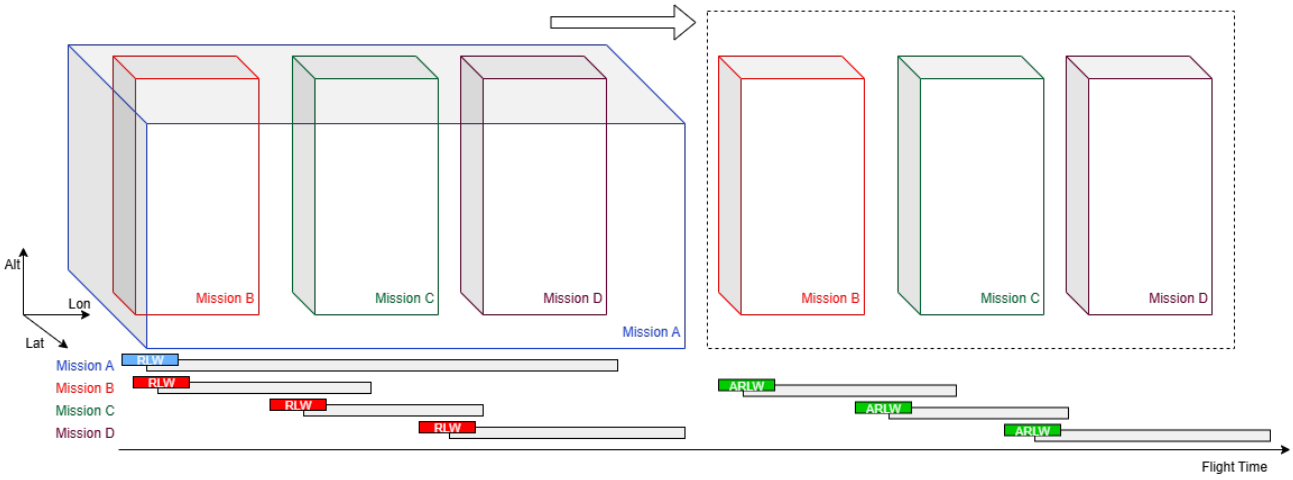


Figure 2. Mitigation for finding alternatives.

This gap motivates complementary, post-SDS rescheduling strategies, but these mechanisms build atop the SDS baseline described above and do not alter its pre-flight safety assurance role.

III. Advanced Optimization for Strategic Deconfliction and Rescheduling

This section introduces a Single-Interval alternative solving model that extends time-based SDS by producing, for each rejected FP, a contiguous interval of admissible take-off time adjustments rather than a simple reject. The approach preserves the DO's originally requested geometric trajectories and focuses mitigation exclusively on time shifts, thereby minimizing operational disruption while maintaining required separation minima.

A. Spatiotemporal Interactions

A spatiotemporal interaction occurs when trajectories intersect geometrically and their usage of that intersection overlaps in time. In left side of Fig. 2, Mission A intersects the paths of B, C, and D, and their declared durations overlap - so A interacts with each of B/C/D. By contrast, B, C, and D only overlap in time (their paths do not intersect), hence no interaction exists among them.

If Mission A's request is processed first, it is approved and the others are rejected, because no shift of A's ALW within its RLW can remove all conflicts. Acceptance ratio: 25%. If any of B/C/D arrives first, those three can all be accepted while A is rejected, yielding 75%. Under a Batch policy that maximizes approvals, the same set (75%) is achievable.

In the right side of Fig. 2, if the Batch mitigation initially approves Mission A and rejects B, C, and D. After extending the original RLW (dashed square), ARLWs can be found for the previously rejected missions. However, even with an extended RLW, some FPs may still remain infeasible.

B. Modeling Launch Windows and Shift Variables

For each mission i, the Do specifies a RTOT and a RLW. The authorization process returns an ALW that is a sub-interval of the RLW. Any actual take-off time chosen within the ALW is admissible. Throughout, launch-time flexibility is encoded by a scalar shift d_i with respect to the RTOT. Without loss of generality, the ALW can be normalized to $[0, u]$ (length u); thus

$$\mathrm{ATOT}_i = \mathrm{RTOT}_i + d_i, \qquad d_i \in [0, u]. \tag{1}$$

(The more general case $[\ell, u]$ follows by constant time translation and is omitted for brevity.) In pairwise discussions below, the requested take-off times are denoted t_1 and t_2, with shifts d_1 and d_2 constrained by their respective RLWs:

$$t_1 + d_1 \in \mathrm{RLW}_1, \qquad t_2 + d_2 \in \mathrm{RLW}_2. \tag{2}$$

C. From 4D conflicts to 1D forbidden intervals.

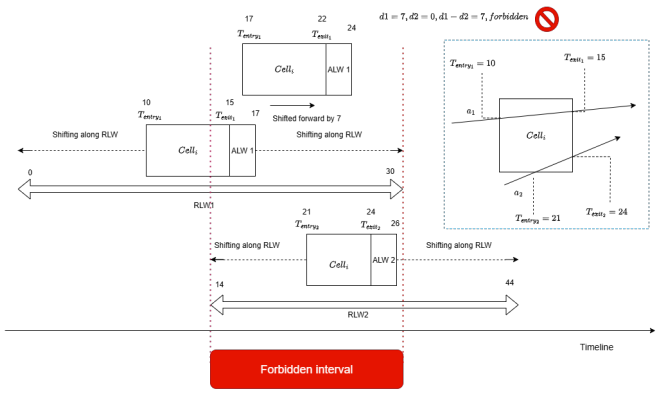
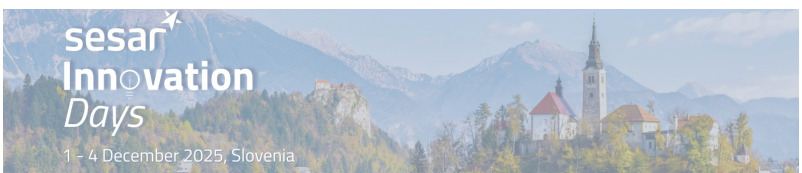


Figure 3. Forbidden interval.

As shows in Figure 3, consider two flights a_1 and a_2 traversing a discretized cell c of the voxelized airspace. Let $(c^{t_e}_{a_k}, c^{t_s}_{a_k})$ be the nominal (entry, exit) times of a_k in c. With



time shifts $d_k \in \mathrm{ALW}_k \subseteq \mathrm{RLW}_k$, the actual entry/exit times are

$$t^{a_k}_{e,\mathrm{act}} = c^{t_e}_{a_k} + d_k, \qquad t^{a_k}_{s,\mathrm{act}} = c^{t_s}_{a_k} + d_k. \tag{3}$$

A spatio-temporal conflict in c occurs when the adjusted occupancy intervals overlap:

$$[c^{t_e}_{a_1} + d_1,\ c^{t_s}_{a_1} + d_1] \cap [c^{t_e}_{a_2} + d_2,\ c^{t_s}_{a_2} + d_2] \neq \emptyset. \tag{4}$$

This is equivalent to the relative shift $\Delta = d_1 - d_2$ lying in the forbidden interval

$$[s, e] \ = \ \left[\, c^{t_e}_{a_2} - c^{t_s}_{a_1}\ ,\ c^{t_s}_{a_2} - c^{t_e}_{a_1} \,\right]. \tag{5}$$

Hence, deconfliction requires that feasible shifts respect both ALWs and avoid the forbidden interval:

$$d_1 \in \mathrm{ALW}_1, \quad d_2 \in \mathrm{ALW}_2, \quad d_1 - d_2 \notin [s, e]. \tag{6}$$

Operationally, the detection phase computes and aggregates all such forbidden intervals over all relevant cells; the mitigation phase then enforces these 1D constraints.

D. Optimization Background

With the forbidden interval abstraction, the mitigation process, under a Batch policy aiming to maximize the count of approved missions, can be formulated as a mathematical optimization problem. The model is presented below:

1) Entities:

$$Missions: \quad a_1, \ldots, a_n \tag{7}$$
$$Conflicts: \quad c_1, \ldots, c_m \tag{8}$$

- 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:

$$dvar_i = (d_i, p_i), \quad \text{for } i \in \{1, \ldots, n\} \tag{9}$$

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

3) Constraints:

$$\forall c : (a_i, a_j, [s, e]) \quad ((p_i = 1 \wedge p_j = 1) \Rightarrow (d_i - d_j < s \vee d_i - d_j > e)) \tag{10}$$

4) Optimization goal:

$$\max \sum_i p_i \tag{11}$$

In this formulation, the decision variables are each mission's time shift, which determines its ALW, and a binary presence variable. Setting $p_i = 0$ rejects mission a_i; $p_i = 1$ admits it. For every detected conflict $(a_i, a_j, [s, e])$, if both missions are present, their time-shift difference must avoid the forbidden interval:

$$p_i = p_j = 1 \ \Rightarrow \ d_i - d_j \notin [s, e].$$

The objective is to maximize the number of approved missions.

E. Single Interval Alternative Solving Model

The Single-Interval alternative solving model generalizes pointwise adjustments into decision intervals. For each mission i, the admissible take-off shift d_i is allowed to vary within $[s_i, e_i]$ (rather than being fixed at a single value), while preserving the requested trajectory and all precomputed forbidden interval constraints. The admissible take-off time is therefore

$$\mathrm{ATOT}_i = \mathrm{RTOT}_i + d_i, \qquad d_i \in [s_i, e_i] \ \subseteq \ \mathrm{RLW}_i. \tag{12}$$

1) Entities:

$$Missions: \quad a_1, a_2, \ldots, a_n \tag{13}$$
$$Conflicts: \quad c_1, c_2, \ldots, c_m \tag{14}$$
$$Regulated \quad Missions: \quad R \tag{15}$$

- n: number of missions; m: number of conflicts.
- R: indices of missions already approved (immutable), $i \in \{1, \ldots, n\}$.
- Each conflict c is represented as a tuple $(a_i, a_j, [s, e])$, where $[s, e]$ is the forbidden interval for the pair (i, j).

2) Decision variables:

$$dvar_i = (s_i, e_i, p_i), \quad i \in \{1, \ldots, n\} \tag{16}$$

- 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:

$$\forall i \in \{1, \ldots, n\} : (e_i \geq s_i) \tag{17}$$
$$i \in R \Rightarrow (p_i = 1 \wedge s_i = e_i = d_i) \tag{18}$$
$$\forall c : (a_i, a_j, [s, e]), \quad ((p_i = 1 \wedge p_j = 1) \Rightarrow (e_i - s_j < s \vee s_i - e_j > e)) \tag{19}$$
$$\sum p_i \geq P \tag{20}$$

4) Optimization goal:

$$\max \sum_i (e_i - s_i) \tag{21}$$

The constraints are formulated as follows:

- Interval Validity (Equation 17): This constraint requires that, for every adjustment interval, the end time is not earlier than the start time, i.e., $e_i \geq s_i$.
- Regulated Missions (Equation 18): This constraint locks the intervals of missions already approved, preserving their assigned schedules without modification.
- Interval Deconfliction (Equation 19): This constraint captures the key feature of the model. For intervals $[s_i, e_i]$ and $[s_j, e_j]$ associated with missions i and j, respectively, the full set of possible temporal differences is $[s_i - e_j,\ e_i - s_j]$. This derived interval must be disjoint from the predefined forbidden interval of the pair (i, j).
- Acceptance Rate Consistency (Equation 20): This constraint preserves the approval level by enforcing that

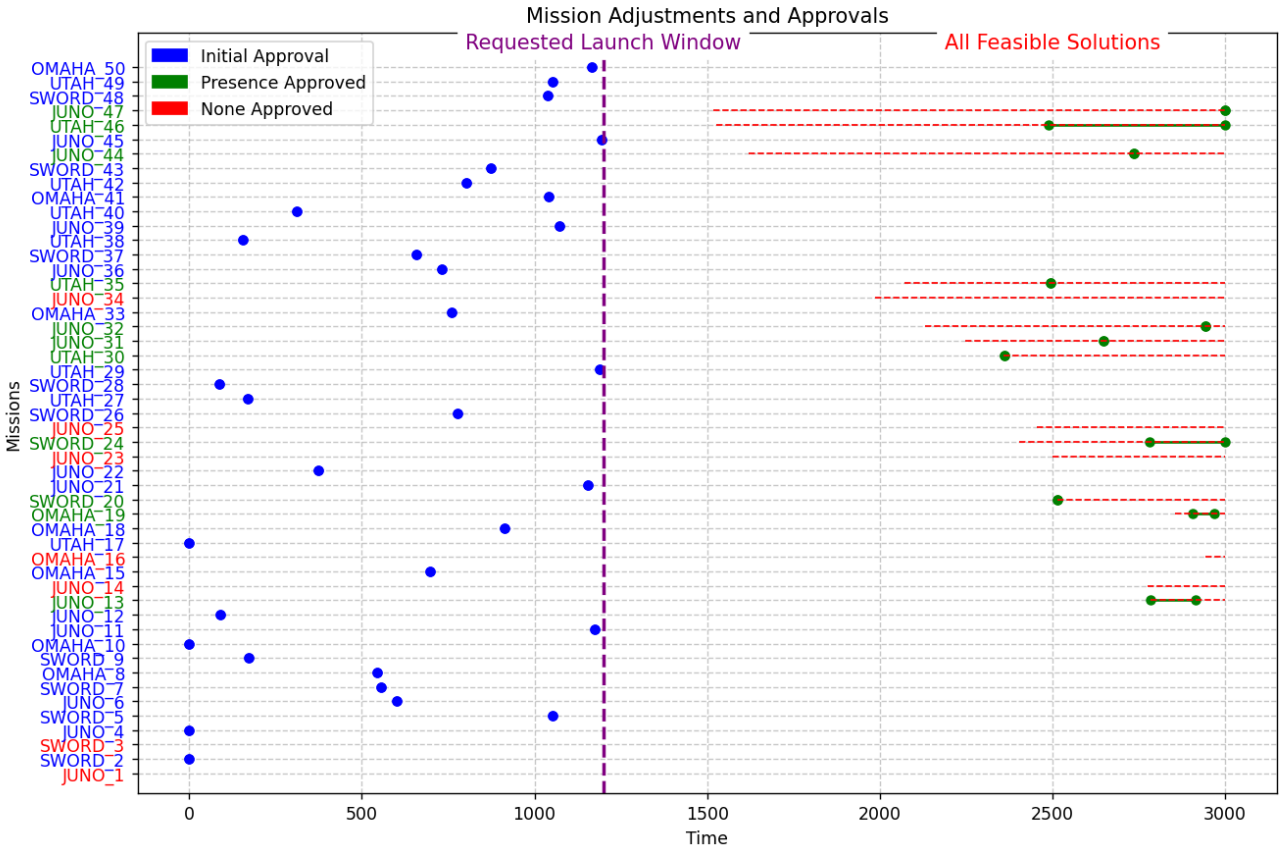


Figure 4. A visualized result from the Single-Interval alternative solving model.

the total number of accepted missions equals P, where P is the maximum achievable count determined in the preceding optimization stage.

F. Optimization Tools

This study adopts Google OR-Tools [20] and, specifically, the CP-SAT solver as the optimization backbone. OR-Tools offers a stable open-source ecosystem with high performance (C++ core, multi-language bindings) and has been validated across canonical optimization tasks in CP, MILP, and VRP [20]–[23]. The alternative-searching problem features mixed decision variables (integer time-shift variables and Boolean mission-presence variables). Prior guidance indicates that, for typical mixed problems involving both integer and Boolean variables there is often no significant runtime advantage between classical MIP and CP-SAT approaches, making either viable [24]. Given OR-Tools' CP-SAT job-shop scheduling example [25] and seamless integration needs, this work adopts OR-Tools' CP-SAT to solve the alternative-searching model.

G. Simulation and Result

The study draws on the CORUS-XUAM trials in Castelldefels (Barcelona), which demonstrated that U-space can oversee logistics-UAS in controlled urban and suburban airspace. The setup emulated a mesh of vertiports - Points of Departure (PODs) - used by four operators for last-mile deliveries. Each POD acted as a hub, transferring consignments from ground transport to multirotors that served thirteen delivery points within a 3 km radius [3], [4]. Traffic moved through four parallel corridors - two westbound and two eastbound - separated vertically at 30/70 m Above Ground Level (AGL, westbound) and 40/80 m AGL (eastbound), with a lateral offset to enable safe crossings and to absorb unplanned climbs without affecting upper levels.

For simulation, synthetic traffic sets were generated with randomly assigned RTOT, and outcomes were estimated via Monte Carlo sampling. Each run selected 50 missions with RTOTs uniformly spread over 30 minutes. Every mission received a 20-minute RLW starting at its RTOT. Missions rejected in the initial mitigation step were then checked within an ARLW: a 50-minute window that also begins at the original RTOT.

The Single-Interval alternative solving outputs, for each initially rejected mission, an admissible take-off time interval $[s_i, e_i]$. When $s_i = e_i$, the recommendation collapses to a single time.

In the experiment, the minimum number of second-round approvals among initially rejected missions was constrained to 11 (Eq. 20). As shown in Fig. 4, mission labels are color-coded as follows: blue for missions approved in the initial

stage, green for missions initially rejected but approved in the alternative-generate stage, and red for missions that remain rejected. Several alternatives appear as intervals (endpoints marked by green dots) instead of single points. The initial stage mitigation approved 32 of 50 missions; among the remaining 18, exactly 11 were accepted in the alternative-generate phase, meeting the constraint. Notably, 4 of these 11 (JUNO-13, OMAHA-19, SWORD-24, UTAH-46) received nonzero-length ARLWs, and the sum of all interval lengths across the 11 approvals totaled 920 s. Compared with the initial stage acceptance ratio of 64% (32/50), the alternative-generate stage adds 22 percentage points, raising overall acceptance to 86% (43/50).

IV. Deep Reinforcement Learning Based Multi-Agent Negotiation Framework

The proposed multi-agent negotiation framework is motivated by heterogeneous DOs and the coordination needs of dense VLL airspace. DOs differ in service levels, operating costs, and risk attitudes; modeling them as heterogeneous agents with tailored reward functions captures these divergent preferences and strategic behaviors. The Global Solution Table (GST) derived from the Single-Interval alternative solving model defines a compact, verifiable action space for negotiation. DRL based multi-agent negotiation framework clearing can then align individual valuations with safety and capacity constraints, producing explainable and auditable allocations that improve airspace utilization while respecting regulatory requirements and DO diversity.

A. Global Solution Table

The GST compresses the Single-Interval alternative solving model outputs into negotiation-ready bundles. For each previously Rejected Flight Plan (RFP), the GST logs a conflict-free ARLW together with the RFP and DO identifiers. As Table I shows, rows correspond to jointly feasible solutions that satisfy all forbidden interval constraints across missions; columns group entries by DO. Empty cells mark that, in that solution, no admissible ARLW exists for the corresponding RFP.

TABLE I. Example of Global Solution Table

	DO-A		DO-B	
RFPs	**RFP1**	**RFP2**	**RFP3**	**RFP4**
Solution 1	ARLW1,1	-	ARLW1,3	ARLW1,4
Solution 2	-	ARLW2,2	-	-
Solution 3	ARLW3,1	ARLW3,2	-	ARLW3,4
Solution 4	-	-	ARLW4,3	-

To balance confidentiality with accountability, the USSP provides each DO with a filtered view of the GST that exposes only its own admissible ARLWs and the corresponding solution indices, while hiding all competing FP details. This redaction preserves commercial sensitivity and fair competition, yet maintains an auditable trail via the shared solution references. Since every GST entry is pre-validated for separation, downstream preference aggregation can proceed directly on these solutions without re-running the SDS process.

The observation can be made from Table I, DO-B prefers solution 1 as it reassigns both RFPs - RFP3 and RFP4, while DO-A favours solution 3 since it accommodates two RFPs instead of one in solution 1. This creates strong competition, making winning the negotiation crucial for each DO.

B. Multi-Agent Negotiation Framework

Building on the GST, a multi-round sealed-bid auction runs in a competitive setting with heterogeneous DO agents. From their filtered views, agents choose a preferred solution and submit bids subject to a budget of virtual Priority Coins (P-coins). P-coins do not correspond to real money or any external economic value. Instead, they function purely as internal tokens that quantify each agent's bidding capacity, and deducted only for winners at the final round, encouraging exploration across rounds and preventing premature budget depletion. The USSP clears one executable solution per episode, publishes the outcomes as feedback, and repeats the process until a preset round limit is reached or the system converges (e.g., only one active bidder) [19].

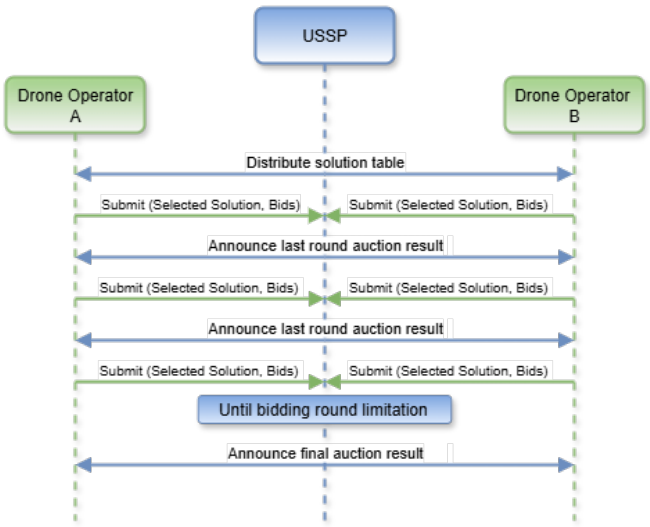


Figure 5. The process of negotiation framework.

As illustrated in Fig. 5, the auction drives a cyclical negotiation loop. After each clearing event, a fresh round opens in which DOs reassess their tactics in light of previous outcomes. Agents may revise bid levels - reflecting remaining P-coins and observed competitive pressure - or switch to a different solution if earlier attempts failed. Rounds repeat until the preset cap is met, or terminate early if a single bidder remains. At the end of the sequence, the USSP publishes the definitive results, thereby fixing the executable solution. DOs that secured their preferred solutions proceed accordingly; others operate under the allocations assigned to their RFPs within the finalized solution.

Figure 6 depicts the closed-loop coupling between the DO agents and the USSP - auction environment within the deep

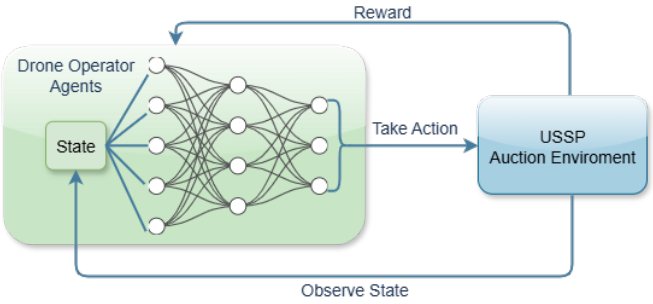


Figure 6. The structure of deep reinforcement learning.

reinforcement learning scope following the Markov Decision Process (MDP) [26]. At each bidding round, every agent encodes the current auction context into a state vector, processes it through its policy network, and produces an action (a bid paired with a selected solution). The USSP evaluates the joint action profile, updates the auction book, and returns feedback (reward): an immediate scalar reward for the round, the next state vector, and at episode end, both global and agent-specific rewards. This observe-act-feedback routine repeats until the predefined number of bidding rounds has been completed.

Four archetypal agent roles are considered that capture distinct bidding strategies and priorities in the negotiation process:

- Achiever bids aggressively to maximize mission success, often winning slots at high P-coin cost.
- Economizer conserves resources, bidding sparingly and only when returns justify expenditure.
- Optimizer balances acceptance probability and cost, adapting bids to secure reliable outcomes without overspending.
- Normalizer provides a neutral baseline with measured, pattern-driven bids, neither overpaying nor disengaging, serving as a stable reference for comparison.

C. Simulation and Result

After training for 300 episodes, the learned policies were evaluated on 100 competitive validation auctions constructed from GST disjoint from the training set.

Figure 7 shows the resulting win rates where a win denotes that an agent secured execution of its most preferred solution in the final auction round. The Achiever class tops the table with 51 wins (51%), followed by the Optimizer with 35 (35%). In contrast, the more budget-conscious Normalizer and Economizer secure 9 (9%) and 5 (5%) victories, respectively. This split mirrors the mission-yield analysis: aggressive coin spending (Achiever, Optimizer) reliably converts into higher clearing scores, whereas risk-averse bidding (Economizer, Normalizer) seldom attains first place. Even so, their combined 14% share indicates that low-budget strategies can prevail when congestion or price spikes temper the high-spenders.

V. Conclusion

This paper introduced a generate-and-negotiate framework that converts strategic rejections into actionable, conflict-free re-accommodations without altering operator trajectories. A Single-Interval alternative solving model computes, for each previously rejected mission, a continuous ARLW within bounded limits. These ARLWs are assembled into a global solution table that exposes only feasibility-checked, jointly executable solutions and supports a sealed-bid, multi-agent negotiation aligned with heterogeneous DO preferences and budgets.

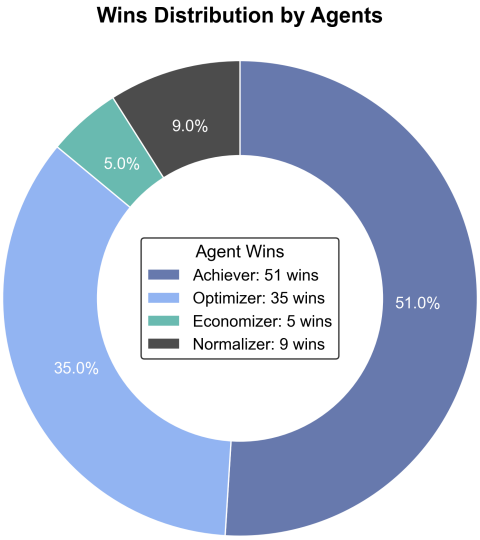


Figure 7. Validation result: wins distribution

Across representative urban-corridor scenarios, overall acceptance in a 50-mission instance rose from 64% to 86% with sub-second solves, and several re-accommodated flights received nonzero-length ARLWs, returning useful flexibility. The framework integrates as a post-strategic deconfliction service add-on: ingest accepted and rejected requests, run the Single-Interval alternative solving model within predefined bounds, publish a filtered-view GST per DO, and clear one executable solution via sealed bids. This placement preserves SDS safety assurance, ensures traceability through common solution identifiers, and supports deployment in USSP-operated environments with standard observability.

Negotiation occurs over pre-validated solutions, shrinking the action space to safe, globally feasible choices. This yields allocative efficiency, computational efficiency, and operational stability. Learned agent policies showed role-consistent behavior - aggressive bidders winning more often, with budget-conservative policies prevailing under congestion - indicating that value signaling through bids can reconcile competition with system efficiency.

The current stack deliberately trades flexibility for simplicity: the Single-Interval model preserves operator-requested geometries and mitigates conflicts only via time shifts within RLW/ALW/ARLW. This keeps the forbidden-interval decisions explainable and auditable, but leaves capacity untapped in corridors where bounded speed trims, single-step flight-level swaps, or short lateral nudges would clear otherwise blocking

interactions. On the market side, sealed-bid P-coin auctions over a pre-validated GST reduce tactical complexity yet do not fully neutralise bid shading, or systematic disadvantage for DOs. The experiments assume a single USSP, deterministic inputs, and moderate instance sizes; at city scale or in multi-provider settings, CP-SAT throughput, GST size, confidentiality constraints, and shared audit semantics can become bottlenecks.

A pragmatic roadmap is to augment CP-SAT mitigation with a tightly bounded playbook - within $\pm(5\text{-}10)\%$ speed control, a one-step flight-level change where corridor rules allow, and micro re-routes capped by a small lateral/length budget - activated only when it strictly dominates a time-shift-only solution to preserve explainability. The auction layer can be hardened with privacy-preserving audit logs, bid caps/floors and per-round spend limits, plus a small reserved quota or discount budget for late/small entrants. For scale, in multi-USSP environments, exchange only minimal interoperability profiles using secure aggregation or differential privacy, and agree on common audit semantics so negotiated outcomes remain explainable across organizational boundaries.

Overall, the pipeline elevates the USSP from passive authorizer to active capacity manager: computing and generating conflict-free alternatives and facilitating drone operator negotiation so strategic rejections are re-accommodated without trajectory changes, with explainable, auditable outcomes.

Acknowledgment

This research is part of the national Spanish project: "Improving the planning and management of flight contingencies in the scalable deployment of drone operations" (Grant PID2023-152053OB-C22 funded by MCIN/AEI/10.13039/501100011033). Opinions expressed in this article reflect only the author's views.

References

[1] SESAR Joint Undertaking, "U-space blueprint," 2017. [Online]. Available: https://www.sesarju.eu/u-space-blueprint

[2] European Union Aviation Safety Agency, "Easy access rules for unmanned aircraft systems (regulations (eu) 2019/947 and 2019/945) including u-space," https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-unmanned-aircraft-systems-regulations-eu, 2024.

[3] CORUS-XUAM, "Concept of operations for european u-space services - extension for urban air mobility," 2020. [Online]. Available: https://doi.org/10.3030/101017682

[4] Single European Sky ATM Research 3 Joint Undertaking, "U-space concept of operations (conops) — fourth edition," SESAR 3 Joint Undertaking, Luxembourg, Tech. Rep., 2023. [Online]. Available: https://data.europa.eu/doi/10.2829/207917

[5] European Union, "Commission implementing regulation (eu) 2021/664 of 22 april 2021 on a regulatory framework for the u-space," https://eur-lex.europa.eu/eli/reg_impl/2021/664/oj, 2021, official Journal of the European Union, L 139, 23 Apr 2021, pp. 161-183.

[6] EASA, "Easa opinion no 01/2020: High-level regulatory framework for the u-space," 2020. [Online]. Available: https://www.easa.europa.eu/en/documentlibrary/opinions/opinion-012020

[7] A. Geser and C. Munoz, "A geometric approach to strategic conflict detection and resolution [atc]," in *Proceedings. The 21st Digital Avionics Systems Conference*, vol. 1, 2002, pp. 6B1–6B1.

[8] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operations Research*, vol. 46, no. 3, pp. 406–422, 1998. [Online]. Available: https://doi.org/10.1287/opre.46.3.406

[9] N. Schefers, J. J. R. González, P. Folch, and J. L. Munoz-Gamarra, "A constraint programming model with time uncertainty for cooperative flight departures," *Transportation Research Part C: Emerging Technologies*, vol. 96, 2018.

[10] N. Schefers, M. A. Amaro Carmona, J. J. Ramos González, F. Saez Nieto, P. Folch, and J. L. Munoz-Gamarra, "Stam-based methodology to prevent concurrence events in a multi-airport system (mas)," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 186–208, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18313408

[11] European Commission, "PARTAKE: cooPerative depArtuRes for a compeTitive ATM networK sErvice," https://cordis.europa.eu/project/id/699307, 2018, grant Agreement ID 699307, Horizon 2020 project, accessed 16 Jul 2025.

[12] 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, 2024. [Online]. Available: https://www.mdpi.com/2504-446X/8/9/426

[13] J. J. Ramos González, Z. Liu, J. L. Muñoz-Gamarra, E. Pastor, and C. Barrado, "Impact assessment of the strategic planning performance in shared u-space volumes," *European Journal of Transport and Infrastructure Research*, vol. 25, no. 1, Jan. 2025. [Online]. Available: https://journals.open.tudelft.nl/ejtir/article/view/7472

[14] B. Chen, H. H. Cheng, and J. Palen, "Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 1, pp. 1–10, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X08000314

[15] W. Zhang, H. Liu, F. Wang, T. Xu, H. Xin, D. Dou, and H. Xiong, "Intelligent electric vehicle charging recommendation based on multi-agent reinforcement learning," in *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*. Association for Computing Machinery, Inc, 4 2021, pp. 1856–1867.

[16] K. Li, L. Niu, Y. Yang, and Y. Wang, "An approach to optimize lab-seat allocation problem based on multi-agent negotiation," in *IEIR 2022 - IEEE International Conference on Intelligent Education and Intelligent Research*. Institute of Electrical and Electronics Engineers Inc., 2022, pp. 247–250.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 7 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[18] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, 2000.

[19] Z. Liu, J. J. R. Gonzalez, J. Luis, and M. Gamarra, "A multi-agent negotiation framework for planning conflict-free u-space scenarios," in *ICRAT*, 2024.

[20] Keleno, "Or-tools: A simple guide to google optimization tools," https://medium.com/keleno/or-tools-fc7f9a6f59b0, 2020.

[21] W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible job-shop scheduling via graph neural network and deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1600–1610, 2023.

[22] T. O. Pocola, V. Robu, J. Rietveld, S. Norbu, B. Couraud, M. Andoni, D. Flynn, and H. V. Poor, "Optimal sizing and control of a grid-connected battery in a stacked revenue model including an energy community," *Applied Energy*, vol. 397, p. 126122, 2025.

[23] 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 Transactions on Cybernetics*, vol. 52, no. 10, pp. 11 107–11 120, 2022.

[24] Google, "Google optimization: Mixed integer programming (MIP)," 2024, accessed: 2025-07-23. [Online]. Available: https://developers.google.com/optimization/mip

[25] Google Developers, "Job shop scheduling problem," https://developers.google.com/optimization/scheduling/job_shop, 2024, accessed 14 Aug 2024.

[26] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, 1999.