Integrated Conflict Management for UAM with Strategic Demand Capacity Balancing and Learning-based Tactical Deconfliction

Shulu Chen, Antony Evans, Marc Brittain and Peng Wei

***Abstract*—Urban air mobility (UAM) has the potential to revolutionize our daily transportation, offering rapid and efficient deliveries of passengers and cargo between dedicated locations within and around the urban environment. Before the commercialization and adoption of this emerging transportation mode, however, aviation safety must be guaranteed, i.e., all the aircraft have to be safely separated by strategic and tactical deconfliction. Reinforcement learning has demonstrated effectiveness in the tactical deconfliction of en route commercial air traffic in simulation. However, its performance is found to be dependent on the traffic density. In this project, we propose a novel framework that combines demand capacity balancing (DCB) for strategic conflict management and reinforcement learning for tactical separation. By using DCB to precondition traffic to proper density levels, we show that reinforcement learning can achieve much better performance for tactical safety separation. Our results also indicate that this DCB preconditioning can allow target levels of safety to be met that are otherwise impossible. In addition, combining strategic DCB with reinforcement learning for tactical separation can meet these safety levels while achieving greater operational efficiency than alternative solutions.**

***Index Terms*—Safety, Separation Assurance, Demand Capacity Balancing, Multi-agent Reinforcement Learning**

I. Introduction

A. Motivation

According to projections, the number of air vehicles operating in urban areas will experience a significant increase in the next two decades [1]–[3]. One major part of this forecasted traffic surge is from electric vertical take-off and landing (eVTOL) cargo and passenger air taxis in Urban Air Mobility (UAM) operations. The current Air Traffic Control (ATC) system is heavily human-based, which is not expected to support the emerging high-density urban air traffic operations [4]. Automation tools and autonomous agents to manage the urban airspace and UAM traffic are required. Autonomous ATC was proposed in 2005 with the introduction of the NASA Advanced Airspace Concept (AAC) [5]. This rule-based autonomous ATC tool was further developed and validated over the following 10 years to augment human ATC, increase traffic capacity and enhance operation safety [6], [7].

Specifically for UAM, the US Federal Aviation Administration (FAA) and National Aeronautics and Space Administration (NASA) proposed concepts for Unmanned Aircraft System (UAS) Traffic Management (UTM) in recent years [8]–[11]. From these proposals, one of the most challenging requirements for an autonomous ATC system is to mitigate conflicts in high-density traffic flows. This can be achieved through a combination of strategic conflict management, which is used to resolve predicted conflicts prior to departure by adding a ground delay or rescheduling another flight route, and tactical deconfliction, which focuses on real-time decision making for airborne aircraft separation through maneuver advisories like speed or heading changes.

Various autonomous conflict management systems have been developed, but one persistent challenge in the integration of such systems is to ensure the advisories are coordinated to achieve the desired safety level. If this is not the case, the strategic and tactical deconfliction methods may affect each other's results and introduce new risks. To address this issue, we propose an integrated conflict management framework (ICMF) that combines both strategic conflict management and tactical deconfliction. By implementing this comprehensive autonomous system in air traffic management (ATM) for UAM, we seek to guarantee safety levels within target values, while also optimizing traffic efficiency.

B. Related Work

Strategic conflict management involves strategic decisions like ground delays made by air traffic managers to balance traffic demand with airspace capacity at bottlenecks, e.g. airport runways, merging points, and air route intersections. For traditional ATM, such an approach has been designed effectively and has shown measurable improvements for airlines in the National Airspace System (NAS). For example, Traffic Management Initiatives (TMI) such as the Ground Delay Program (GDP), Airspace Flow Program (AFP), and the Collaborative Trajectory Options Program (CTOP) are tools used by air traffic flow managers to balance demand with capacity in congested regions [12]. These programs have resulted in reduced delays and cancellations for airlines operating in the NAS, while also improving safety levels by reducing the number of aircraft in the airspace and preventing potential conflicts. However, strategic conflict management for UAM is still a challenge because of the high-density traffic and high-population areas over which that traffic operates

S. Chen is with the Department of Electrical and Computer Engineering, George Washington University, Washington, DC 20052 shulu@gwu.edu

A. Evans is the Director of System Design for Airbus UTM at Acubed, an Airbus innovation center, Sunnyvale, CA 94086 tony.evans@airbus-sv.com

M. Brittain is a member of the Technical Staff at MIT Lincoln Laboratory, Lexington, MA 02421 marc.brittain@ll.mit.edu

P. Wei is with the Department of Mechanical and Aerospace Engineering, George Washington University, Washington, DC 20052 pwei@gwu.edu

[13]. Therefore, further research is required to study and analyze the effectiveness of strategic conflict management in the UAM setting, specifically with the integration of tactical deconfliction technologies.

The field of aircraft separation assurance has seen the introduction of many advanced methods, as highlighted by recent studies [14]. One such approach involves using Markov Decision Processes (MDP) to formulate the separation assurance problem by incorporating a probabilistic model that can handle uncertainties encountered during flight [15]. Offline MDP-based methods are useful for strategic deconfliction, while online MDP-based methods are more suitable for tactical deconfliction [16]–[18]. However, offline methods can become intractable if uncertainty occurs en route since the policy is designed ahead of time, and it is challenging for online methods to solve the problem efficiently [19]. To address these challenges, researchers have turned to deep reinforcement learning (DRL) for separation assurance problems [19]–[23]. For instance, the deep distributed multi-agent variable (D2MAV-A) framework incorporates an attention network and employs a modified Proximal Policy Optimization (PPO) algorithm to solve complex sequential decision-making problems with a variable number of agents [22]. Nevertheless, a key concern with DRL is its generalization ability - if the density of traffic flow exceeds the training environment, the DRL agent may provide erroneous advisories and lead to an aircraft conflict, or even a near mid-air collision. Thus, preconditioning air traffic to proper density levels using strategic conflict management is essential for DRL to ensure safe separation.

C. Contributions and Structure

The major contributions of this paper are summarized as follows:

1) **An integrated conflict management framework for UAM.** This new framework is a coordination between strategic conflict management and tactical deconfliction. Through our analysis, we demonstrate that by utilizing strategic conflict management methods, we can ensure a reliable foundation for effective tactical deconfliction for UAM. These complementary approaches work together to enhance the safety and efficiency of the UAM system.
2) **Game theory to improve MARL convergence rate.** This paper focuses on analyzing the potential safety threats posed by multiple aircraft operating in close proximity, such as when two aircraft merge together. Specifically, we investigate the instability and convergence issues that arise when training a multi-agent reinforcement learning (MARL) model. Through our analysis, we identify the reasons behind the model's instability and introduce a new policy to mitigate this issue using game theory. Our numerical results demonstrate a significant improvement over the previous model, highlighting the effectiveness of our proposed approach.
3) **A open-source UAM conflict mitigation sandbox.**[1] We have made the code base of our integrated conflict management simulation, which utilizes the BlueSky simulator, publicly available. Our code includes baseline methods and evaluation metrics, enabling users to easily assess the performance of their own strategic and tactical algorithms by replacing the existing ones. This open framework allows for continued development and testing of conflict management approaches in the context of UAM, ultimately improving the safety and efficiency of the system.
4) **Revealing essential insights into the interactions between strategic conflict management and tactical deconfliction.** In this paper, we demonstrate that strategic conflict management methods, such as departure separation and DCB, can effectively precondition tactical deconfliction and maintain safety metrics at nearly constant levels. In addition, tactical deconfliction methods improve traffic efficiency by permitting higher capacity near bottlenecks. However, the maneuvers employed by tactical deconfliction also result in demand uncertainty at each capacity constrained resource, which diminishes the effectiveness of DCB.

In Section II, the problem formulation and system framework are described. In Section III, we described the strategic conflict management methods, including departure separation and three different approaches for DCB. In Section IV, the multi-agent reinforcement learning separation method and a baseline method for tactical deconfliction are described. In Section V, five numerical experiments are described to demonstrate the effectiveness and interactions between strategic and tactical methods. Finally, we present conclusions in Section VI.

II. Problem Formulation

This paper aims to develop a system that ensures aviation safety metrics remain below target levels while optimizing traffic efficiency. To achieve this objective, we introduce an integrated conflict management platform (ICMP) for UAM, which integrates strategic and tactical separation methods to mitigate conflicts. As previously demonstrated in [13], a combination of strategic conflict management and tactical deconfliction is an effective approach for balancing safety and efficiency.

A. Framework for Integrated Conflict Management Platform

Figure 1 illustrates the ICMP framework, which divides the flight operation into two stages: pre-departure and airborne. The pre-departure stage utilizes strategic conflict management to determine an appropriate departure time by introducing ground delays. This paper presents one departure separation method and two demand capacity balancing algorithms to suit different scenarios, as outlined in Section III. The airborne stage employs tactical deconfliction methods to provide speed advisories for all aircraft to resolve conflicts. This includes a MARL-based separation assurance method and a rule-based separation algorithm, the latter representing a benchmark against which the performance of the other methods can be compared. These tactical deconfliction methods are described

[1] Code is available at https://github.com/Shulu-Chen/bluesky-DCB.git

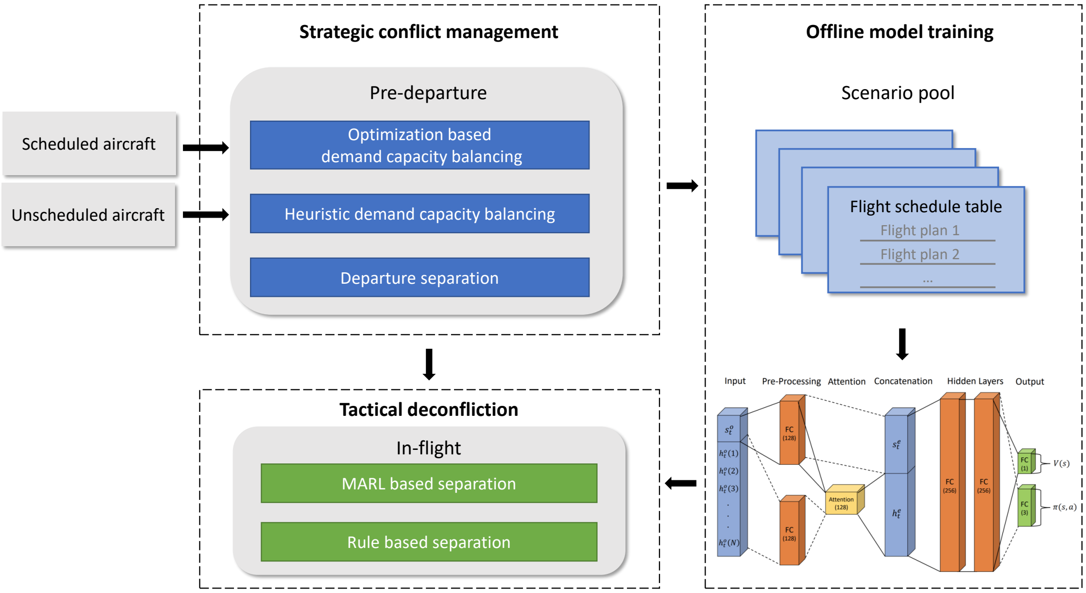


Fig. 1: The framework of integrated conflict management platform. In the sub-figure depicting the neural network, both s_t^o and h_t^o symbolize the original state information for the ownship and intruders, respectively. Meanwhile, s_t^e and h_t^e represent the encoded information derived after processing through the attention layer.

in Section IV. Additionally, strategic conflict management generates simulated flight plans for the MARL offline model training process, which is then used for online operations.

B. Safety Metrics

In this paper, four safety metrics are measured:

1) **Number of Loss of Well Clear (LoWC) events per flight hour.** A LoWC event is defined as a loss of horizontal separation between any aircraft, and the range is set as 500 meters in this paper under the recommendation of [24].
2) **Number of Near Mid Air Collisions (NMAC) per flight hour.** Since Mid Air Collisions (MACs) between aircraft are rare, a Near Mid Air Collision (NMAC) is defined which represents a precursor to a Mid Air Collision. For crewed aviation, an NMAC is typically defined as a loss of 500 feet (152 meters) of horizontal separation and 100 feet (30 meters) of vertical separation [25]. Since we simulate operations flying at a co-altitude, we define an NMAC as a loss of 150 meters of horizontal separation, as described in Table I.
3) **Estimated Number of Mid Air Collisions (MAC) per flight hour.** We define a Mid Air Collision as a loss of horizontal separation of 10 meters, which is representative of the wingspan or maximum horizontal dimension of a UAM aircraft. However, since actual MACs are infrequent, especially with advanced conflict management, we instead observe the number of NMACs, and use a conditional probability, $\mathbb{P}(\text{MAC}|\text{NMAC})$, to estimate the probability of MAC. It's worth noting that this paper does not model the effect of collision avoidance systems such as the Airborne Collision Avoidance System X (ACAS X), which provides vertical and horizontal maneuvers to avoid mid-air collisions [27], [28]. Instead, we utilize a $\mathbb{P}(\text{MAC})$ risk ratio β to compensate for the impact of airborne collision avoidance on the likelihood of a mid-air collision.
The ACAS X risk ratio β is defined as:

TABLE I: parameters for estimated MACs

Parameter	Value
MAC horizontal separation threshold	10m
NMAC horizontal separation threshold	150m
Number of simulation runs (unmitigated)	200
Total testing flight hours (unmitigated)	1000
ACAS X risk ratio β^*	0.005
$\mathbb{P}(\text{MAC}\|\text{NMAC})$	5.038×10^{-3}

[*] In this paper, we choose β as 0.005, which is calculated from [26] table 5, where the $P(\text{NMAC})$ without ACAS X = 3.01×10^{-3} and $P(\text{NMAC})$ with ACAS X vertical and speed advisories = 1.50×10^{-5}.

$$\beta = \frac{\mathbb{P}(\text{NMAC, with ACAS X})}{\mathbb{P}(\text{NMAC, without ACAS X})} \tag{1}$$

We estimate the number of MAC events $\mathcal{N}_{\text{MAC}}$ by:

$$\mathbb{E}(\mathcal{N}_{\text{MAC}}) = \mathbb{P}(\text{MAC}|\text{NMAC}) \cdot \beta \cdot \mathcal{N}_{\text{NMAC}} \tag{2}$$

where $\mathcal{N}_{\text{NMAC}}$ is the number of NMACs observed in the simulation without the implementation of ACAS X. The $\mathbb{P}(\text{MAC}|\text{NMAC})$ is obtained by using Monte Carlo simulation on a scenario without any intervention. Table

I presents each of the parameter values used in the estimation of the number of MAC events.

4) **Risk Ratio** The risk ratio is calculated as the ratio of the number of estimated MACs for the non-intervention scenario and the number of estimated MACs for the other methods applying conflict management.

C. Efficiency Metrics

We calculate three different efficiency metrics:

1) **Ground delay** due to strategic conflict management. If departure demand is sufficiently high that demand exceeds capacity for any constrained resources, DCB will calculate a new departure time for the aircraft that will prevent the demand from exceeding capacity. Ground delay is calculated as:

$$\text{ground delay} = \max\{0, (R_f - S_f)\} \tag{3}$$

where R_f is the required departure time of flight f and S_f is the original scheduled departure time.

2) **Airborne delay** due to tactical deconfliction. For each aircraft, we estimate total flying time T_f based on the distance and the aircraft cruise speed. During simulation, we implement tactical deconfliction and measure the actual flying time A_f. Airborne delay is calculated as follows:

$$\text{airborne delay} = \max\{0, (A_f - T_f)\} \tag{4}$$

3) **Number of alerts** is the total number of speed-change advisories requested by the tactical deconfliction methods. Operators generally seek to minimize the number of maneuvers in the air, which use increased energy and increase workload on pilots. Hence, the number of alerts is applied as an efficiency metric.

III. Strategic Conflict Management

DCB is a mechanism that has been identified by the Federal Aviation Administration (FAA) as potentially being required to support urban air mobility (UAM) operations as the number of operations increases [10]. DCB involves defining airspace capacity and managing demand strategically to prevent demand from exceeding capacity. This can help to balance efficiency and predictability in UAM operations, particularly when operational uncertainties are high. By using DCB, it is possible to manage the demand for constrained resources, such as airspace network intersection points, in a way that helps to ensure the smooth and safe operation of UAM vehicles.

Figure 2 shows a schematic diagram of the DCB algorithm. At each bottleneck (crossing or merge point), the time horizon is divided into multiple time windows, each with a fixed duration S. The capacity C of the resource defines the maximum number of flights that can fly through the resource in the same time window. The goal of DCB is to strategically control the throughput at each bottleneck by delaying operations on the ground.

This paper introduces two DCB algorithms with different applications. An optimization-based DCB algorithm is centralized and takes the scheduled departure time of all aircraft as input and calculates the optimal departure time required to minimize total delay in advance. In contrast, the heuristic DCB algorithm has no guarantee to minimize total departure delay but can be used to determine ground delays required to ensure that demand does not exceed capacity for unscheduled aircraft in real-time. While optimization-based DCB works well for scheduled demand, heuristic DCB is useful for inserting unscheduled demand into the calculated departure flow. Figure 3 provides an example of how DCB works across multiple resources.

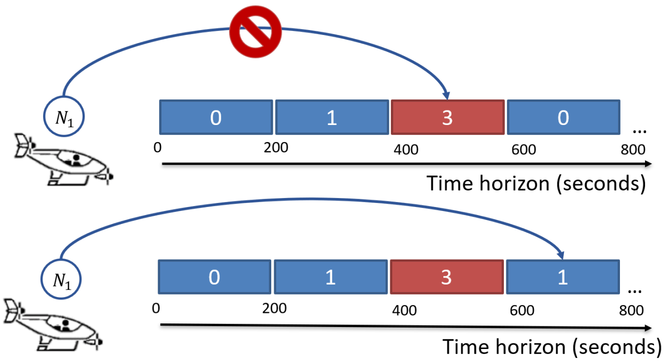


Fig. 2: Diagram of DCB. For the given bottleneck with a capacity of 3 operations every time window of 200 seconds, the aircraft's estimated arrival time falls within a fully occupied time window. To ensure the aircraft's arrival at the next available time window, the operation is assigned a ground delay.

A. Optimization Based Demand Capacity Balancing

In this problem, we formulated the DCB problem into a mix-integer programming problem, which can solve for networks with multiple capacity constrained resources. The formulation is shown below.

$$\min_{\boldsymbol{\omega}\in\mathbb{B}^+,\boldsymbol{R}\in\mathbb{R}^+} \sum_{d\in\mathcal{D}} \sum_{f\in\mathcal{F}^d} (R_{d,f} - S_{d,f}) \tag{5}$$

$$\text{s.t.} \quad R_{d,f+1} - R_{d,f} \geq \Delta, \qquad \forall d \in \mathcal{D}, f \in \mathcal{F}^d \tag{6}$$

$$R_{d,f} \geq S_{d,f}, \qquad \forall d \in \mathcal{D}, f \in \mathcal{F}^d \tag{7}$$

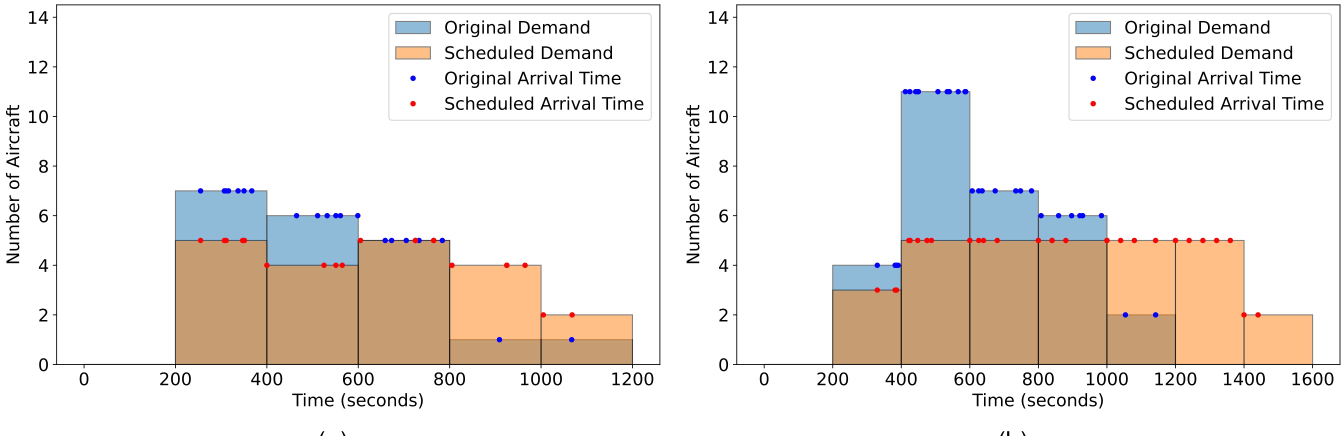
$$\sum_{n\in\mathcal{N}} \omega_{n,d,f,i} = 1, \quad \forall d \in \mathcal{D}, f \in \mathcal{F}^d, i \in \mathcal{I}^d \tag{8}$$

$$(R_{d,f} + T_{d,i} - B_n)\omega_{n,d,f,i} \geq 0, \tag{9}$$
$$\forall d \in \mathcal{D}, f \in \mathcal{F}^d, i \in \mathcal{I}^d, n \in \mathcal{N}$$

$$(R_{d,f} + T_{d,i} - B_n)\omega_{n,d,f,i} \leq W, \tag{10}$$
$$\forall d \in \mathcal{D}, f \in \mathcal{F}^d, i \in \mathcal{I}^d, n \in \mathcal{N}$$

$$\sum_{d\in\mathcal{D}} \sum_{f\in\mathcal{F}^d} \sum_{i\in\mathcal{I}^d, i=p} \omega_{n,d,f,i} \leq C^p, \tag{11}$$
$$\forall n \in \mathcal{N}, p \in \mathcal{P}$$

In this formulation, two decision variables are introduced: the time window identifier ω, and the required time of departure R. The objective (equation (5)) of the problem is to



(a)

(b)

Fig. 3: Example of how DCB can be applied across multiple resources. The blue bars show the original demand across different time windows, and the orange bars show the optimized traffic demand, while the dark orange part is the overlap between blue and orange bars. The blue and orange dots are the exact departure times of the modeled operations. (a) Traffic demand on resource 1. (b) Traffic demand on resource 2.

minimize the ground delay of all aircraft $f \in \mathcal{F}^d$ on all routes $d \in \mathcal{D}$. Here, $R_{s,d}$ is the required time of departure, and $S_{d,f}$ is the original scheduled departure time.

Constraint (6) ensures that any two aircraft departing from the same vertiport have a minimum separation of Δ. Constraint (7) ensures that the required departure time is not earlier than the scheduled time. Constraints (8)-(10) are used to identify the time window to which the estimated arrival time belongs. Here, $\omega_{n,d,f,i} = 1$ means that aircraft f departing from d will arrive at resource i during time window n. $R_{d,f} + T_{d,i} - B_n$ is the relative arrival time compared to time window n, where $T_{d,i}$ is the estimated flying time from d to i, B_n is the start time of time window n, and W is the length of the time window (set to 200 seconds in this paper). The identifier is activated as 1 only when the relative arrival time is within the interval $[0, W]$, and it can only be activated once. Finally, constraint (11) ensures that the number of aircraft at each resource $p \in \mathcal{P}$ does not exceed the capacity C^p of the resource. It is worth noting that resource set $\mathcal{I}^d$ includes only the resources involved in the route starting from d, while resource set $\mathcal{P}$ includes all the actual capacity constrained resources in the airspace.

B. Heuristic Demand Capacity Balancing

A single resource heuristic DCB algorithm is proposed in [13].

In our paper, we improved the algorithm to support networks with multiple resources. When the system receives new demand for the resource, it first checks the departure time of the leading aircraft. If the departure separation is within the required separation Δ, the system then uses a mapping function to check the remaining volume of the corresponding window. If the demand in the window reaches any of the involved resource capacities C_i, the following departure will be prevented from departing until the next window that is under the capacity limit appears. This algorithm is detailed in Algorithm 1.

Algorithm 1 Heuristic Demand Capacity Balancing

```
Collect initial DCB window list $\boldsymbol{\omega}$
Initialize start time $t$
while $t < T$:
    BlueSky.step()
    $t+ =$ SIMDT
    if received departure request from aircraft f at route r:
        check departure time of ahead aircraft $R_{r,f-1}$
        if $(R_{r,f} - R_{r,f-1}) \geq \Delta$:
            if $\boldsymbol{\omega}$.map$(t + D_i) < C_i$ for all bottlenecks:
                Release aircraft $f$
                $\boldsymbol{\omega}$.map$(t + D_i)+ = 1$
```

IV. Tactical Deconfliction

As introduced in [13], strategic deconfliction can mitigate conflicts and guarantee safe separation but at a significant cost to efficiency. To enhance safety and efficiency under uncertainty, airborne operations require tactical deconfliction, which provides maneuver advisories to resolve potential conflicts. In this paper, we introduce two tactical deconfliction methods, i.e., a learning-based method, and a rule-based method.

A. MARL Tactical Deconfliction

The multi-agent reinforcement learning (MARL) algorithm to control individual aircraft in a simulated air traffic environment is originally introduced in [20] and improved in [22]. By using MARL, the algorithm can adapt to changing conditions and learn from past experience, which can help to improve the performance of the system over time. Additionally, by training all of the agents using a shared model, all of the aircraft are following the same separation policy, which can help to prevent conflicts and maintain a safe and efficient flow of traffic. Overall, this approach combines the advantages of MARL and shared model training to provide a powerful tool for aircraft tactical deconfliction.

The tradeoff between exploration and exploitation is a well-recognized challenge in reinforcement learning. In our research, we adopt the synchronous variant of the asynchronous advantage actor-critic (A3C) algorithm, termed advantage actor-critic (A2C) [29], and integrate the loss function from proximal policy optimization (PPO) [30]. A2C, a policy-based approach, employs a unified neural network to estimate both the policy (actor) and value (critic) functions. By running multiple agent threads concurrently, A2C broadens the exploration of the state space. PPO, on the other hand, adeptly navigates the exploration-exploitation balance by implementing a novel loss function, ensuring policy updates remain proximal to the previous iteration. This minimizes the risk of excessively aggressive adjustments. Such a combined approach provides a thorough exploration of the action space. Simultaneously, it empowers the model to hone its strategy, emphasizing rewarding actions and enhancing the resilience and efficiency of our proposed framework.

We employed the deep distributed multi-agent variable-A (D2MAV-A) framework [22] as the foundation for our model training. The fundamental architecture of the neural network can be visualized in the sub-figure referenced by Figure 1. An attention layer is used to allow the arbitrary-length input and provide a fixed-length output for the later A2C algorithm.

The MARL components omitting the flight subscript f are listed below:

1) **State Space:** In reinforcement learning, the state space refers to the set of all possible states that an agent can encounter at a given time. In this particular study, we assume that the aircraft's state and dynamics information is fully accessible to others, like position, speed, and distance to the destination. Assessing the realism of this assumption is pivotal. Indeed, this assumption finds alignment with real-world scenarios where the sharing of information is a mandated requirement. Notable examples include the use of automatic dependent surveillance-broadcast (ADS-B) by commercial airlines [31], and the enforcement of remote identification for drones [32].

Specifically, the state space for each agent is formulated as follows:

$$s_t^o = \{d_{goal}^{(o)}, v^{(o)}, \theta^{(o)}, d_{\text{NMAC}}\} \tag{12}$$

$$h_t^o(i) = \{d_{goal}^{(i)}, v^{(i)}, \theta^{(i)}, d_o^{(i)}\} \tag{13}$$

where s_t^o represents the state of the ownship, which contains the distance to the goal $d_{goal}^{(o)}$, aircraft speed $v^{(o)}$, aircraft heading $\theta^{(o)}$, and the NMAC boundary d_{NMAC}. The state of the intruder is quite similar to the ownship while replacing the NMAC boundary with the distance between the ownship and the intruder $d_o^{(i)}$. All state values are continuous.

2) **Action Space:** In this study, the action space is defined as the set of possible actions that an aircraft can take at each decision-making step. These actions include slowing down, holding the current speed, or speeding up:

$$A_t = [-\Delta v, 0, +\Delta v] \tag{14}$$

Note that the action defined here is the maneuver operation sent to the simulator, not the exact speed change, so the action space is discrete, with a chosen $\Delta v = 5$ knots. At each time step, occurring every 4 seconds, the agent can execute maneuvers including accelerating by 5 knots, decelerating by 5 knots, or maintaining its current speed. Furthermore, we applied speed boundaries to the agent, setting the maximum speed at 70 knots and the minimum speed at 10 knots, based on the aircraft dynamics. Should the agent reach these speed limits, any additional actions beyond these boundaries would be rendered invalid.

3) **Reward Function:** A reward function in reinforcement learning can provide a scalar feedback signal to an agent, indicating the desirability of the state-action pair taken by the agent in an environment. In this paper, three types of penalties are considered at each time step:

$$R(s,t,a) = R(s) + R(t) + R(a) \tag{15}$$

Since maintaining separation is the primary objective in this paper, the majority of the reward function during the training process is allocated to the separation penalty term, denoted as $R(s)$. The separation penalty is defined as follows:

$$R(s) = \begin{cases} -1 & \text{if} \quad d_o^{(i)} < d_{\text{NMAC}} \\ -\alpha + \delta \cdot d_o^{(i)} & \text{if} \quad d_{\text{NMAC}} \leq d_o^{(i)} \leq d_{\text{LoWC}} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

If the distance between the ownship and the intruder is within the NMAC threshold d_{NMAC}, the agent incurs a penalty of -1 and is removed from the simulation. If the distance falls between the NMAC threshold and the LoWC threshold, the penalty is linearly proportional to the distance.

In this paper, we also take into account energy consumption, with the goal of optimizing traffic efficiency while maintaining a target level of safety. To realize this objective, we introduce a second component to the reward function, represented as R(t), which serves as a penalty for flying time.

$$R(t) = \begin{cases} -1 & \text{if} \quad t > T \\ -\eta & \text{otherwise} \end{cases} \tag{17}$$

Should an aircraft surpass its designated maximum flying time T without reaching its destination, it receives a penalty of -1 and is consequently withdrawn from the simulation. Surpassing the maximum allowable flying time can be equated to an in-flight loss of power, potentially resulting in a crash or necessitating an emergency landing—a situation comparable in consequence to a mid-air collision. Therefore, imposing the highest penalty in such instances is justified. In other circumstances, a constant penalty η is administered at each step and accumulates over time. This mechanism encourages the agent to prevent scenarios in which all aircraft maintain minimum speed until the simulation terminates, instead promoting increased speeds to avoid local optima.

In the real world, frequent speed changes can increase pilot workload (in the case of a piloted aircraft), with associated safety implications, and can also result in higher energy use.

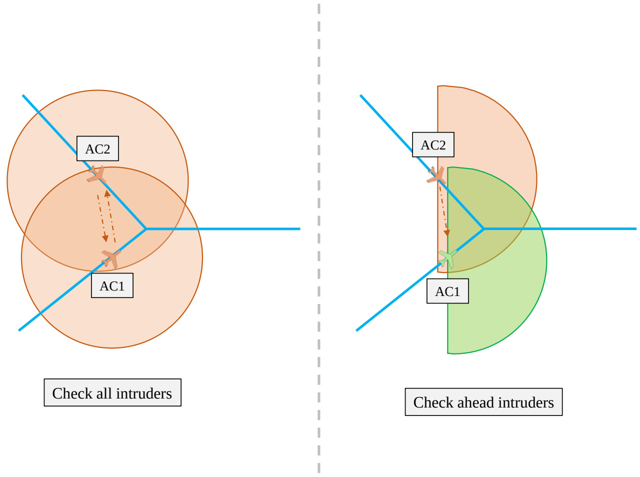


Fig. 4: Different intruder detection policies.

To mitigate these risks, we introduce the action penalty term $R(a)$

$$R(a) = \begin{cases} 0 & \text{if a = 0} \\ -\psi & \text{otherwise} \end{cases} \tag{18}$$

Whenever an aircraft changes its speed, a fixed penalty ψ is applied and accumulated over time. This penalty is intended to discourage unnecessary speed changes and encourage smoother flight paths.

The specific hyperparameters in the reward function for the finalized use-case are listed in Table II:

TABLE II: reward function hyperparameters

Hyperparameter	**Value**
NMAC threshold d_{NMAC}	150 meters
LoWC threshold d_{LoWC}	500 meters
Max flying time T	1800 seconds
Separation coefficient α	0.1
Separation coefficient δ	0.0002
Flying time coefficient η	0.001
Speed change coefficient ψ	0.0001

B. Implementation of Game Theory

To gain a comprehensive understanding of multi-agent decision-making problems and enhance the efficacy of tactical deconfliction methods, it is valuable to analyze the relationships among agents. However, solving a detailed multi-stage decision-making problem for all the aircraft from start to end becomes challenging when applying game theory. The equilibrium is hard to reach because of the computation complexity and inefficiency. To break down the problem, we focus on a one-step decision-making scenario between two merging aircraft, as illustrated in Figure 4.

TABLE III: cost table

		Aircraft 2		
		Speed up	Hold	Slow down
Aircraft 1	Speed up	-1.0, -1.0	-0.5, -0.5	0.0, -0.1
	Hold	-0.5, -0.5	-1.0, -1.0	-0.5, -0.5
	Slow down	-0.1, 0.0	-0.5, -0.5	-1.0, -1.0

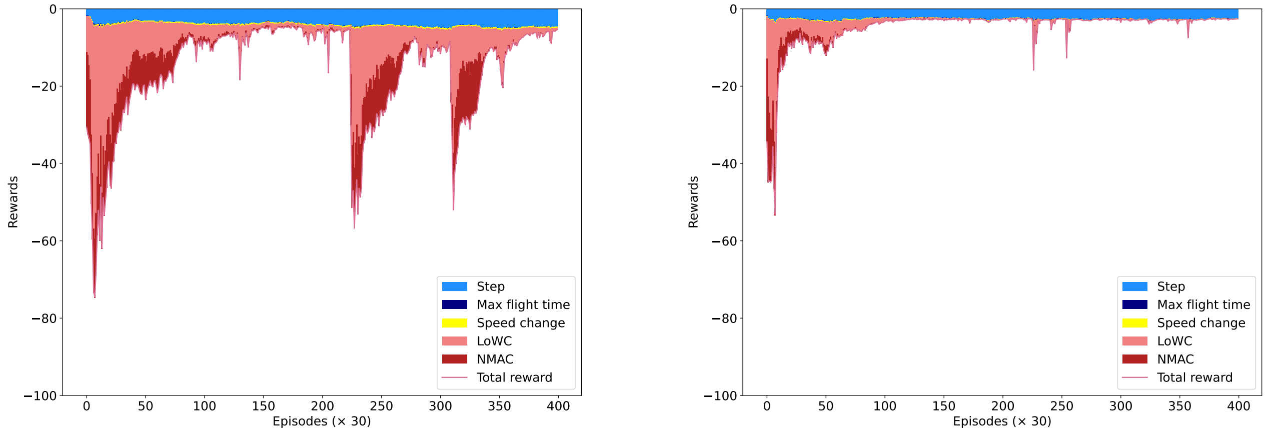
Drawing on the principles of game theory, we introduce a ”check ahead” policy. This policy refines the state space for the MARL model, effectively reducing the likelihood of ambiguous relationships arising between two proximate aircraft. The cost matrix of two aircraft on merging trajectories can be abstracted to that shown in Table III. In this context, having one agent opt for ”slow down” while another chooses ”speed up” is the only effective strategy to alleviate the conflict. Conversely, if both agents select the same action, it can result in a NMAC, which carries a large penalty of -1.0. Moreover, a lack of speed differentiation might lead to a LoWC, associated with a slightly lesser penalty of -0.5. It's also worth noting that opting to reduce speed introduces a minor efficiency cost, represented by a penalty of -0.1. In the previous work's setting [20], [22], when Aircraft 1 and Aircraft 2 identify each other as intruders, the case is a general sum game with two equilibriums ([speed up, slow down], [slow down, speed up]). This ambiguous relationship leads to difficult decision-making for both aircraft, resulting in a lower convergence rate for multi-agent reinforcement learning (MARL) training. However, if a policy is implemented where aircraft only check for leading aircraft and make decisions in order, the case can be changed to a Stackelberg game, with only one dominant equilibrium ([speed up, slow down]). This new relationship is simpler, making it easier for agents to select the correct actions. Figure 5 shows the learning curve for MARL with different intruder detection policies.

C. Rule-based Tactical Deconfliction

The rule-based tactical deconfliction method relies on predefined rules to determine the actions of aircraft to avoid NMACs, which is described in Figure 6. The rules are based on specific thresholds for distances between aircraft, including the NMAC threshold d_{NMAC}, low separation boundary d_{ls}, and high separation boundary d_{hs}.

In the case where the distance between two aircraft is closer than the NMAC threshold, the following aircraft will choose to hover or reduce speed to a minimum level to avoid a potential collision. This situation is defined as an NMAC event. If the distance between the following aircraft and the lead aircraft is lower than the low separation boundary, the following aircraft will choose to slow down. On the other hand, if the distance is larger than the high separation boundary, or if there is no leading aircraft, the following aircraft will choose to speed up.

The rule-based tactical deconfliction method serves as a benchmark in the study described in [13]. However, it should be noted that rule-based methods may have limitations in complex and dynamic environments, and it only provides a baseline approach for separation assurance but may require



(a)

(b)

Fig. 5: The learning curve of MARL with different intruder detection policies. (a) Detect all intruders nearby; (b) Only detect forward intruders.

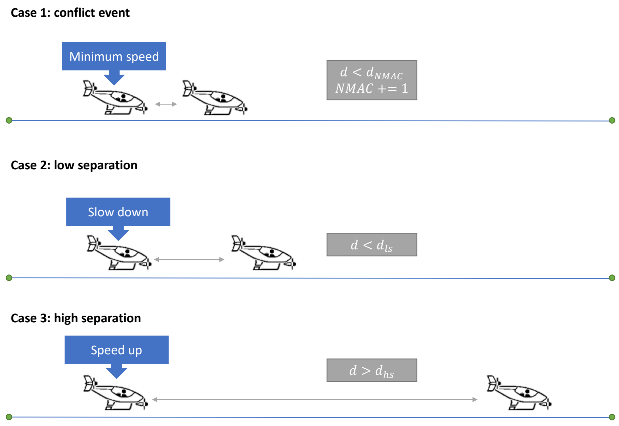


Fig. 6: Description of rule-based tactical deconfliction. In Case 1, a conflict event is depicted, prompting the trailing aircraft to immediately decelerate to minimum speed. Case 2 showcases a low separation scenario, where the trailing aircraft slows down to increase the separation distance. Case 3 portrays a high separation scenario, in which the trailing aircraft accelerates to reduce the separation.

further refinement and improvement for more complex situations.

V. Numerical Experiments

A. Simulation Environment

In this study, we use BlueSky [33] as our simulator to run a fast-time simulation. BlueSky is a widely acknowledged and accepted open-source platform in academia for aviation research. It is capable of running a large number of aircraft simulations in parallel efficiently. In addition, it is highly configurable, e.g., allowing the configuration of vertiport locations, waypoint locations, UAM routes, and aircraft performance parameters.

To study and evaluate the performance of the integrated conflict management system in structured airspace, we develop an evaluation scenario as shown in Figure 7, which defines capacity constrained resources as the typical bottlenecks in an airspace network. Three routes are included in the scenario:

- Route1: N-7 → N-1 → N-2 → N-3
- Route2: N-9 → N-1 → N-2 → N-3
- Route3: M-2 → N-2 → M-4

where N-1 and N-2 are two resources in this network. We implement the optimization-based DCB on both resources.

The detailed simulation parameters are listed in Table IV:

TABLE IV: simulation parameters

Category	**Parameter**	**Value**
Aircraft dynamic	Horizontal speed range	[5.14, 36.01] m/s
	Vertical speed range	[-7.62, 7.62] m/s
	Acceleration rate	3.5 m/s^2
Environment parameter	Route 1 distance	9.0 km
	Route 2 distance	9.0 km
	Route 3 distance	5.5 km
	Route altitude	121.9 m / 400 ft

B. Experimental Results

We conducted several numerical experiments to showcase the efficacy of our proposed integrated conflict management framework. Specifically, we first demonstrate the learning curve of the MARL training process on different capacities, which is used to determine the proper traffic density to obtain

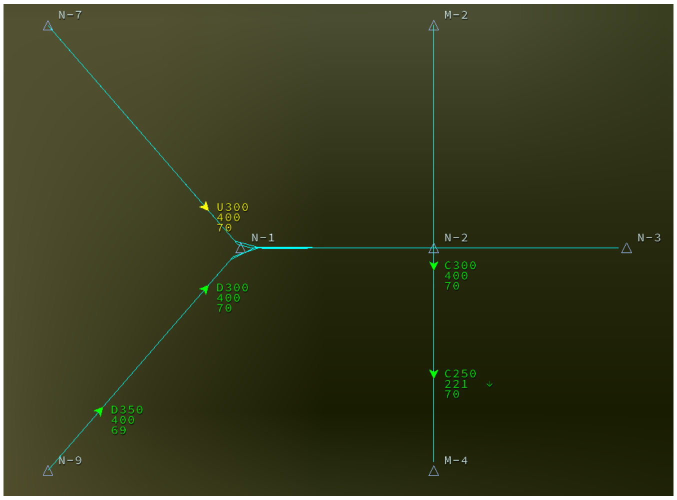


Fig. 7: Illustration of the hybrid scenario in the Bluesky simulator. The blue lines represent three routes, while the grey triangles with labels indicate the origin point, destination, and waypoints. Additionally, the green and yellow triangles represent each aircraft, displaying information such as the aircraft ID, altitude, and speed.

the best MARL model. Next, we determine the maximum capacity for the rule-based tactical method and MARL deconfliction model, as well as the capacity without any tactical deconfliction as a reference. Once we have the proper capacity value for DCB, we use those determined parameters and the trained model to compare the performance of different algorithm combinations using six metrics. Finally, we analyze the speed curve of various tactical deconfliction methods to gain insights into the reasons for differences in performance.

To ensure a fair comparison between the rule-based and MARL methods, we set the observation range to 1500m for both methods. The decision-making of the ownship aircraft would be affected by the intruders who are within this observation range.

1) **Learning Curve for Different Capacities:** The ultimate goal of the MARL model is to reduce penalties and determine the best policy for a given environment. However, if the traffic density is too high, or if aircraft do not have sufficient initial separation, it can be challenging for the MARL model to search for the optimal policy. In fact, high traffic density may lead the model to an unexpected local optimal policy, such as forcing all aircraft to airborne holding to avoid conflicts or even colliding to avoid further penalty steps. Therefore, it is essential to have a DCB layer as a precondition for MARL training.

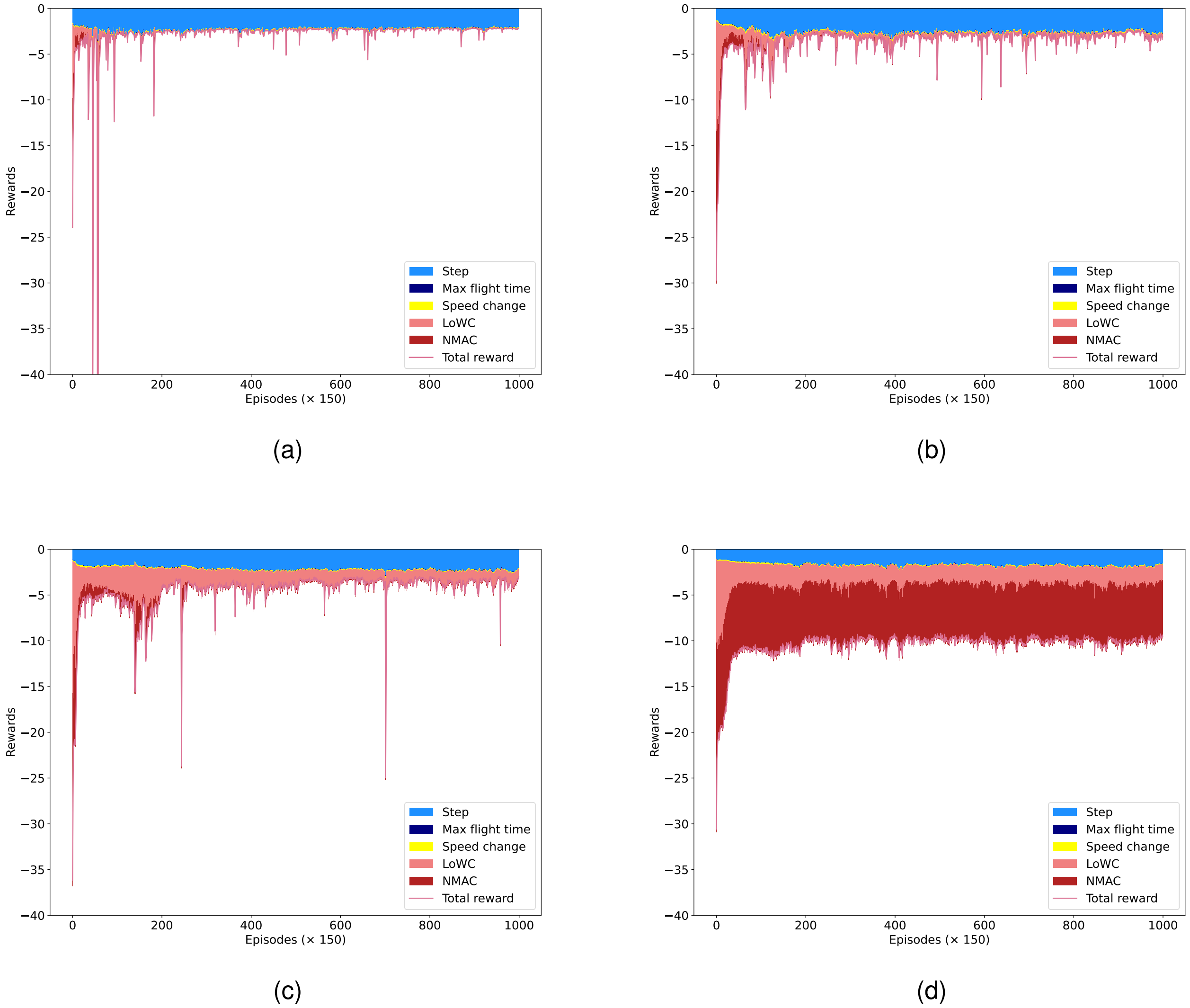
To train the MARL model, we utilized the flight schedule tables optimized by DCB as the training scenario. To generate these tables, we first created a set of original scheduled departure times $S_{d,f}, \forall f \in F^d$, corresponding to three departure points $d \in D$, independently. The departure intervals $S_{d,f+1} - S_{d,f}$ on each route follow a beta distribution, with the average interval λ used to control the traffic demand. Next, we used DCB with a fixed capacity to compute the set of required departure times $R_{d,f}, \forall d \in D, f \in F^d$, and compiled them into a flight schedule table. Each table contains 30 flight plans, which include information such as required departure time, origin, destination, waypoints, cruise speed, and cruise altitude. To avoid overfitting, we generated 100 different flight schedule tables and place them into a scenario pool. During training, the MARL model randomly selected a flight schedule table at each episode to improve its generalization performance. An episode is defined as a simulation round that fully executes the flight schedule table, starting from the first aircraft departure and ending with the final aircraft landing.

The training process consisted of a total of 150,000 episodes and was performed on two Nvidia RTX 3090 graphics cards. The model updated its weight every 30 episodes, and the simulation was executed in parallel with the support of the Ray python package [34]. The entire training process took roughly 4 hours.

Figure 8 depicts the learning curve on capacities of 6, 8, 10, and 30 operations per 200 seconds window, the latter of which corresponds to the case without DCB. The figure indicates that as the capacity increases, the MARL model faces greater difficulty in reaching the optimal policy. For instance, for a capacity of 6 operations per 200 seconds window, the model converges after 30,000 episodes, while for a capacity of 8 operations per 200s window, it continues searching for up to 120,000 episodes. Furthermore, the figure clearly illustrates the different components of the reward function described in Section IV-A. In Figure 8a, LoWC and NMAC events are infrequent, and the only cost incurred is the step penalty, which is introduced from the actual flying time and is unavoidable. In contrast, in Figure 8b and Figure 8c, the occurrence of NMAC is rare, while LoWC is more significant. Additionally, the speed change penalty is higher than in Figure 8a since the agent requires more maneuvers to avoid collisions. Figure 8d shows how MARL attempts to mitigate conflicts with no preconditioning by DCB. The primary component is the NMAC penalty, which implies a failure policy.

After careful consideration, we selected the best model trained with a capacity of 10 operations per 200s window for the subsequent experiments. This is because we want a model that will seek to prevent NMACs and this is the highest capacity that results in very few NMACs. In this paper, we do not seek to minimize LoWC events. It is noted that a MARL model trained on a highly constrained scenario generally performs well on a scenario that is not highly constrained, but the reverse may not hold.

2) **Performance with Different Capacities:** After showing the feasibility of MARL, the next challenge is to determine the maximum capacity that each tactical deconfliction method can support, while meeting a Target Level of Safety (TLS). To address this issue, we employed Monte Carlo simulations and evaluated system performance across a range of capacities from 1 to 11 operations per 200s window. Each capacity was applied for 30 simulation runs and the average value of the estimated MAC was recorded in each case. In order to observe the efficacy of DCB on different capacities, the original traffic demand was set up at a high level, where the average demand



(a)

(b)

(c)

(d)

Fig. 8: MARL learning curve for different capacities. (a) Capacity=6 operations per 200s window. (b) Capacity=8 operations per 200s window. (c) Capacity=10 operations per 200s window. (d) Without DCB.

interval is 30 seconds on each route. To select the appropriate capacity, we compared the average estimated MAC against a TLS of 0.94 MAC per 100,000 flight hours, in accordance with the United States Department of Transportation's proposed TLS for General Aviation aircraft in 2023 [35].

Table V displays the average estimated MACs for different capacities. As the capacity increases, the estimated MACs also increase for all three tactical methods, indicating that DCB can function effectively to precondition for tactical deconfliction. The table also reveals that, at any capacity level, the performance of the MARL model is superior to that of the rule-based approach. Based on the predefined TLS, we selected a capacity of 4 operations per 200s window for the system with the rule-based tactical method and a capacity of 7 operations per 200s window for the system with the MARL tactical method. This indicates that the MARL method is able to meet the TLS at a higher demand than the rule-based method. Furthermore, if the system lacks any tactical deconfliction method, only a capacity of 1 operation per 200s window is viable. This is effectively strategic deconfliction since only 1 operation is released into each time window.

It is noted that the estimated MACs per flight hour for the rule-based method at a capacity of 4 are below those of the MARL method. Moreover, for the MARL method, the MACs per flight hour decrease as the capacity increases from 5 to 7. These deviations from the general trends that inform our conclusions can be attributed to the inherent variability in the Monte-Carlo simulation. However, it's crucial to highlight that all these values remain within the TLS. We do not consider these variations to impact the broader observed trends or our conclusions regarding the MARL method's performance.

3) **Model Comparison:** In experiments 1 and 2 we successfully trained an effective MARL model for tactical deconfliction and established the maximum capacities of various

TABLE V: estimated MACs on different capacities

		Estimated MACs per 100,000 flight hours		
Traffic demand		**High**		
Target level of safety		**0.94**		
Tactical method		**None**	**Rule-based**	**MARL**
Capacity of DCB	1	**0.00**	0.00	0.00
	2	41.55	0.12	0.00
	3	56.89	0.68	0.00
	4	75.51	**0.74**	0.85
	5	84.32	10.30	0.90
	6	61.52	43.06	0.71
	7	122.14	115.34	**0.66**
	8	112.65	242.24	3.70
	9	145.36	370.52	23.88
	10	163.31	623.08	23.49
	11	155.52	673.24	32.39

tactical methods for strategic conflict management. In the experiment described here, we integrated these two components and conducted a comprehensive analysis, comparing different algorithm combinations using the six metrics outlined in section II. To evaluate the impact of different traffic demand levels, we tested each method under high, medium, and low traffic demand levels, corresponding to average departure intervals of 30, 60, and 120 seconds on each route, respectively. To ensure accuracy and eliminate the effects of randomness, we ran each experiment setting 30 times and reported the average values for each metric.

The final results are presented in Table VI. They lead us to draw several important conclusions.

- DCB is essential for safe separation. By incorporating a suitable maximum capacity for DCB, we were able to mitigate conflicts and maintain estimated MACs under the TLS. The first three rows in Table VI do not apply DCB. The first represents no tactical deconfliction, the second the rule-based tactical deconfliction method, and the third the MARL tactical deconfliction method, all applied without preconditioning by DCB to reduce the demand on the tactical systems to levels that would allow them to meet the TLS. Hence we do not expect the estimated MAC per 100,000 flight hours to meet the TLS in these cases. The last three rows correspond to the same tactical deconfliction methods, but with the DCB applied to precondition the traffic demand to a level that will allow the tactical deconfliction method to meet the TLS. It is evident that DCB plays a crucial role in eliminating conflicts and ensuring safety.
- DCB can help save energy by reducing fuel consumption and emissions. When traffic demand is high, DCB can lower the number of alerts and shorten flying time, which improves the efficiency metrics. However, to implement DCB, aircraft are delayed on the ground, with the length of the delay depending on the traffic demand and maximum capacity applied. It's worth noting that ground delay is not unique to DCB and exists in all three non-DCB methods as well. This is because the basic departure separation method used for tactical deconfliction in all cases also causes some small ground delays.
- Advanced tactical deconfliction methods, such as MARL, can increase system capacity and increase efficiency accordingly. MARL combined with DCB has similar safety metrics to the rule-based method with DCB and DCB with no tactical deconfliction, and all of these methods could guarantee safe separation. However, as the maximum capacity of each resource decreases, ground delay significantly increases. Thus, MARL is the most efficient method simulated because it allows for a higher airspace capacity, which ultimately leads to a decrease in ground delay.
- The performance of the rule-based tactical deconfliction method without DCB is worse than the no-intervention case. When the traffic density is too high, the risk ratio can be greater than 1, indicating that the rule-based method can lead to a higher risk of collisions than if no intervention is made at all (i.e., induce airspace risk). The rationale behind this assertion is based on the potential for aircraft to experience blockages en route in the absence of DCB regulation. In scenarios where DCB is not implemented, aircraft may reach their minimum speed, leaving them with limited options to avoid collisions. While it is possible to execute other rule-based tactical maneuvers to prevent blockages, our paper does not model them for the sake of simplicity. This observation highlights the necessity of using DCB in such scenarios, which can help reduce the risk of collisions and improve overall efficiency.

4) **Speed Curve Analysis:** Given the differences observed between the MARL and rule-based methods for tactical deconfliction in the previous experiments, we sought to investigate the factors contributing to these differences. To do so, we recorded and plotted the speed curves of the simulated aircraft, as shown in Figure 9. To facilitate readability, we selected eight aircraft uniformly from the total of 30 aircraft simulated.

We observed that the rule-based method for tactical deconfliction resulted in aircraft changing speed dramatically from maximum to minimum, often with rapid acceleration and deceleration. In contrast, the MARL tactical deconfliction method provides speed advisories considering a longer-term view. For instance, for aircraft D533 (the brown curve in Figure 9), the MARL method advised holding at a relatively lower speed range for a period, helping the aircraft avoid slowing down to the minimum speed recommended by the rule-based method. This adjustment allowed the aircraft to arrive earlier than the rule-based method suggested. We also observed speed oscillations in the rule-based separation method, as illustrated by aircraft D118 (the orange curve in Figure 9). This occurred because the aircraft was in a situation where the distance to the leading aircraft was exactly on the boundary of the threshold for speed-up and slow-down.

In summary, the MARL tactical deconfliction method pro-

TABLE VI: numerical results

Traffic demand		High	Medium	Low	High	Medium	Low	High	Medium	Low
Safety metrics		**LoWCs/ flight hr**			**Estimated MACs/ 100,000 flight hrs**			**Risk ratio**		
Algorithm	**No Intervention**	467.0	313.0	162.4	205.53	137.62	76.19	-	-	-
	Rule-based	1263.3	908.5	376.7	908.25	559.46	193.73	4.4195	4.0654	2.5423
	MARL	792.4	232.5	127.8	195.51	17.53	30.85	0.9513	0.1274	0.4049
	DCB, C=1	0.0	0.0	0.0	0.00	0.00	0.00	0.0000	0.0000	0.0000
	Rule-based+DCB, C=4	25.9	34.8	30.8	0.74	0.77	0.92	0.0036	0.0056	0.0120
	MARL+DCB, C=7	45.6	62.8	49.6	0.66	0.65	0.51	0.0032	0.0047	0.0068
Efficiency metrics		**Number of alerts**			**Airborne delay (seconds)**			**Ground delay (seconds)**		
Algorithm	**No Intervention**	-	-	-	0.0	0.0	0.0	28.7	9.1	3.5
	Rule-based	73.1	66.0	50.2	260.3	191.3	93.6	28.7	9.1	3.5
	MARL	25.9	22.8	15.5	71.4	78.3	26.4	28.7	9.1	3.5
	DCB, C=1	-	-	-	0.0	0.0	0.0	2566.1	2580.9	2444.4
	Rule-based+DCB, C=4	22.6	32.1	25.6	15.0	23.7	18.7	505.2	406.2	293.0
	MARL+DCB, C=7	18.1	19.5	16.5	30.8	32.9	22.2	158.4	74.4	19.9

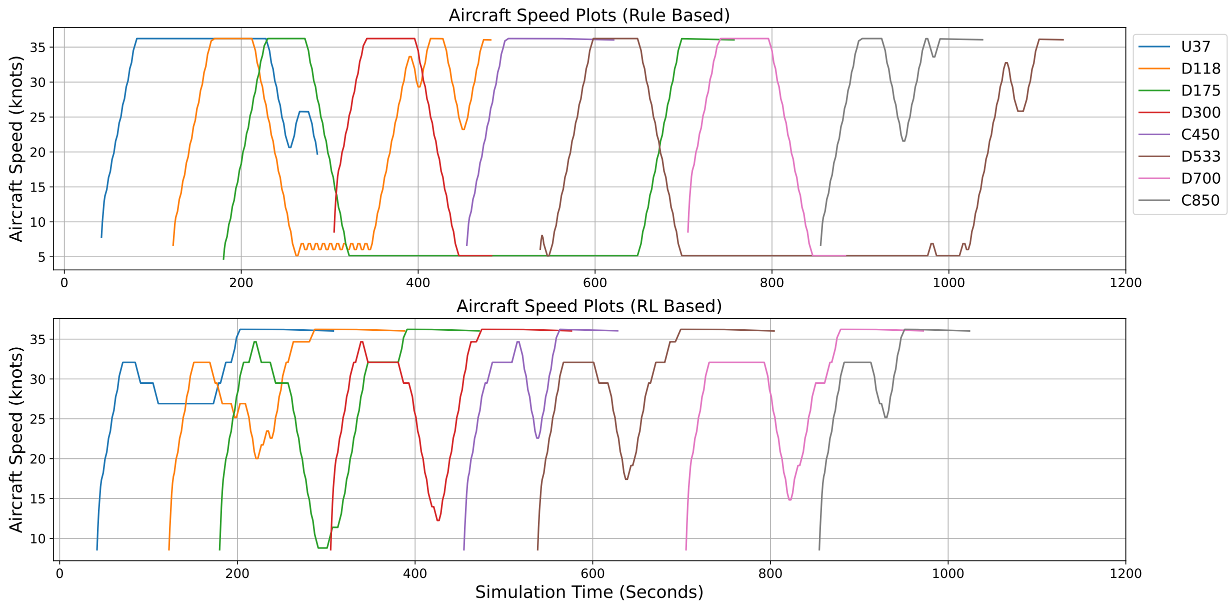


Fig. 9: The comparison of speed curves of aircraft with the rule-based tactical method and the MARL methods. The y-axis in each plot represents the aircraft's actual speed in knots, while the x-axis is the simulation time in seconds. Each line represents an aircraft.

vides more optimal speed advisories compared to the rule-based method, allowing aircraft to arrive earlier and avoid rapid acceleration and deceleration, which may lead to more efficient and stable flight operations.

VI. Conclusion

Our approach demonstrated promising results in reducing the number of conflicts and improving the efficiency of UAM operations at scale. The integrated conflict management framework, which combines strategic conflict management and tactical deconfliction methods, offers a comprehensive solution to address some of the challenges in high-density UAM operations. Our research showed that the optimization-based multiple resource demand capacity balancing algorithm plays a crucial role in preconditioning for tactical deconfliction. The successful implementation of game theory also improved the performance of the tactical deconfliction model, saving computational resources and making it possible to apply the system in the real world. In addition, the Monte-Carlo simulation we used to study the interactions between the strategic and tactical safety assurance methods provided valuable insights that can contribute to the development of more effective and efficient

UAM systems in the future.

One of the next steps in this research is to thoroughly investigate and understand the interplay between strategic and tactical conflict management methods. Currently, strategic conflict management computes the optimal departure time based on a deterministic estimated flying time based on known operations. However, tactical deconfliction within the system may introduce speed changes that can affect the estimated time of arrival (ETA) at resources. As airspace networks become more complex, these time differences can accumulate and result in reduced effectiveness of the preconditioning by strategic conflict management systems. Therefore, formulating the ETA stochastically by considering the method of tactical deconfliction could increase the system's robustness in complex networks.

Acknowledgements

Authors Brittain and Wei are partially supported by the NASA Grant 80NSSC21M0087 under the NASA System-Wide Safety (SWS) program.

References

[1] K. Balakrishnan, J. Polastre, J. Mooberry, R. Golding, and P. Sachs, "Blueprint for the sky: The roadmap for the safe integration of autonomous aircraft," *Airbus UTM, San Francisco, CA*, 2018.

[2] D. Jenkins, B. Vasigh, C. Oster, and T. Larsen, *Forecast of the commercial UAS package delivery market*. Embry-Riddle Aeronautical University, 2017.

[3] B. A. Hamilton, "Urban air mobility market study," Presentation to NASA Aeronautics Research Mission Directorate URL: https://go.nasa.gov/2MVSbth, 2018.

[4] N. R. Council *et al.*, *Autonomy research for civil aviation: toward a new era of flight*. National Academies Press, 2014.

[5] H. Erzberger, "Transforming the nas: The next generation air traffic control system," Tech. Rep., 2004.

[6] H. Erzberger and K. Heere, "Algorithm and operational concept for resolving short-range conflicts," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 224, no. 2, pp. 225–243, 2010.

[7] H. Erzberger and E. Itoh, "Design principles and algorithms for air traffic arrival scheduling," Tech. Rep., 2014.

[8] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, "Unmanned aircraft system traffic management (utm) concept of operations," in *16th AIAA Aviation Technology, Integration, and Operations Conference*. AIAA, 2016, pp. 1–16.

[9] D. P. Thipphavong, R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K. H. Goodrich, J. Homola *et al.*, "Urban air mobility airspace integration concepts and considerations," in *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3676.

[10] Federal Aviation Administration, *Urban Air Mobility (UAM) Concept of Operations v1.0*. U.S. Department of Transportation Federal Aviation Administration, Washington DC, 2020.

[11] ——, *Urban Air Mobility (UAM) Concept of Operations v2.0*. U.S. Department of Transportation Federal Aviation Administration, Washington DC, 2023.

[12] G. Zhu, "Decision making under uncertainties for air traffic flow management," Ph.D. dissertation, Iowa State University, 2019.

[13] S. Chen, P. Wei, A. D. Evans, and M. Egorov, "Estimating airspace resource capacity for advanced air mobility operations," in *AIAA AVIATION 2022 Forum*, 2022, p. 3317.

[14] P. Razzaghi, A. Tabrizian, W. Guo, S. Chen, A. Taye, E. Thompson, A. Bregeon, A. Baheri, and P. Wei, "A survey on reinforcement learning in aviation applications," *arXiv preprint arXiv:2211.02147*, 2022.

[15] J. P. Chryssanthacopoulos and M. J. Kochenderfer, "Accounting for state uncertainty in collision avoidance," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 951–960, 2011.

[16] H. Y. Ong and M. J. Kochenderfer, "Markov decision process-based distributed conflict resolution for drone air traffic management," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 1, pp. 69–80, 2017.

[17] J. Bertram and P. Wei, "Distributed computational guidance for high-density urban air mobility with cooperative and non-cooperative collision avoidance," in *AIAA Scitech 2020 Forum*, 2020, p. 1371.

[18] A. G. Taye, J. Bertram, C. Fan, and P. Wei, "Reachability based online safety verification for high-density urban air mobility trajectory planning," in *AIAA AVIATION 2022 Forum*, 2022, p. 3542.

[19] M. Brittain and P. Wei, "Scalable autonomous separation assurance with heterogeneous multi-agent reinforcement learning," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2837–2848, 2022.

[20] M. Brittain and P. Wei, "Autonomous separation assurance in an high-density en route sector: A deep multi-agent reinforcement learning approach," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3256–3262.

[21] M. W. Brittain and P. Wei, "One to any: Distributed conflict resolution with deep multi-agent reinforcement learning and long short-term memory," in *AIAA Scitech 2021 Forum*, 2021, p. 1952.

[22] M. W. Brittain, X. Yang, and P. Wei, "Autonomous separation assurance with deep multi-agent reinforcement learning," *Journal of Aerospace Information Systems*, vol. 18, no. 12, pp. 890–905, 2021.

[23] W. Guo, M. Brittain, and P. Wei, "Safety enhancement for deep reinforcement learning in autonomous separation assurance," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 348–354.

[24] A. Weinert, S. Campbell, A. Vela, D. Schuldt, and J. Kurucar, "Well-clear recommendation for small unmanned aircraft systems based on unmitigated collision risk," *Journal of air transportation*, vol. 26, no. 3, pp. 113–122, 2018.

[25] A. Weinert, L. Alvarez, M. Owen, and B. Zintak, "A quantitatively derived nmac analog for smaller unmanned aircraft systems based on unmitigated collision risk," 2020.

[26] S. M. Katz, L. E. Alvarez, M. Owen, S. Wu, M. W. Brittain, A. Das, and M. J. Kochenderfer, "Collision risk and operational impact of speed change advisories as aircraft collision avoidance maneuvers," in *AIAA AVIATION 2022 Forum*, 2022, p. 3824.

[27] M. P. Owen, A. Panken, R. Moss, L. Alvarez, and C. Leeper, "Acas xu: Integrated collision avoidance and detect and avoid capability for uas," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, 2019, pp. 1–10.

[28] L. E. Alvarez, I. Jessen, M. P. Owen, J. Silbermann, and P. Wood, "Acas sxu: Robust decentralized detect and avoid for small unmanned aircraft systems," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, 2019, pp. 1–9.

[29] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[31] Radio Technical Commission for Aeronautics, *Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B)*. RTCA, Incorporated, 2002.

[32] Federal Aviation Administration. UAS Remote Identification Overview. Accessed: January 2022.

[33] J. M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: an open data and open source approach," in *Proceedings of the 7th International Conference on Research in Air Transportation*, vol. 131. FAA/Eurocontrol USA/Europe, 2016, p. 132.

[34] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica, "Ray: A Distributed Framework for Emerging AI Applications," 2017.

[35] U.S. Department of Transportation, "DOT Progress in Aviation Safety: FY 2022," https://assets.performance.gov/APG/files/2022/may/FY2022_May_DOT_Progress_Aviation_Safety.pdf, May 2022.

Shulu Chen received a bachelor's degree from the School of Automation Science and Electrical Engineering at Beihang University and received Master's degree from the Department of Industrial Engineering at the University of Illinois at Urbana Champaign. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering at George Washington University. He is also working as a Research Assistant with the Intelligent Aerospace Systems Laboratory (IASL), under the supervision of Prof. Peng Wei. His research interests include deep reinforcement learning, optimization, and game theory, with applications in air traffic management and airline revenue management.

Antony D. Evans is the Director of System Design for Airbus UTM at Acubed, the Airbus innovation center in Silicon Valley, California. Tony has 17 years of research experience in air transportation, and has published widely on air traffic management, aviation and the environment, unmanned traffic management and urban air mobility. He has two Masters degrees from MIT and a PhD from the University of Cambridge.

Marc Brittain is a member of the Technical Staff at MIT Lincoln Laboratory in the Air Traffic Control mission area. His research interests include decision making under uncertainty, safe artificial intelligence, and reinforcement learning in air transportation. At Lincoln Laboratory, his works includes the development of the AI Testbed for Advanced Air Mobility (AAM-Gym) and the evaluation of AI algorithms for separation assurance in AAM corridors. Marc serves as a member on the AIAA Air Transportation Technical Committee. He holds a Ph.D. in Aerospace Engineering from Iowa State University.

Peng Wei (Member, IEEE) received the Ph.D. degree in aerospace engineering from Purdue University, in 2013. He is currently an Assistant Professor with the Department of Mechanical and Aerospace Engineering, George Washington University, with courtesy appointments at the Electrical and Computer Engineering Department and the Computer Science Department. He is also leading the Intelligent Aerospace Systems Laboratory (IASL). By contributing to the intersection of control, optimization, machine learning, and artificial intelligence, he develops autonomy and decision support tools for aeronautics, aviation, and aerial robotics. His current research interests include safety, efficiency and scalability of decision making systems in complex, uncertain, and dynamic environments. His recent applications include Air Traffic Control/Management (ATC/M), Airline Operations, UAS Traffic Management (UTM), eVTOL Urban Air Mobility (UAM), and Autonomous Drone Racing (ADR). He is an Associate Editor of the AIAA Journal of Aerospace Information Systems.