

RESEARCH ARTICLE

Priority-based intelligent resolution method of multi-aircraft flight conflicts

D. Sui¹, Z. Zhou¹ and X. Cui

Nanjing University of Aeronautics and Astronautics, College of Civil Aviation, Nanjing, China

Corresponding author: Z. Zhou; Email: 2329669117@nuaa.edu.cn

Received: 6 November 2023; Revised: 4 June 2024; Accepted: 18 June 2024

Abstract

The rising demand for air traffic will inevitably result in a surge in both the number and complexity of flight conflicts, necessitating intelligent strategies for conflict resolution. This study addresses the critical challenges of scalability and real-time performance in multi-aircraft flight conflict resolution by proposing a comprehensive method that integrates a priority ranking mechanism with a conflict resolution model based on the Markov decision process (MDP). Within this framework, the proximity between aircraft in a multi-aircraft conflict set is dynamically assessed to establish a conflict resolution ranking mechanism. The problem of multi-aircraft conflict resolution is formalised through the MDP, encompassing the design of state space, discrete action space and reward function, with the transition function implemented via simulation prediction using model-free methods. To address the positional uncertainty of aircraft in real-time scenarios, the conflict detection mechanism introduces the aircraft's positional error. A deep reinforcement learning (DRL) environment is constructed incorporating actual airspace structures and traffic densities, leveraging the Actor Critic using Kronecker-factored Trust Region (ACKTR) algorithm to determine resolution actions. The experimental results indicate that with 20–30 aircraft in the airspace, the success rate can reach 94% for the training set and 85% for the test set. Furthermore, this study analyses the impact of varying aircraft numbers on the success rate within a specific airspace scenario. The outcomes of this research provide valuable insights for the automation of flight conflict resolution.

Nomenclature

σ_{CT}	cross-track standard deviation
σ_{AT}	along-track standard deviation
R_0, R_1	the lengths of the long axes of the confidence ellipses
D_H	horizontal separation
D_V	vertical separation
t_h	the advance detection time
t_i	the time interval for conflict detection
Con_{ij}	the degree of urgency between conflict aircraft i and conflict aircraft j
t_{lose}	the moment of loss of safe separation between aircraft i and aircraft j
S_1	sorting sequence
S_2	execution instruction sequence
S_{EXE}	whether to execute the command sequence
N_n	number of aircraft
N_t	number of discrete time periods
N_d	specific information about each aircraft at a given moment in time
O_t	the observation of the agent
S_t	the state of the corresponding rectangular airspace
$Info_1^T$	the specific flight information of the i th aircraft in the framed airspace at time T
cmd	a specific command executed by the aircraft

T_w	the waiting time for the execution of the action
T_a	an upper limit on the execution time of an instruction
θ	the overall network parameter
$p(\tau)$	the distribution of trajectories

Abbreviations

<i>MDP</i>	Markov decision processes
<i>DRL</i>	deep reinforcement learning
<i>ATCOs</i>	air traffic controllers
<i>PBN</i>	performance-based navigation
<i>CD&R</i>	conflict detection and resolution
<i>TCS</i>	tactical conflict solver
<i>SGD</i>	stochastic gradient descent
<i>ACKTR</i>	Actor-Critic Kronecker-factored Trust Region
<i>ATOSS</i>	Air Traffic Operation Simulation System
<i>BADA</i>	base of aircraft data

1.0 Introduction

As the impact of the epidemic receded, the civil aviation industry experienced a resurgence. In February 2023, China Civil Aviation transported 43.2 million passengers [1], marking a 38% increase compared to the same period in 2022, along with 452,000 tons of cargo and mail, a rise of 21,000 tons over the corresponding period. The increased air traffic will inevitably lead to a higher incidence of flight conflicts and an increased workload for air traffic controllers (ATCOs), posing a threat to flight safety. Traditionally, ATCOs' workload has been mitigated by creating new sectors; however, due to the constraints of airspace resources, the expansion of sectors is not a sustainable solution. The integration of intelligent technologies to assist ATCOs in their duties presents a promising resolution to this challenge. Therefore, Eurocontrol's European ATM Master Plan document [2] outlines a strategic approach for utilising intelligent decision support technologies to support ATCOs in air traffic management, thereby categorising the hierarchy of intelligent assistance.

One of the fundamental responsibilities of air traffic control is to resolve flight conflicts. In a dynamic and open-air traffic environment, flight conflicts can arise between two aircraft, with more complex scenarios emerging when multiple aircraft are involved. Historically, the resolution of flight conflicts has been predominantly investigated through operations research, intelligent search algorithms and optimal control methodologies. In recent years, scholars have increasingly turned to deep learning techniques to address this issue. Nevertheless, a mature solution capable of effectively managing the challenges associated with multi-aircraft conflict resolution in practical settings is currently lacking. This is primarily due to the high demands on success rate, solution time, scalability with the number of conflicting aircraft, management of flight trajectory uncertainties and the need to ensure compatibility between manoeuvres and aircraft performance.

This paper proposes a generic flight conflict resolution method that combines a priority ranking mechanism with an MDP-based conflict resolution model. The priority-based sorting mechanism effectively addresses the scalability issue inherent in the flight conflict resolution model. Integration with the DRL algorithm significantly reduces conflict resolution time to seconds, enhancing resolution efficiency. Considering the aircraft's horizontal position error and flight performance constraints, this model was trained and tested in an air traffic simulation system. The multi-aircraft conflict resolution strategy derived from this method proves to be more apt for real-world scenarios.

2.0 Previous related works

In the early stages of flight conflict resolution research, conventional methodologies predominantly included mathematical planning, optimal control and swarm intelligence methods, etc. Mathematical planning methods are typically fast and scalable [3–5], but they are often simplified and set a series of assumptions in the modeling process, potentially limiting their practical applicability. Optimal control strategies generate conflict resolution programs typically defined as trajectories comprising sequential position points rather than the manoeuvre instructions required by air traffic controllers [6, 7]. Swarm intelligence methods [8, 9] cannot guarantee the global optimal solution, and the solution speed in complex dynamic scenes makes it difficult to meet the requirements of practical applications.

In recent years, reinforcement learning-like intelligent methods have been introduced into the domain of flight conflict resolution, yielding promising results. Two-aircraft flight conflicts represent a common scenario in air traffic control. One study [10] utilised a deep deterministic policy gradient (DDPG) model to resolve a conflict between two aircraft in a continuum-type action space. The model achieved a success rate of approximately 87% under various levels of uncertainty. Literature [11] considered real-world ATCOs' experience while training a single agent to resolve a two-aircraft conflict, making the given instructions more acceptable to the ATCOs and increasing anthropomorphism. Literature [12] trained a single-agent model to resolve conflicts between two aircraft in high-density and high-uncertainty traffic flow environments, with the proposed DDPG-2S algorithm demonstrating strong performance. Literature [13] extended the conflict resolution environment into three dimensions, trained a model to address two-aircraft conflicts while accounting for uncertainty and successfully combined it with a deep reinforcement learning algorithm to achieve notable success rates.

Multi-aircraft flight conflicts, characterised by spatio-temporal correlation, present significant challenges in conflict resolution due to unstable environments, limited scalability and the lack of effective coordination mechanisms. Efforts to address multi-aircraft conflicts typically rely on multi-agent deep reinforcement learning approaches that incorporate implicit coordination mechanisms. These strategies leverage the concept of stochastic games within multi-agent reinforcement learning algorithms to train agents in cooperative behaviours. Sui et al. [14] solved multi-aircraft conflicts based on the Independent Deep Q Network (IDQN) algorithm, which adopts a downwards-compatible approach to solve the problem of a variable number of conflicting aircraft and reduces the solution time to the level of 0.011s. Ramon et al. [15] combined message-passing neural networks with multi-agent reinforcement learning, enhancing agents' coordination by exchanging critical information for optimal joint action. Lai et al. [16] employed the multi-agent reinforcement learning algorithm (Multi-Agent Deep Deterministic Policy Gradient) MADDPG to resolve conflicts by diverting flight paths and achieved reasonable success rates on a fixed number of aircraft. Wei's team [17] developed the Distributed Reinforcement Learning for Interval Maintenance D2MAV framework, integrating LSTM to enhance scalability and minimise separation loss for homogeneous agent aerial vehicles. The team further developed the distributed multi-agent autonomous separation assurance (MAASA) framework [18], which can predict intruding aircraft's movement intentions and perform better than the D2MAV-A framework. Ralvi et al. [19] utilised the MADDPG algorithm for conflict resolution and considered conditions such as time, fuel and airspace complexity while achieving a 93% success rate. Huang et al. [20] introduced a multi-agent asynchronous advantage actor-critic (MAA3C) framework and used mask recurrent neural networks (RNN) to solve the scalability problem, which provides a solution idea for conflict management at the strategic level. Chen et al. [21] proposed a general multi-agent approach combined with adaptive manoeuvre strategies. The article considered trajectory uncertainty and scalability of the number of aircraft, but the experimental environment was only on a two-dimensional scale. George et al. [22] advanced conflict resolution strategies by employing an augmented graph-convolutional reinforcement learning method in a multi-agent environment, featuring a centralised training-decentralised execution training scheme. Despite the potential of multi-agent deep reinforcement learning approaches in enhancing information exchange and collaboration among homogeneous or heterogeneous agents, care must be taken to avoid secondary conflicts stemming from poor coordination. Furthermore, managing

the scalability of multi-agent systems remains intricate, with challenges such as dimensional explosion emerging as the number of agents increases.

In contrast to the implicit coordination mechanism inherent in multi-agent reinforcement learning, an explicit coordination mechanism at the macro level external to the agents could offer additional benefits. This approach involves training a single-agent resolution model and subsequently integrating it with an explicit coordination mechanism to facilitate conflict resolution in multi-aircraft flight scenarios. By doing so, aircraft in conflict can be managed sequentially without internal agent coordination. For the training of single-agent models, the literature [23] has addressed two- and multi-aircraft conflicts using single-agent systems, developing various conflict resolution frameworks for comparative analysis. Literature [24] leveraged the PPO algorithm to train conflict resolution strategies, combining a priori domain knowledge with advanced data analysis.

For the study of the explicit coordination mechanism, Zhang et al. [25] from the perspective of low-altitude rescue aircraft operation, regarded each aircraft as an independent agent, and each agent determined the priority according to the task and state it performed and subsequently made the conflict resolution strategy according to the conflict resolution order respectively. Wang et al. [26] solved the multi-aircraft flight conflict problem based on satisfactory game theory, in which a set of priority ranking mechanisms is proposed to characterise the importance of different aircraft, and the aircraft selects the most satisfactory resolution strategy based on its priority and the preference of others. Wu et al. [27] established a flight conflict network based on flight conflict relationships between aircraft using aircraft as nodes. The mechanism of nodes and connecting edges ensures that critical aircraft are prioritised for deployment and high-risk conflicts are prioritised for resolution. Florence et al. [28] proposed two decentralised coordination approaches to cater to the decentralised requirements in the Multi-Agent Path Finding (MAPF) problem. One method involved prioritising agents according to their cost functions, with lower-priority agents using paths generated by higher-priority agents as constraints to compute subsequent paths. By determining the resolution order, the explicit coordination mechanism coordinates conflicting aircraft in a multi-aircraft flight conflict. This dramatically simplifies the cooperation between aircraft and enables the orderly handling of different numbers of conflicting aircraft.

In order to compare the intelligent conflict resolution methods related to reinforcement learning, Table 1 can be obtained by combing the conflict types, theoretical methods, success rates, scalability, uncertainty, application scenarios, etc. Comparative analyses show that intelligent multi-aircraft flight conflict resolution methods that satisfy all the above requirements still need further exploration. The novel multi-aircraft conflict resolution approach delineated in this study effectively addresses scalability concerns, expands the experimental domain to three dimensions and attains a commendable success rate in densely trafficked environments, all while accommodating aircraft performance constraints.

The remainder of this article is organised as follows: Section 3 provides a detailed discussion on the concept of multi-aircraft conflict and associated uncertainties. Section 4 describes the logical framework for conflict detection and resolution, Section 5 models the MDP for multi-aircraft conflict resolution, Section 6 introduces the ACKTR algorithm in order to train the agents, Section 7 conducts simulation experiments to validate the feasibility of the proposed scheme, Section 8 discusses the strengths and weaknesses of the model, and finally, Section 9 draws a conclusion and proposes directions for future research.

3.0 Problem statement

This section focuses on the problem statement, including the definition of a multi-aircraft flight conflict scenario and the concept of horizontal uncertainty.

3.1 Multi-aircraft flight conflict scenarios

A multi-aircraft flight conflict is a situation in which the prescribed separation criteria is violated between multiple neighbouring aircraft within a relatively short period, which can also be viewed as

Table 1. Reinforcement learning based flight conflict resolution literature

Author	Number of conflict aircraft	Specific method	Success rate	Scalability	Uncertainty	Application scenarios	Aircraft performance	Solution time
Pham [10]	Two-aircraft	DDPG	87%	×	✓	2D	×	–
Tran [11]	Two-aircraft	DDPG	–	×	×	2D	✓	3.34s
Pham [12]	Two-aircraft	DDPG-2S	95% ¹	×	✓	2D	×	–
Sui [13]	Two-aircraft	ACKTR	99%	×	✓	3D	✓	Average 1.13s
Sui [14]	Multi-aircraft	IDQN	85%	✓	×	3D	✓	Average 0.011s
Ramon [15]	Multi-aircraft	PPO	–	×	×	2D	×	–
Lai [16]	Multi-aircraft	MADDPG	100% ²	×	×	2D	×	–
Brittain [17]	Multi-aircraft	PPO	–	✓	×	2D	×	–
Brittain [18]	Multi-aircraft	PPO	–	✓	✓	2D	×	–
Ralvi [19]	Multi-aircraft	MADDPG	93%	×	×	2D	×	–
Huang [20]	Multi-aircraft	MAA3C	100% ³	✓	×	3D	×	–
Chen [21]	Multi-aircraft	Rainbow DQN	100% ⁴	✓	✓	2D	×	–
George [22]	Multi-aircraft	DGN	97% ⁵	✓	×	3D	✓	–
Sheng [23]	Multi-aircraft	DQN	–	×	×	2D	×	–
Zhao [24]	Multi-aircraft	PPO	–	✓	✓	2D	×	–

Note: 1. This success rate is the conflict resolution rate at low uncertainty levels; 2. This success rate was achieved for a low-density traffic flow with an aircraft count of three; 3. This success rate targets the environment of airspace demand capacity balancing issues in the strategic phase as well as conflict separation; 4. This success rate was achieved in a specific experimental environment and with aircraft less than ten.; 5. The authors claim that this is a micro-average result.

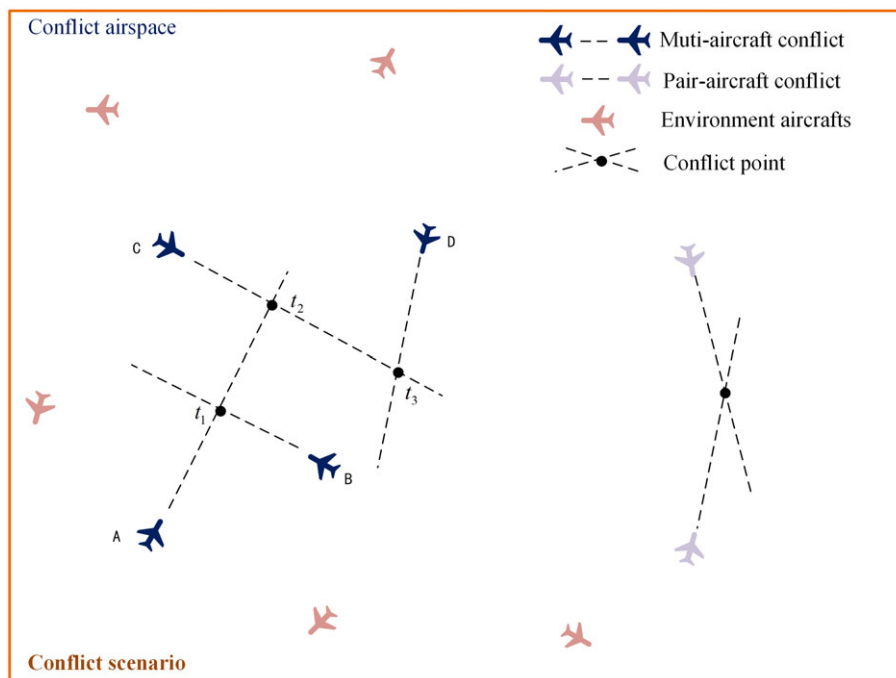


Figure 1. Schematic of the flight conflict scenario.

a combination of multiple two-aircraft flight conflicts. However, due to the potential interconnections between these individual conflicts, it is not reasonable to simply decompose a multi-aircraft flight conflict into multiple two-aircraft flight conflicts and then resolve them as a two-aircraft conflict.

Figure 1 shows a four-aircraft conflict scenario with spatio-temporal correlation, where the conflict detection mechanism successively detects that during time Δt , the conflict between aircraft A and aircraft B occurs at time t_1 , the conflict between aircraft A and aircraft C occurs at time t_2 , and the conflict between aircraft C and aircraft D occurs at time t_3 . The spatio-temporal correlation here is reflected in the fact that the occurrence times of t_1 , t_2 , and t_3 are all during the time Δt , and the midpoints of the conflict lines between aircraft A and aircraft B, aircraft A and aircraft C, and aircraft C and aircraft D are all in the delineated conflict airspace. Spatio-temporal relevance is also reflected in the fact that an aircraft may be involved in multiple two-aircraft conflicts, which requires the aircraft to consider more constraints when performing conflict resolution manoeuvres and to try to resolve as many conflicts as possible with a single manoeuvre. Typically, the airspace consists of conflicting aircraft within flight conflicts, as well as non-conflicting environmental aircraft.

3.2 Uncertainty of horizontal position

Due to navigation errors, flight technique errors, imperfections in the prediction system, and other factors, there may be an error between the aircraft's actual position and the prediction results, and this uncertainty must be considered when resolving conflicts. In this study, we address the problem of uncertainty of flight trajectories in the horizontal plane by referring to the model of Lauderdale [29] and measuring the position uncertainty of an aircraft using a probabilistic type of method that relies on a Gaussian representation of the position error. A plane rectangular coordinate system is established with the aircraft heading as the x-axis positive direction and the predicted aircraft position as the origin, as shown in Fig. 2.

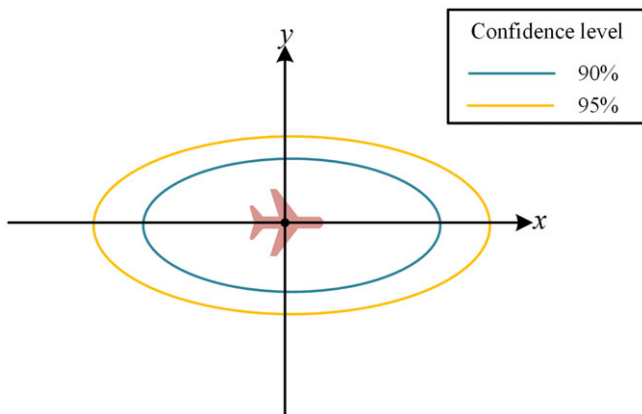


Figure 2. Uncertainty in the actual position of aircraft in the horizontal plane.

Let the actual position of the aircraft be (X, Y) , then the two-dimensional random variable (X, Y) obeys the following two-dimensional Gaussian distribution:

$$\text{Pos}(A)_{(x,y)} \sim N(\mu(t, \psi), \sum (t, \psi)^2) \tag{1}$$

where $N(\cdot)$ is the normal distribution with mean μ and standard deviation \sum , t is the time in the future, and ψ is the aircraft heading. By decomposing the normal distribution using principal axes and assuming that there is no deviation from the mean, the equation can be transformed into the following form:

$$\text{Pos}(A)_{(x,y)} \sim N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{AT}^2 & 0 \\ 0 & \sigma_{CT}^2 \end{pmatrix} \right) \tag{2}$$

where σ_{CT} is the cross-track standard deviation, σ_{AT} is the along-track standard deviation. σ_{CT} is a constant. Because the cross-track position is controlled by the flight management system, the error does not increase over time. σ_{AT} is a linear function of time, The along-track-position position error accumulates over time and is satisfied:

$$\sigma_{AT} = \sigma_{AT_0} + \alpha t \tag{3}$$

where α describes how the along-track uncertainty grows with prediction time t and has units of nautical miles per minute.

Combined with the above definition of uncertainty, the conflict detection criteria for two-aircraft in this paper are described as follows: as shown in Fig. 2, the uncertainty in defining the aircraft’s position is expressed as the expansion of the aircraft from the standard protection zone. Depending on the different confidence levels, the confidence ellipse of the 2D Gaussian distribution can be determined. A circle is then made with the ellipse’s long axis as the radius, assuming that the aircraft’s actual position does not extend beyond this circle. Given that aircraft can autonomously adjust their heading, the radius of their flight path circle may expand over time. However, in a performance-based navigation (PBN) operational setting, such as under the assumption of the RNAV1 navigation specification, this radius is constrained to be at most one nautical mile. As shown in Fig. 3, where R_0 and R_1 , etc., are the lengths of the long axes of the confidence ellipses of the two aircraft, and the blue ellipse is the confidence ellipse, a flight conflict is considered to have occurred between the two aircraft if $D_H \leq 10$ km and $D_V \leq 300$ m; otherwise, there is no conflict.

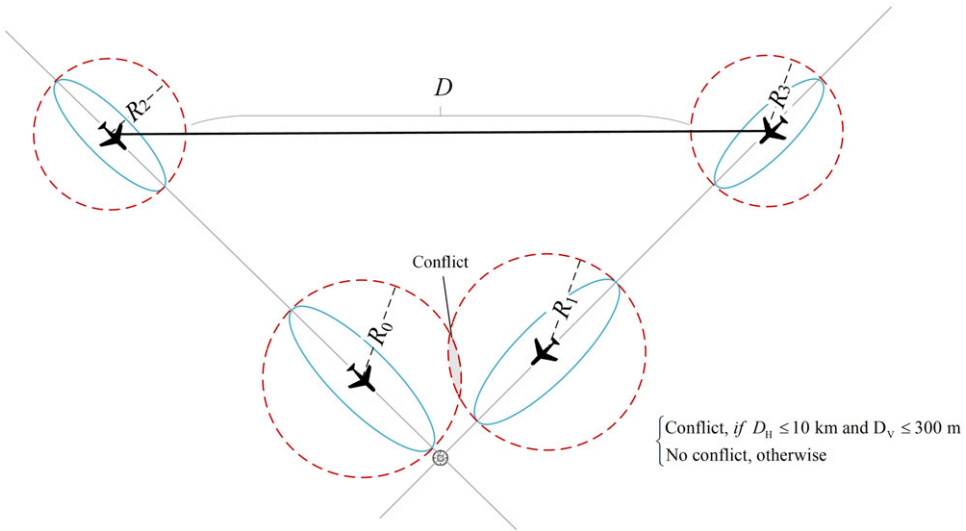


Figure 3. Conflict detection based on horizontal uncertainty.

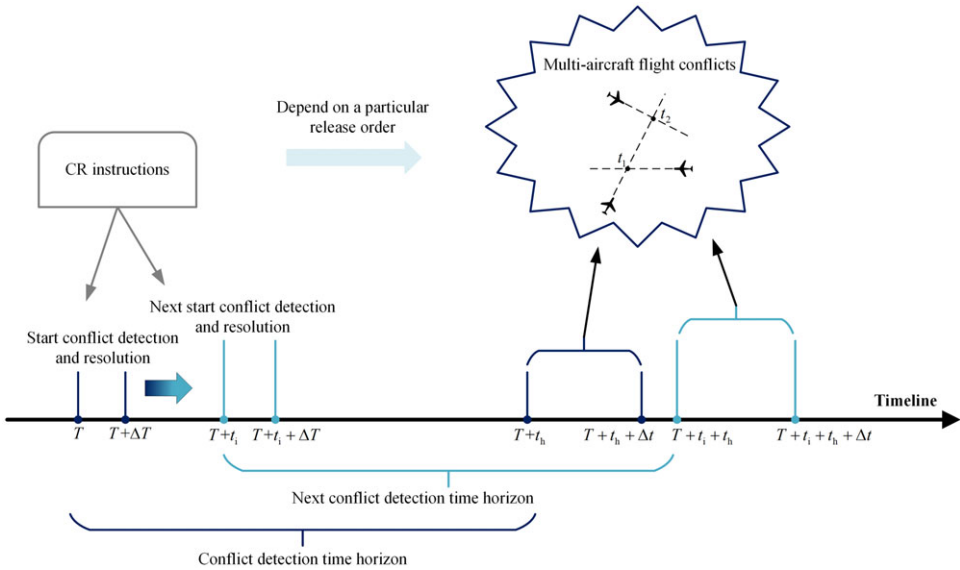


Figure 4. Conflict detect and resolution mechanism.

4.0 Flight conflict detection and resolution framework

This section introduces the framework of conflict detection and resolution, including the mechanism of conflict detection and resolution, the overall ATCOs decision-making framework, and finally our proposed priority-based ranking mechanism.

4.1 Conflict detection and resolution mechanisms in the continuous time domain

A well-developed conflict detection mechanism is essential for the early detection of real-time flight conflicts. This paper extends conflict detection along the temporal direction to detect multi-aircraft conflicts

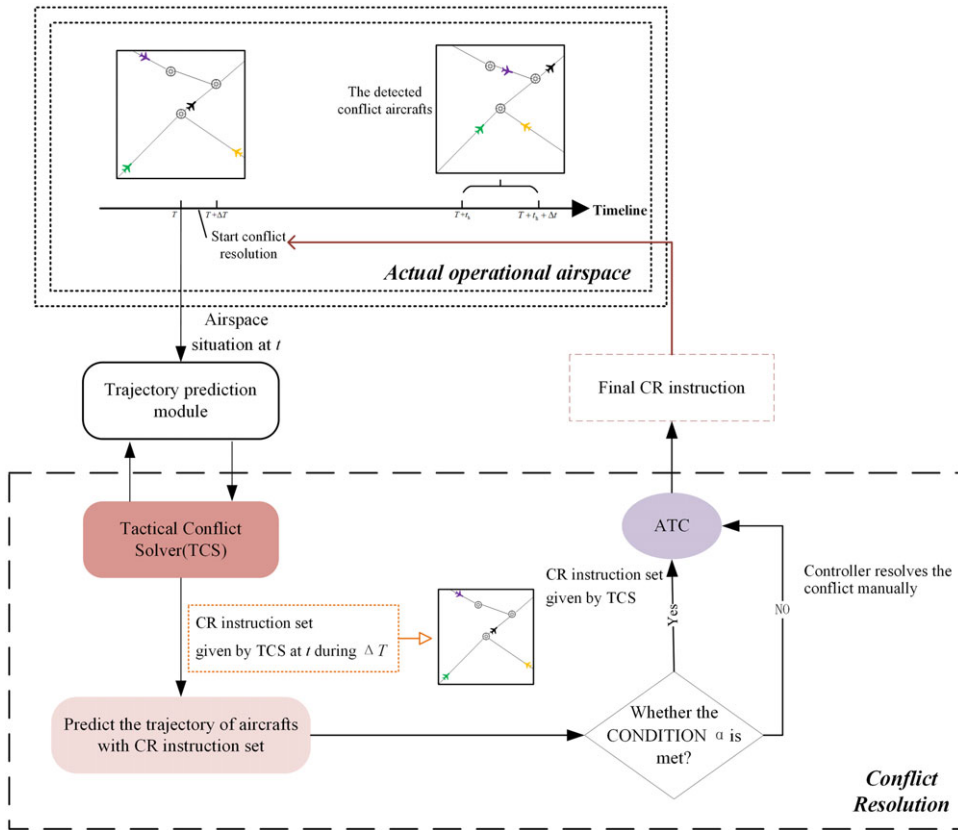


Figure 5. The framework of tactical conflict solver.

with spatio-temporal correlation. In the continuous time domain, assuming conflict detection starts at time T , Fig. 4 illustrates the conflict detection and resolution process. By utilising the trajectory speculation module for aircraft trajectory speculation in the airspace, conflicts occurring in $[T + t_h, T + t_h + \Delta t]$ can be detected. Here, Δt corresponds to Δt presented in Section 3.1, t_h is the advance detection time, t_i is the time interval for conflict detection. At this point, after a certain amount of time of calculation, the instruction to resolve the conflict can be given in ΔT time, and the conflicting aircraft executes the instruction after receiving it. After an interval of time t_i , the conflict detection mechanism continues with the subsequent detection, where t_i is usually set to be slightly larger than Δt , conflicts between the next start of conflict detection and the end of the previous conflict detection will be detected in subsequent detection intervals. This conflict detection mechanism on the continuous time domain can realise the comprehensive detection and resolution of flight conflicts in the airspace, which guarantees flight safety in the airspace.

4.2 The TCS framework

The tactical conflict solver (TCS) framework combines the conflict detection and resolution (CD&R) mechanisms mentioned above. As shown in Fig. 5, the airspace situation at the time T is passed to the aircraft trajectory prediction module for conflict detection. The identified flight conflict is transmitted to the conflict resolution module, which effectively resolves the conflict and generates a tailored aircraft deployment instruction as the final solution. The aircraft trajectory prediction module determines whether the conflict is resolved; if it is not resolved, the ATCOs manually carry out the conflict resolution deployment; if it has been resolved, the instructions obtained from the solution will be conveyed to

the corresponding aircraft under the ATCOs' supervision. The criteria $CONDITION\alpha$ for the success of conflict resolution is: From the moment conflict detection begins until 270s after the first two-aircraft conflict occurs, there should be no conflict between conflicting aircraft nor between conflicting and environmental aircraft. Under this framework, ATCOs can operate according to the TCS's recommendations, and their workload will be substantially reduced.

4.3 Priority-based ranking mechanism

Two primary approaches exist for resolving conflicts involving multiple flights: simultaneous resolution and sequential resolution. Simultaneous resolution entails coordinating conflicting aircraft to take simultaneous actions to address the flight conflict, often achieved through a multi-agent reinforcement learning framework. However, this approach faces challenges in coordinating actions among the conflicting aircraft, making it difficult to rectify the situation if the conflict persists after post-simultaneous resolution attempts. In such cases, aircraft that have already manoeuvred may require readjustment, which does not align with current air traffic control rule. This paper adopts the sequential resolution method, whereby conflicts between multiple flights are resolved by gradually deploying conflicting aircraft in a specific order, effectively addressing scalability concerns by sequentially resolving flight conflicts.

In practice, ATCOs typically prioritise flight conflict aircraft according to air traffic control rules. High-urgency aircraft are usually deployed first to resolve an imminent conflict, while low-urgency aircraft are deployed afterward, considering their previous deployments. The degree of urgency between two aircraft involved in a flight conflict is defined as the inverse of the time difference between the moment conflict detection begins and the moment of loss of safe separation. The higher the degree of urgency, the more the conflict must be dealt with immediately.

Define the set of conflicting aircraft in a multi-aircraft flight conflict scenario:

$$C = \{Acf_{t_1}, Acf_{t_2}, \dots, Acf_{t_n}\} \tag{4}$$

where n is the number of conflicting aircraft in a multi-aircraft conflict.

According to Section 3.1, the two-aircraft flight conflict matrix involved in a multi-aircraft flight conflict scenario is defined as:

$$P = \begin{bmatrix} Con_{11} & \dots & Con_{1n} \\ \vdots & \ddots & \vdots \\ Con_{n1} & \dots & Con_{nn} \end{bmatrix} \tag{5}$$

where Con_{ij} is the degree of urgency between conflict aircraft i and conflict aircraft j :

$$Con_{ij} = \begin{cases} 1/(t_{lose} - T), & \text{conflict between aircraft } i \text{ and aircraft } j \\ 0, & \text{no conflict between aircraft } i \text{ and aircraft } j \text{ or } i = j \end{cases} \tag{6}$$

where t_{lose} is the moment of loss of safe separation between aircraft i and aircraft j , T is the moment to start conflict detection.

Define the urgency of the k th conflicting aircraft as:

$$U_k = \sum_{j=1}^n Con_{kj}, k = 1 \dots n \tag{7}$$

After calculating the urgency of each conflicting aircraft, a set of ranked sequences is obtained by arranging them in descending order. Define three sequences based on this: 1. Sorting Sequence S_1 : Storing conflicting aircraft that have not yet executed a pre-command for conflict resolution; 2. Execution Instruction Sequence S_2 : Storing conflicting aircraft that have executed a conflict resolution pre-command; 3. Whether to Execute the Command Sequence S_{EXE} : False and True are filled into the queue depending on whether or not the conflict resolution pre-command is executed. The specific flow of conflict resolution according to the order of conflict resolution is shown in Table 2.

Table 2. Priority-based ranking mechanism for multi-aircraft flight conflict resolution

Algorithm: Multi-Aircraft Flight conflict resolution order based on priority

- 1 A two-aircraft flight conflict matrix is calculated based on the conflict relationships between the n conflicting aircraft in the multi-aircraft conflict scenario. From this, the urgency of each conflicting aircraft is calculated and ranked in descending order to form a sorting sequence S_1 reflecting the priority level. Set Execution Instruction Sequence $S_2 = \phi$. Whether to execute the command Sequence $S_{EXE} = \phi$. The collection of environment aircraft that restricts successful conflict resolution $S_{ENV} = \phi$. Initialise index = 0.
 - 2 Pick the first aircraft in Sorting Sequence S_1 , a conflict resolution model generates a pre-instruction, and a trajectory prediction module is used to determine whether a conflict associated with that aircraft will still exist after the pre-instruction is executed.
 - 3 While index < n :
 - If number of last several consecutive 'False' in $S_{EXE} = \text{len}(S_1)$:
 - Break ##Continued adjustment of conflicting aircraft in the remaining S_1 fails to resolve the conflict, and a dead loop occurs.
 - End if
 - If conflicts related to that aircraft still exist:
 - Withdrawal of the current aircraft pre-command, Add the False character to the S_{EXE} , Adjust the aircraft to the end of S_1 , Move to 2.
 - Else:
 - Execute the current pre-command, and move the selected aircraft out of the S_1 , transfer to S_2 , Add the True character to the S_{EXE} . Update the urgency of each conflicting aircraft in the S_1 and re-sort from most to least urgent. Set index = index + 1. If judged by the trajectory prediction module to have satisfied CONDITION α , Then move to 6, otherwise move to 2.
 - End if
 - End while
 - 4 For aircraft i in S_1 :
 - Generating pre-commands for aircraft i using a conflict resolution model, and the trajectory prediction module is used to determine whether a conflict related to aircraft i will still exist after the execution of this instruction. Determined by the trajectory prediction module, if Aircraft i is still in conflict with the conflicting aircraft after executing the pre-command:
 - Withdrawal of the pre-command for Aircraft i
 - Start selecting the next aircraft in S_1
 - continue
 - Elif Aircraft i is determined by the trajectory prediction module to be in conflict only with environmental aircraft after executing the pre-command:
 - Record the environmental aircraft that limit conflict resolution of aircraft i , generate pre-commands for those environmental aircraft in turn, and withdraw pre-commands of aircraft i . Only the first environmental aircraft that does not have a new conflict retains the pre-command; in all other cases, the command is withdrawn. The aim is to move away the environmental aircraft and free up the solution space while minimising the number of adjusted environmental aircraft. If there is no environmental aircraft that meets the conditions, withdrawal of the pre-command for Aircraft i , directly end this cycle and start selecting the next aircraft in S_1 . If there is an environmental aircraft that meets the conditions, it will perform the pre-command. After executing the pre-command, another pre-command is regenerated for aircraft i and executed. If CONDITION α is satisfied, move to 6. If CONDITION α is not satisfied and the remaining conflict in the airspace is still related to aircraft i , move to 5. If CONDITION α is not satisfied and the remaining conflict is not related to aircraft i , continue to the next loop.
 - End if
-

Table 2. Continued.

Algorithm: Multi-Aircraft Flight conflict resolution order based on priority

- 5 This multi-aircraft flight conflict scenario failed to resolve and was manually deployed by the ATCOs, and the scenario ended.
- 6 This multi-aircraft flight conflict scenario is successfully resolved. A queue of sequential resolution commands is generated, and the commands are passed to the ATCOs according to the resolution command sequence, and the scenario ends.

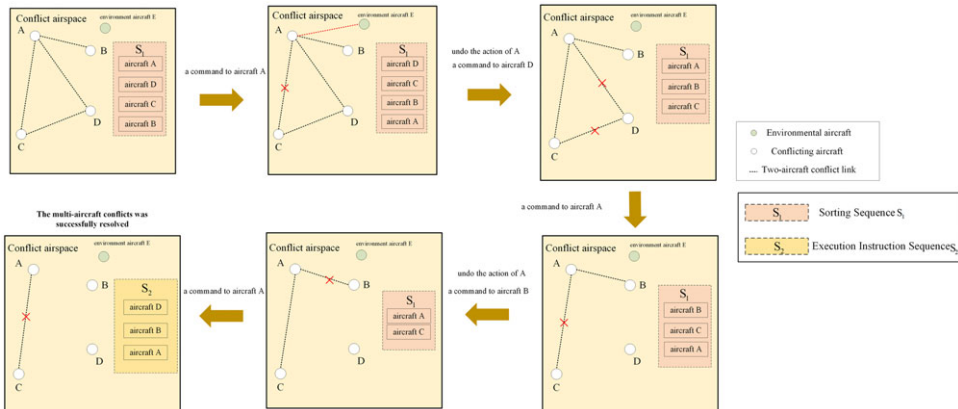


Figure 6. Schematic diagram of the conflict resolution process for a multi-aircraft flight.

Figure 6 gives a specific example. According to the initial sequence S_1 , the conflicting aircraft A should execute the command in priority, and the conflict resolution model is used to generate a pre-command for A. However, after the trajectory prediction module predicts that the pre-command of aircraft A leads to a conflict with the environmental aircraft and does not resolve all of two-aircraft conflict involving aircraft A, the pre-command of aircraft A is withdrawn, and the order of A is moved to the end of the sequence. Currently, the conflicting aircraft D is scheduled to generate pre-command first. The trajectory prediction module determines that aircraft D's pre-command resolves all conflicts it is involved in and does not create new conflicts, then aircraft D executes the pre-command, moves aircraft D out of the S_1 and into the S_2 , and updates the urgency matrix, recalculates the urgency for each of the conflicting aircraft, and sorts them. The foremost conflicting aircraft A in S_1 is selected to generate the resolution pre-command. However, aircraft A's pre-command only resolves the conflict between A-C and not between A-B, so it is still necessary to withdraw aircraft A's pre-command and place aircraft A at the end of S_1 . Next, the conflict aircraft B at the top of S_1 is selected, and the instructions generated using the conflict resolution model can resolve the conflict between A-B and no new conflict occurs, moving B out of the S_1 sequence and into the S_2 sequence and updating the S_1 ordering. At this time, after selecting the conflict aircraft A again to generate and execute the command, the conflicts in the airspace are all resolved, satisfying α , and this multi-aircraft flight conflict scenario is resolved successfully. This design of the conflict resolution sequence is dynamically changing with the resolution process, which aligns with the actual situation of ATCOs' resolution in real life. At the same time, this approach has the flexibility to adjust the resolution order with the airspace situation automatically and can successfully train the agent's resolution ability in TCS.

5.0 Flight conflict resolution model based on markov decision-making process

The flight conflict resolution process is Markovian and, therefore, modeled based on MDP. MDP needs to define tuples with five elements $\langle S, A, T_r, R, \gamma \rangle$, where S is the state space, A is the action space, T_r is the state transition function, R is the reward space, γ is the discount factor.

At each time step, the environment is in a specific state $s \in S$, and the agent performs the action $a \in A$ according to the policy. The environment then moves to the next state and gives agent a rewarding feedback $r(s, a)$. s' follows a state transfer probability function that satisfies the principle of Markovianity $p(s'|s, a)$. The ultimate goal of MDP modeling is to find a policy that specifies the agent's actions at each step and maximises the expectation of cumulative discount rewards $E[\sum_{t=0}^{T_0} \gamma^t r(s^t, a^t)]$ over time. Where T_0 is the time range of the MDP, $0 \leq \gamma \leq 1$ is the discount factor, s^t and a^t are the state and action at moment t . π represents strategy, the stochastic strategy can be expressed as $\pi(s):S \rightarrow A$, giving the probability of action in each state.

The optimal strategy for an MDP can be solved using reinforcement learning (RL), which defines two value functions to evaluate strategies with discounted long-term rewards. The first function, known as the state value function, measures the expected discounted rewards of starting from a state and following the strategy π after that:

$$V^\pi(s) = E_{a^t, s^{t+1}, \dots} [\sum_{\tau=0}^{T_0} \gamma^\tau r^{t+\tau+1} | s^t = s, \pi] \tag{8}$$

The other value function is called the action-value function (Q-function) and is defined as:

$$Q^\pi(s, a) = E_{s^{t+1}, a^{t+1}, \dots} [\sum_{\tau=0}^{T_0} \gamma^\tau r^{t+\tau+1} | s^t = s, a^t = a, \pi] \tag{9}$$

It measures the expected discounted reward for taking action a in specific state s and adopting strategy π .

The Bellman equation for the Q-function is:

$$Q^\pi(s^t, a^t) = \sum_{s^{t+1}} p(s^{t+1} | s^t, a^t) \times \left[r^{t+1} + \gamma \sum_{a^{t+1}} \pi(a^{t+1} | s^{t+1}) Q^\pi(s^{t+1}, a^{t+1}) \right] \tag{10}$$

This paper uses the model-free approach, and the transfer function T_r is derived using the trajectory prediction module. The discount factor γ is fixed, the state space, action space and reward function must be explicitly defined.

(1) Observation space:

The state space reflects the state of the target airspace at a particular moment in time and is a dynamic space. With each time increment, the agent derives its observations by examining the state space, resulting in observations that are either subsets of the state space or processed versions. The evolution of the state space is determined by the trajectory prediction module. The observation space of agent in this paper is an N-dimensional vector, N is defined as:

$$N = N_n \times N_t \times N_d \tag{11}$$

where N_n is the number of aircraft, including conflict and environmental aircraft. N_t is the number of discrete time periods, N_d is specific information about each aircraft at a given moment in time.

Define the midpoint of a two-aircraft conflict as the two-aircraft conflict point, then the conflict point in a multi-aircraft conflict is calculated as the average of the multiple conflict points derived from the respective two-aircraft conflicts. A 200 km \times 200 km \times 6000 m rectangular airspace is constructed centred on this value to describe the current observational range of the agent. Define the observation of the agent at the moment t to be O_t , the state of the corresponding rectangular airspace is S_t , O_t can be defined as:

$$O_t = (S_t, S_{t+180}, S_{t+360}, S_{t+540}, S_{t+720}) \tag{12}$$

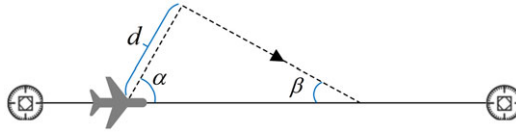


Figure 7. Schematic diagram of the dog-leg manoeuver.

where O_t included the state of the rectangular airspace from moment t to moment $t + 720$ in time intervals of 180s, corresponding to above N_t . Where S_t includes the information of 30 aircraft in the airspace at time t . If there are less than 30 aircraft in the airspace, the remaining element is set to 0, and the number of aircraft corresponds to the size of N_n above:

$$S_T = (Info_1^T, Info_2^T, \dots, Info_{30}^T), T = t, t + 180, t + 360, t + 540, t + 720 \tag{13}$$

where $Info_i^T$ is the specific flight information of the i th aircraft in the delimited airspace at time T , containing:

$$Info_i^T = (x_i^T, y_i^T, z_i^T, vspd_i^T, hspd_i^T, h_i^T, type_i^T, ur_i^T) \tag{14}$$

where the number of elements corresponds to N_d . The meanings are, respectively, longitude, latitude, altitude, rate of climb, horizontal speed, heading, type of aircraft (denoted by serial number), and the length of the long axis of the confidence ellipse of the i th aircraft at the time T . Here i is obtained by ordering the Euclidean distances to the aircraft executing the conflict resolution command, setting the aircraft executing the conflict resolution command as the first aircraft, the closest aircraft to that aircraft as the second aircraft, and so on. Furthermore, the data within the observation space were normalised.

(2) Action space:

The action space is the set of instructions that can be taken by a conflicting aircraft to execute a conflict resolution command, and the action space can be defined as:

$$A = \{cmd, T_w | T_w = 20x \cdot \text{seconds}, x \in [0, T_a]\} \tag{15}$$

where cmd represents a specific command executed by the aircraft, containing altitude adjustment, speed adjustment and heading adjustment. This paper addresses discrete actions. T_w represents the waiting time for the execution of the action. In practice, the executed instructions are usually not executed immediately but at the right time, so the delay in executing the conflict resolution action should be considered and discretised in units of 20s. T_a is an upper limit on the execution time of an instruction. We specify that the instruction must be executed before a conflict occurs. In addition, the heading conflict resolution manoeuver considered in this paper is dog-leg, the schematic diagram of which is shown in Fig. 7. The angle of departure α is used as a variable and takes the value of 30 or 60 degrees, the angle of return β is set to 30 degrees, and the side of departure is specified as 8 nautical miles. The cmd contains a collection of actions as shown in Table 3.

(3) Reward function:

The agent’s goal is to maximise long-term rewards and update the parameters of the neural network based on instantaneous rewards. The setting of the reward function directly affects conflict resolution. To effectively address the multi-aircraft conflict scenario outlined in this study, three essential conditions must be met: (i) the agent’s actions must adhere to air traffic control regulations, ensuring the avoidance of unreasonable actions; (ii) the manoeuvres performed by the agent must not create new conflicts and, simultaneously, not conflict with the environmental aircraft; and (iii) all conflicting aircraft work to resolve conflicts related to themselves first so that the entire multi-aircraft scenario endeavors to satisfy CONDITION α . We set the reward function as follows:

$$R = R_1 + R_2 + R_3 \tag{16}$$

Table 3. Table of commands

CR commands	Resolution action	Adjustment value
Altitude adjustment	Climbing or descending/m	300, 600, 900,1200
Speed adjustment	Acceleration or deceleration/kt	10, 20,30
Heading adjustment	Right or left offset/degree	30,60

where R_1 indicates whether the conflict resolution manoeuvre performed by the aircraft complies with the ATC regulations and a positive reward is given if it does. Otherwise, a negative reward is given. The ATC regulations here mainly mean that the relevant manoeuvres to be performed by the aircraft are under the aircraft's performance limits. In case of a manoeuvre that does not comply with the ATC regulations, the manoeuvre will not be performed while giving a negative reward. R_1 specifically expressed as:

$$R_1 = \begin{cases} +\phi, & \text{if action is valid} \\ -\phi, & \text{else} \end{cases} \quad (17)$$

R_2 aims at the conflicting aircraft doing the manoeuvre and denotes the number of two-aircraft conflicts involving that aircraft remaining after that aircraft has done the manoeuvre, which is *conflict_num*. Each two-aircraft conflict gives a negative feedback value of λ .

$$R_2 = \text{conflict_num} \times \lambda \quad (18)$$

The R_3 directly reflect whether the conflicting aircraft doing the action resolves all associated conflicts., defined as:

$$R_3 = \begin{cases} +\varphi, & \text{if solved successfully} \\ -\varphi, & \text{if solved failed} \end{cases} \quad (19)$$

(4)Termination state:

In this study, we define the conflict resolution process within a multi-aircraft scenario as a 'round'. Two termination states, or end-of-round conditions, are identified: ① the round ends after all conflicting aircraft in the training phase have followed their resolution instructions in the order of resolution, at which time resolution is judged successful if $\text{CONDITION}\alpha$ is satisfied; otherwise, resolution fails; ② before the last conflicting aircraft executes the instruction, the multi-aircraft conflict environment has already satisfied $\text{CONDITION}\alpha$, and then it is judged that the resolution is successful.

6.0 Acktr-based algorithm

In general, the application of deep reinforcement learning to solve MDP models exhibits favourable outcomes in terms of both performance and computational efficiency. While many deep reinforcement learning techniques rely on a simplified version of stochastic gradient descent (SGD) to train neural networks, SGD may prove inefficient in exploring the weight space, often necessitating several days for convergence in control tasks. In this investigation, we employ the actor-critic using the ACKTR algorithm [30], which leverages natural gradient descent principles. By utilising the Kronecker factor approximation of the natural policy gradient, ACKTR enables the effective inversion of the gradient's covariance matrix, enhancing computational efficiency.

The ACKTR algorithm is rooted in the actor-critic paradigm. Conventionally, the actor network is responsible for training the actor neural network, while the critic network focuses on training the critic neural network. The output of the actor network is characterised by a distribution, facilitating the straightforward definition of its Fisher information matrix. However, in contrast, the output of a standard critic network is a scalar rather than a distribution, rendering the definition of its Fisher information matrix unfeasible.

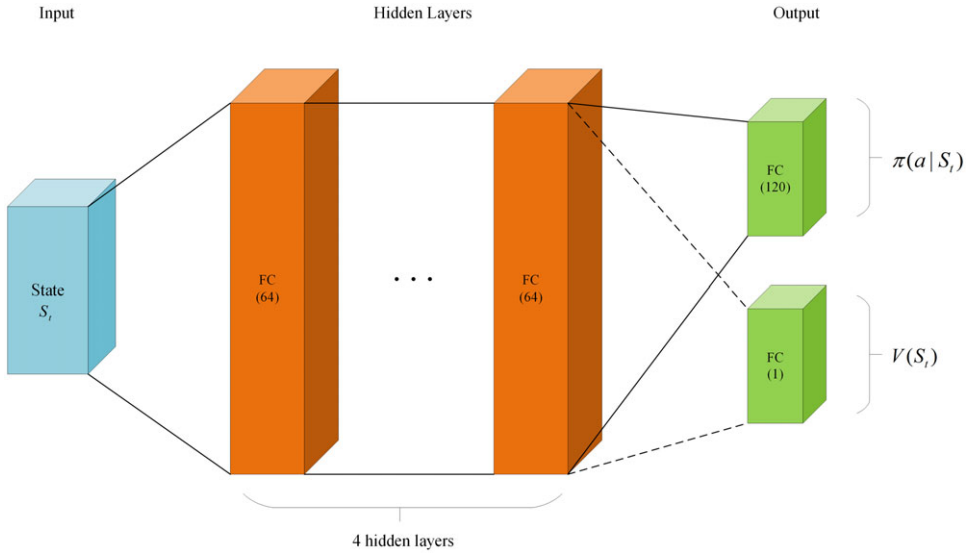


Figure 8. Neural network architecture for agent training.

The solution given by the ACKTR algorithm is that the actor network shares with the critic network a fully connected network with four hidden layers, as shown in Fig. 8 where the output of the critic network is defined to be a Gaussian distribution $p(v | S_t) \sim \mathcal{N}(v; V(S_t; \mathbf{w}), \sigma^2)$, which in practice σ can be simply set to 1. Finally, the respective fully connected layers are connected to obtain the policy and value, respectively, in order to update the model using the method of natural gradient descent, ACKTR defines the output of the overall network as $p(a, v | s) = \pi(a | s) p(v | s)$ by assuming independence of the policy and the value distribution. The proposed neural network structure can be viewed as a whole and the overall network is updated using a loss function that synchronises the update of the critic and actor.

The ACKTR algorithm uses a multi-threaded approach to train the agents. In each thread, the agent interacts with its environment, and the central model performs data aggregation and parameter updates. Let there be a total of l threads, each thread interacts k steps in each multi-threaded interaction, and collects a collection of empirical data, and $|N| = k \times l$.

The update formula for the overall network parameter θ is:

$$\theta \leftarrow \theta - \eta \mathbf{F}^{-1} \nabla_{\theta} \mathcal{J}(\theta) \tag{20}$$

where $\mathbf{F} = \mathbb{E}_{p(\tau)} [\nabla \ln p(a, v | s) \nabla \ln p(a, v | s)^T]$, $p(\tau)$ is the distribution of trajectories, given by $\prod_{t=0}^T \pi(A_t | S_t) p(S_{t+1} | S_t, A_t)$, \mathcal{J} is the function to be optimised.

The loss function is defined as shown in the equations (21)–(24):

$$pgLoss = \frac{1}{|N|} \sum_{n \in N} (R_{t+1} + \gamma V(S_{t+1}; \theta) - V(S_t)) \times (-\ln \pi(A_t | S_t; \theta)) \tag{21}$$

$$vfLoss = \frac{1}{2} \sum_{n \in N} (V(S_t; \theta) - R_{t+1})^2 \tag{22}$$

$$entropy = \sum_{a \in \mathcal{A}} \pi(a | S_t; \theta) \ln \pi(a | S_t; \theta) \tag{23}$$

$$\mathcal{L} = pgLoss - e_coef \times entropy + v_coef \times vfLoss \tag{24}$$

where e_coef and v_coef are constant coefficients.

Table 4. ACKTR algorithm flow for training the TCS agent

Algorithm: ACKTR algorithm for training the TCS agent

Initialise parameter of the overall network θ_0

Loop through iterations $1, 2, \dots, M$.

Each thread randomly select a conflict scenario sample and initialise the agent, observe O_0

Loop through the steps $1, 2, \dots, T$.

Each thread individually generates trajectories with the policy $\pi(\theta)$ $O_0, A_0, R_1, O_1, A_1, R_2, \dots, O_t$.

Until the episode **end**.

Aggregate the data from each thread and calculate the loss function according to Equation (24).

For overall network layers $i = 0, 1, 2, \dots$ **do**:

$\text{vec}(\Delta\theta_k^i) = \text{vec}(X_i^{-1} \nabla_{\theta_k^i} \mathcal{L}(\theta_k^i) Z_i^{-1})$, In the calculation of X_i^{-1} and Z_i^{-1} , $L = \ln p(a, v | s)$.

End for.

The K-FAC approximation of the natural gradient updates parameters of the overall network:

$$\theta_{k+1} = \theta_k + \eta_k \Delta\theta_k$$

Until the iterations **end**.

Table 4 shows the specific flow of the ACKTR algorithm combined with the conflict resolution environment. It is worth noting that for the ACKTR algorithm using multiple threads, many conflicting training samples should be generated for the agent to learn. Where η_k for the update parameter inside the table is set to $\min(\eta_{\max}, \sqrt{\frac{2\delta}{\Delta\theta_k^T \hat{\mathbf{F}}_k \Delta\theta_k}})$ [31] η_{\max} is the learning rate and δ is the trust region, each piece in chunked diagonal matrix $\hat{\mathbf{F}}_k$ corresponds to the Fisher information matrix for each layer of the overall network: $\hat{\mathbf{F}}_k^i = X_i \otimes Z_i$. Let the input to its i th layer ($0 \leq i \leq m$) be x_i , and the output before activation be z_i .

7.0 Simulation experiments

1. Airspace scenario setting:

As shown in Fig. 9, the airspace around HFE, the busiest route point in China in 2018, is selected as the simulation scenario in this paper. The source of airspace, waypoint coordinates, route structure, airport coordinates and other data is the 2018 Aeronautical Information Publication (AIP). A $400 \text{ km} \times 400 \text{ km} \times 6 \text{ km}$ rectangular airspace (i.e. airspace A) is generated in the scenario as a specific airspace for traffic flow simulation; a $200 \text{ km} \times 200 \text{ km} \times 6 \text{ km}$ rectangular airspace (i.e. airspace B) is delineated therein, and multi-aircraft conflicts occurring in this airspace are recorded, thus generating multi-aircraft conflict scenario samples. Traffic flow data was generated from flight plans flying over the airspace on 1 June 2018, derived from the 2018 National Aeronautical Information Publication (NAIP).

2. Experimental environment:

The Air Traffic Operation Simulation System (ATOSS) serves as the designated deep reinforcement learning environment for training agents in this study. This environment facilitates the generation of multi-aircraft flight conflict scenarios, trajectory prediction, conflict detection, and state space transfer. ATOSS is independently developed by the New Generation Intelligent ATC Laboratory of Nanjing University of Aeronautics and Astronautics (NUAA). It integrates with the airspace database and incorporates a simulation motion engine based on the base of aircraft data (BADA) [32] to simulate the aircraft's operation posture. BADA includes the performance data sheets of various aircraft types, enabling the evaluation of whether resolution manoeuvres align with flight performance criteria.

The hardware environment for the experiment is : HP Z8-G4 workstation (configured with one Intel Xeon(R) Gold 6242 CPU and 64GB RAM). Software environment: IntelliJ Pycharm as IDE, Python language, deep learning framework-Tensorflow 1.14.0.

Table 5. Hyper-parameter value of ACKTR

Hyper-parameters	Parameters value	Hyper-parameters	Parameters value
Total training steps	6,000,000	σ	0.0005
Discount factor	0.99	e_coef	0.8
Number of threads	32	v_coef	0.3
Interaction steps per thread	5	Learning rate	1e-3

Table 6. Other-parameter value

Parameters	Parameters value	Parameters	Parameters value
t_i	240 s	ϕ	0.6
t_h	360 s	λ	-5
Δt	210 s	φ	10
T_a	180 s		

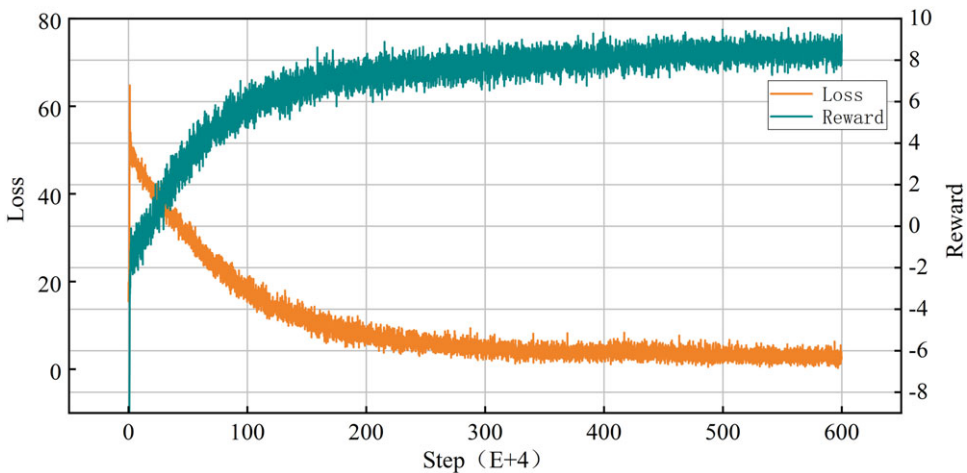


Figure 10. Loss and reward.

the other parameters are shown in Table 6, and their values were determined based on several comparative experimental analyses.

5. Results Analysis:

Randomly loading 20–30 aircraft, including conflict aircraft as well as environmental aircraft, the training environment was constructed in such a way as to train the agent’s ability to resolve conflicts under challenging scenarios. These aircraft are flown along actual flight paths in the ATOSS simulation system, an approach that trains the conflict resolution competence of aircraft from simulated situations.

After 6,000,000 steps, the loss value and reward value changes during training are shown in Fig. 10. The loss value decreases rapidly in the first 2,000,000 steps and converges gradually. The reward value gradually increases and finally converges around 9. This trend signifies a satisfactory training outcome for the model.

As shown in Fig. 11, after less than 150 h of training, the ACTKR algorithm can achieve a 94% success rate with low variance and high stability on the training set. The DQN algorithm, on the other hand, is difficult to deal with such more complex task environments, and its success rate is more volatile. Notably, the success rate pertains to the proportion of successfully resolved multi-aircraft conflict scenarios relative to the total number of training samples.

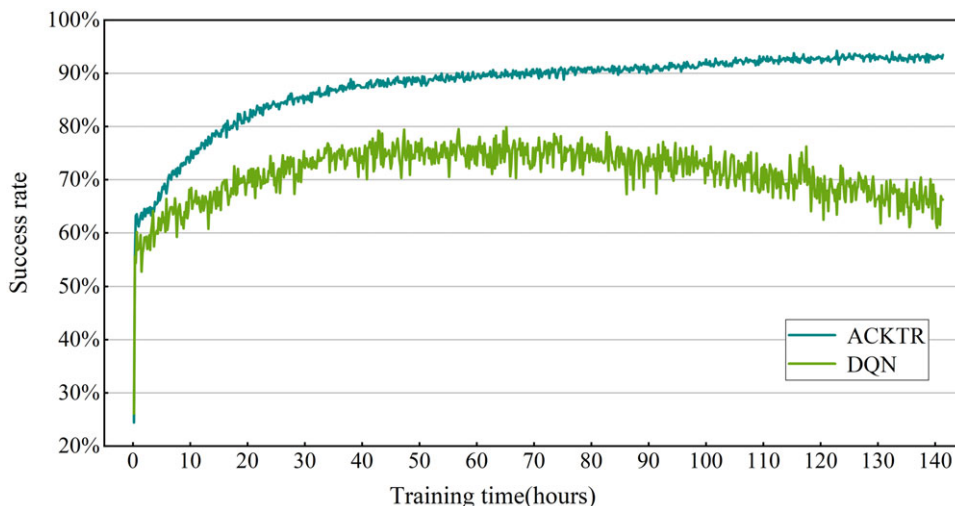


Figure 11. Success rate of DQN and ACKTR on the same training set.

Various factors influence the success rate of aircraft in conflict resolution. One significant determinant is the environment within which the aircraft operate. The most direct influence is the number of aircraft in the airspace, which is a dynamic indicator of the airspace. A higher aircraft density leads to a more constrained solution space for the agent, thus intensifying the complexity of conflict resolution tasks. In addition, static metrics of the airspace can affect the difficulty of conflict resolution, including the number of routes in the airspace, the number of waypoints and the structural complexity of the airspace. This study analyses how the number of aircraft in the airspace affects the success rate based on fixed airspace. The effect of weather factors on success rate is not considered for the time being.

The number of aircraft in the airspace can be adjusted manually, and the variation of success rate with the number of aircraft in the airspace is investigated for certain airspace static metrics and initial horizontal speeds. The model obtained from the training phase was tested on 1,000 test set samples. The two-aircraft success rate is defined as the number of successfully resolved two-aircraft conflicts as a proportion of the total number of two-aircraft conflicts, and the multi-aircraft flight success rate is defined as before, i.e. as the proportion of successfully resolved multi-aircraft samples to the total number of samples. To study the relationship between the number of aircraft in the airspace and the success rate, test results were obtained as shown in Fig. 12, where 20–30 aircraft in the airspace were the same as the number of aircraft in training, and the other scenarios were when the number of environmental aircraft was reduced accordingly. The results show that the obtained conflict resolution model can effectively solve multi-aircraft flight conflicts and is scalable. Reducing the number of aircraft in the airspace will be conducive to successfully resolving multi-aircraft flight conflicts.

In Section 4.3 of this study, a prioritisation mechanism is delineated, and the effectiveness of the mechanism needs to be proved by further comparative experiments. The efficacy of this proposed mechanism is assessed by integrating it with two different ranking approaches: a priority-based dynamic ranking mechanism and a random ranking mechanism. The multi- and two-aircraft success rates on 1,000 test sets are shown in Fig. 13. The results prove that after coordinating conflicting aircraft through the dynamic ranking mechanism, the success rate is significantly improved compared with the random order, and the resolution effect is satisfactory.

Corresponding to the number of aircraft in the airspace, the statistical distribution of the initial horizontal mean speed of aircraft in the airspace is shown in Fig. 14. The average initial horizontal speed of aircraft in the airspace also belongs to the dynamic indicators of the airspace. The aircraft's operability is highly correlated with the horizontal speed, and the larger the average speed, the more difficult it

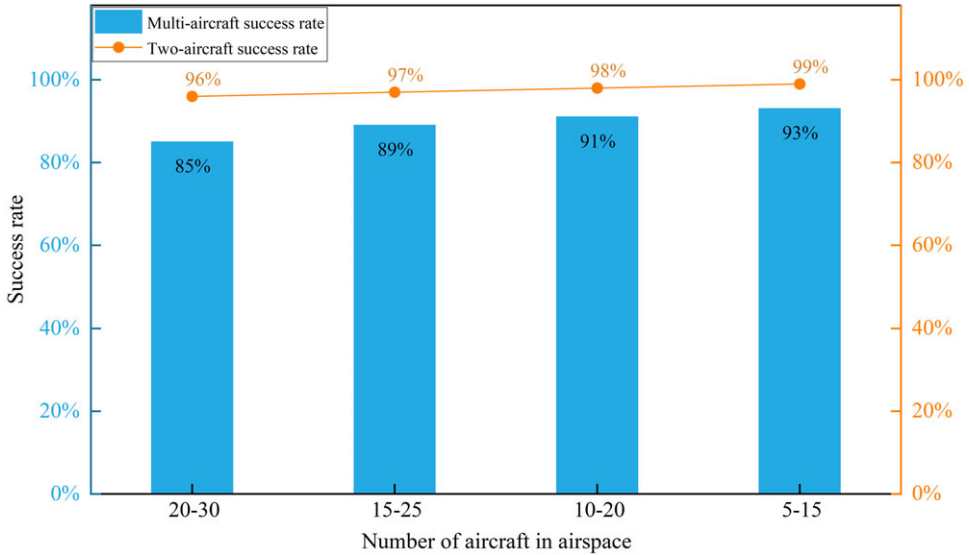


Figure 12. Success rates for different numbers of aircraft in the airspace.

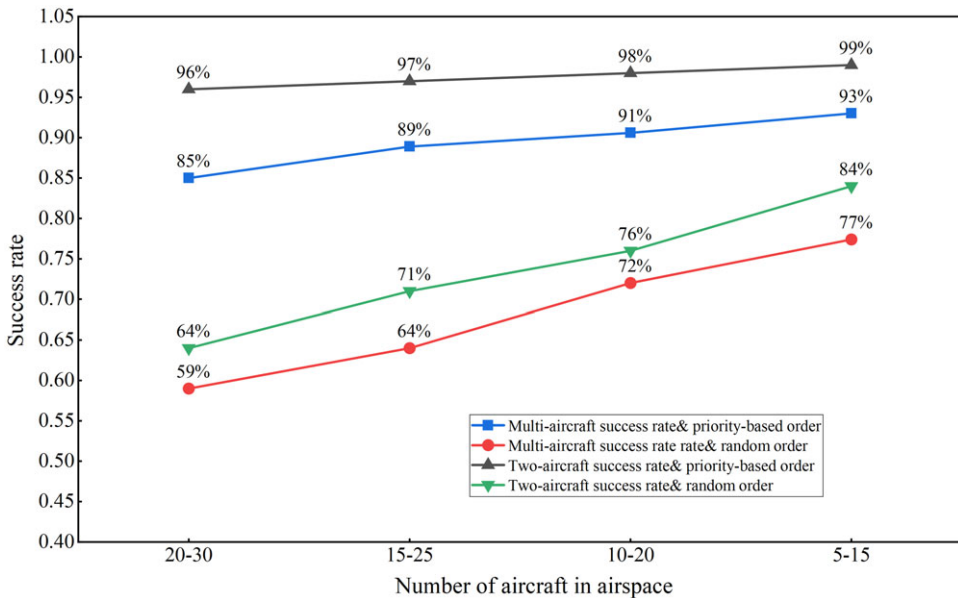


Figure 13. Success rate of dynamic order and random order on the same training set.

will be to resolve the conflict samples. So, the magnitude of the aircraft’s horizontal speed must also be considered when calculating the success rate.

Under the condition that the number of aircraft in the airspace is 20-30, the statistics of the conflicting aircraft selecting the conflict resolution command are shown in Fig. 15, and the TCS agents are more inclined to select the altitude adjustment manoeuvre after 6,000,000 steps of training. In 3D space, altitude adjustment is a timely and effective conflict resolution command. Agents think that speed and

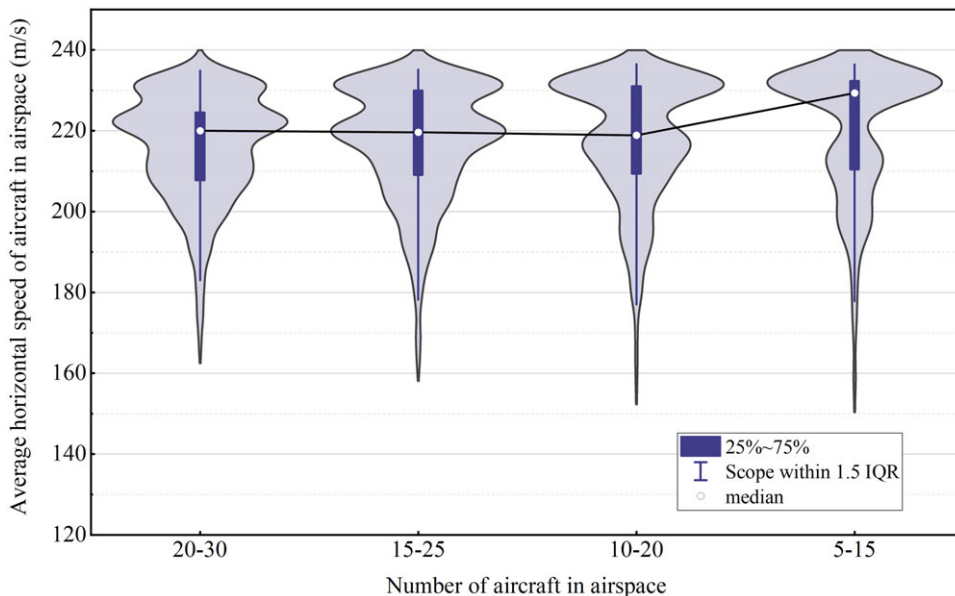


Figure 14. Mean horizontal speed distribution of aircraft in airspace.

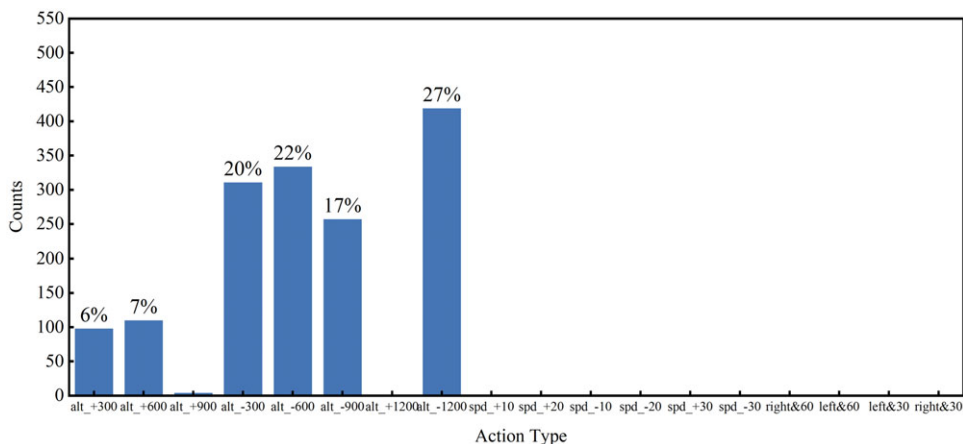


Figure 15. Distribution of conflict resolution actions.

heading adjustments are less effective than altitude adjustments, so they have fewer choices for speed and heading adjustments.

The results obtained after counting the number of conflict resolution instructions for a sample of 1,000 test sets are shown in Table 7. Over half of the multi-aircraft conflicts can be resolved with only one or two commands, which is the favoured resolution method for ATCOs and pilots. For more complex samples of multi-aircraft conflicts, more than two aircraft is required to execute commands for resolution.

After counting the test set samples, the vast majority of the computation time for TCS to give resolution suggestions is within 2 s, which aligns with the time requirement for conflict resolution. While traditional methods, such as genetic algorithms, to solve the multi-aircraft conflict problem may require tens of seconds of solution time, which could be less conducive to timely conflict resolution. Overall,

Table 7. *Minimum resolution action distribution*

Number of conflict resolution actions	One action	Two actions	Three or more actions
Proportions	56.1%	34.3%	9.6%

the model is fast to solve on the test set, the solution is reasonably effective, and it is consistent with the actual air traffic control situation.

8.0 Discussion

After the results are analysed, the flight conflict resolution model based on priority and MDP can solve most multi-aircraft flight conflict scenarios. The priority-based ranking mechanism adeptly coordinates the resolution of such conflicts, mitigating the issue of escalating dimensionality in multi-agent extensions. Notably, it navigates uncertainties while significantly reducing solution times to mere seconds.

Despite its successes, the model exhibits certain limitations. In practical air traffic control contexts, a 100% success rate is imperative for ensuring safety, a benchmark the model does not meet. Moreover, as the airspace's aircraft density increases, the model's conflict resolution efficacy demonstrates a fluctuating trend. Addressing these challenges necessitates the adoption of advanced DRL algorithms, refined ranking mechanisms, and adjustments to enhance environmental adaptability. In addition, the results show that aircraft are more inclined to choose altitude adjustment for conflict resolution. In practical situations, ATCOs may have their resolution habits and may not always fully accept the resolution advice given by the TCS. To address this, empirical statistics on ATCO behaviours within specific sectors can inform reward function modifications, rendering deployment instructions more aligned with human-centred needs.

In general, using the framework of priority and DRL to solve multi-flight conflicts is achievable, and by comparing it with the DQN algorithm, the ACKTR algorithm has a significant advantage in solving effectiveness. Compared with traditional conflict resolution methods, the framework excels in real-time responsiveness, global applicability and simulation capabilities. In contrast to multi-agent conflict resolution approaches, it overcomes scalability issues, possesses fault-tolerance capacities and enhances resolution outcomes.

9.0 Conclusion

This study introduces a tactical deployment method for multi-aircraft flight conflict resolution by integrating a priority-based conflicting aircraft ranking mechanism with an MDP-based conflict resolution model. The CDR framework efficiently identifies conflicts within airspace, addressing temporally and spatially correlated multi-aircraft conflict scenarios that pose significant challenges. Incorporating considerations for uncertainty enhances the model's applicability in real airspace settings. The proposed ranking mechanism fully considers the air traffic control rules and the changes in conflict resolution. It is capable of arranging a suitable resolution sequence for conflicting aircraft. The ACKTR algorithm used in the paper is a stable and novel DRL algorithm, which belongs to the DRL based on policy gradient. After 6,000,000 training steps, both the reward function and the loss value can converge, and the success rate is as high as 94%. Evaluation in testing phases, accounting for varying aircraft numbers within a fixed airspace, reveals the framework's rapid conflict resolution capabilities, inclination towards altitude adjustments and high success rates.

Future work could start by considering robust optimisation in measuring aircraft trajectory uncertainty to more comprehensively reflect aircraft position uncertainty; drawing insights from historical

air traffic control radar data can inform the refinement of reward functions to better align with individual ATCO deployment preferences, thereby augmenting ATCO's acceptance of TCS recommendations. Furthermore, expanding constraints and incorporating game theory principles can enhance the ranking mechanism's rationality. Lastly, the integration of advanced DRL algorithms holds promise for further optimising the success rate of the proposed framework.

Competing interests. The authors declare none.

References

- [1] Civil Aviation of China. Civil Aviation of China February 2023 Key Production Indicators Statistics, 2023.
- [2] Eurocontrol. European ATM Master Plan, 2020.
- [3] Pelegrín, M. and D'Ambrosio, C. Aircraft deconfliction via mathematical programming: Review and insights, *Transport. Sci.*, 2022, **56**, (1), pp 118–140.
- [4] Dias, F.H.C., Hijazi, H. and Rey, D. Disjunctive linear separation conditions and mixed-integer formulations for aircraft conflict resolution, *Eur. J. Oper. Res.*, 2022, **296**, (2), pp 520–538.
- [5] Liu, X. and Xiao, G. Multi-aircraft flight conflict resolution and trajectory recovery scheme based on mixed integer linear programming and geometric rules, *Transport. Res. Rec. J. Transport. Res. Board*, 2022, **2677**, (5), pp 166–182.
- [6] Tang, X.-M., Chen, P. and Li, B. Optimal air route flight conflict resolution based on receding horizon control. *Aerosp. Sci. Technol.*, 2016, **50**, pp 77–87.
- [7] Liu, W., Liang, X., Ma, Y. and Liu, W. Aircraft trajectory optimization for collision avoidance using stochastic optimal control, *Asian J. Control*, 2018, **21**, (5), pp 2308–2320.
- [8] Pei, H., Guanlong, L., Haotian, N., Yang, X. and Cunbao, M. Research on civil aircraft conflict resolution based on TBO, In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2019, pp 52–56.
- [9] Sui, D., Zhang, K. and Rey, D. A tactical conflict detection and resolution method for en route conflicts in trajectory-based operations, *J. Adv. Transp.*, 2022, **2022**, pp 1–16.
- [10] Pham, D.T., Tran, N.P., Goh, S.K., Alam, S., Duong, V. and IEEE. Reinforcement learning for two-aircraft conflict resolution in the presence of uncertainty, In *IEEE - RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2019.
- [11] Tran, P.N., Pham, D.-T., Goh, S.K., Alam, S. and Duong, V. An interactive conflict solver for learning air traffic conflict resolutions, *J. Aerosp. Inf. Syst.*, 2020, **17**, (6), pp 271–277.
- [12] Pham, D.-T., Tran, P.N., Alam, S., Duong, V. and Delahaye, D. Deep reinforcement learning based path stretch vector resolution in dense traffic with uncertainties. *Transport. Res. C Emerg. Technol.*, 2022, **135**, p 103463.
- [13] Sui, D., Ma, C. and Wei, C. Tactical conflict solver assisting air traffic controllers using deep reinforcement learning, *Aerospace*, 2023, **10**, (2), p 182.
- [14] Sui, D., Xu, W. and Zhang, K. Study on the resolution of multi-aircraft flight conflicts based on an IDQN. *Chin. J. Aeronaut.*, 2022, **35**, (2), pp 195–213.
- [15] Dalmau, R. and Allard, E. Air traffic control using message passing neural networks and multi-agent reinforcement learning, In *10th SESAR Innovation Days (SID)*, 2020.
- [16] Lai, J., Cai, K., Liu, Z. and Yang, Y. A multi-agent reinforcement learning approach for conflict resolution in dense traffic scenarios, In *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021.
- [17] Brittain, M.W. and Wei, P. One to any: Distributed conflict resolution with deep multi-agent reinforcement learning and long short-term memory, *AIAA Scitech 2021 Forum*. 2021, pp 1952 (1910 pp.)–1952 (1910 pp).
- [18] Brittain, M. and Wei, P. Scalable autonomous separation assurance with heterogeneous multi-agent reinforcement learning, *IEEE Trans. Autom. Sci. Eng.*, 2022, **19**, (4), pp 2837–2848.
- [19] Isufaj, R., Aranega Sebastia, D. and Angel Piera, M. Toward conflict resolution with deep multi-agent reinforcement learning. *J. Air Transport.*, 2022, **30**, (3), pp 71–80.
- [20] Huang, C., Petrulin, I. and Tsourdos, A. Strategic conflict management for performance-based urban air mobility operations with multi-agent reinforcement learning, In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022.
- [21] Chen, Y., Hu, M., Yang, L., Xu, Y. and Xie, H. General multi-agent reinforcement learning integrating adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution. *Transport. Res. C Emerg. Technol.*, 2023, **151**.
- [22] Papadopoulos, G., Bastas, A., Vouras, G.A., Crook, I., Andrienko, N., Andrienko, G. and Cordero, J.M. Deep reinforcement learning in service of air traffic controllers to resolve tactical conflicts. *Expert Syst. Appl.*, 2024, **236**, p 104125.
- [23] Sheng, L., Egorov, M. and Kochenderfer, M. Optimizing collision avoidance in dense airspace using deep reinforcement learning arXiv. 2019, pp 10.
- [24] Zhao, P. and Liu, Y. Physics informed deep reinforcement learning for aircraft conflict resolution, *IEEE Trans. Intell. Transp. Syst.*, 2022, **23**, (7), pp 8288–8301.
- [25] Zhang, M., Yu, J., Zhang, Y., Wang, S. and Yu, H. Flight conflict resolution during low-altitude rescue operation based on ensemble conflict models. *Adv. Mech. Eng.*, 2017, **9**, (4), p 168781401769665.
- [26] Wang, W.Y. and IOP, A satisficing game theoretic approach to multi-aircraft collision avoidance, In *5th International Conference on Electrical Engineering, Control and Robotics (EECR)*, 2019.

- [27] Wu, M., Yang, W., Bi, K., Wen, X., Li, J. and Ghenaiet, A. Conflict resolution strategy based on flight conflict network optimal dominating set, *Int. J. Aerosp. Eng.*, 2022, **2022**, pp 1–19.
- [28] Ho, F., Geraldes, R., Gonçalves, A., Rigault, B., Sportich, B., Kubo, D., Cavazza, M. and Prendinger, H. Decentralized multi-agent path finding for UAV traffic management, *IEEE Trans. Intell. Transp. Syst.*, 2022, **23**, (2), pp 997–1008.
- [29] Lauderdale, T. Probabilistic conflict detection for robust detection and resolution, In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012.
- [30] Wu, Y., Mansimov, E., Liao, S., Grosse, R. and Ba, J. Scalable Trust-Region Method for Deep Reinforcement Learning using Kronecker-Factored Approximation, 2017.
- [31] Ba, J., Grosse, R. and Martens, J. Distributed Second-Order Optimization using Kronecker-Factored Approximations, 2016.
- [32] Eurocontrol. User Manual for the Base of Aircraft Data (BADA) Revision 3.11, 2013.