Study on the resolution of multi-aircraft flight conflicts based on an IDQN

基于 IDQN 的多机飞行冲突解决研究

Dong SUI, Weiping XU*, Kai ZHANG

蒋遂，徐卫平 *，张凯

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

南京航空航天大学民航学院，南京 211106，中国

KEYWORDS

关键词

Abstract With the rapid growth of flight flow, the workload of controllers is increasing daily, and

随着航班流量的快速增长，管制员的日常工作量日益增加，

Air traffic control; handling flight conflicts is the main workload. Therefore, it is necessary to provide more efficient Conflict resolution; conflict resolution decision-making support for controllers. Due to the limitations of existing meth-Multi-agent system; ods, they have not been widely used. In this paper, a Deep Reinforcement Learning (DRL) algo-Multi-aircraft flight conflict; rithm is proposed to resolve multi-aircraft flight conflict with high solving efficiency. First, the Reinforcement learning characteristics of multi-aircraft flight conflict problem are analyzed and the problem is modeled based on Markov decision process. Thus, the Independent Deep Q Network (IDQN) algorithm is used to solve the model. Simultaneously, a 'downward-compatible' framework that supports dynamic expansion of the number of conflicting aircraft is designed. The model ultimately shows convergence through adequate training. Finally, the test conflict scenarios and indicators were used to verify the validity. In 700 scenarios, 85.71% of conflicts were successfully resolved, and 71.51% of aircraft can reach destinations within 150 s around original arrival times. By contrast, conflict resolution algorithm based on DRL has great advantages in solution speed. The method proposed offers the possibility of decision-making support for controllers and reduce workload of controllers in future high-density airspace environment.

空中交通管制；处理飞行冲突是主要的工作负担。因此，有必要为管制员提供更高效的冲突解决；冲突解决决策支持。由于现有方法的局限性，它们尚未得到广泛应用。在本文中，提出了一种深度强化学习 (DRL) 算法，用于高效解决多机飞行冲突。首先，分析了多机飞行冲突问题的特点，并基于马尔可夫决策过程构建了问题模型。因此，使用独立深度 Q 网络 (IDQN) 算法解决该模型。同时，设计了一个支持动态扩展冲突飞机数量的“向下兼容”框架。经过充分的训练，模型最终显示出收敛性。最后，使用测试冲突场景和指标验证了有效性。在 700 个场景中，85.71% 的冲突得到了成功解决，71.51% 的飞机能够在 150 s 原预计到达时间左右到达目的地。相比之下，基于 DRL 的冲突解决算法在解决速度上具有很大优势。提出的方法为未来高密度空域环境下管制员的决策支持提供了可能性，并有望减轻管制员的工作负担。

1. Introduction

"## 1. 引言"

In recent years, with the rapid development of China's civil aviation industry, air traffic flow has continued to increase. Production and hosting by Elsevier In 2019 China Civil Aviation Industry Development Statistical Bulletin, [1] the civil aviation industry transported 659,934,200 passengers in 2019, an increase of 7.9% over the previous year. China's civil aviation airports handled 11,660,500 takeoffs and landings in 2019, an increase of 5.2% over the previous year. From 2015 to 2019, the numbers of passengers carried and airport takeoffs and landings in civil aviation in China both increased year by year, with the average growth rate reaching 11.6% and 8.0% , respectively. In China' s civil aviation airspace, resources are limited, and the continuous growth of flight flow will lead to airspace congestion, which will increase the possibility of conflicts occurring in the operation of aircraft.

" 近年来，随着中国民航业的快速发展，空中交通流量持续增加。根据爱思唯尔发布的《2019 年中国民航行业发展统计公告》，[1] 2019 年民航业共运输了 6 亿 5993 万 4200 名乘客，同比增长 7.9%。2019 年

中国民航机场共完成起降 1166 万零 5 百次，同比增长 5.2%。从 2015 年至 2019 年，中国民航的旅客运输量和机场起降次数均逐年增长，平均增长率分别达到 11.6% 和 8.0% 。在中国民航空域中，资源有限，航班流量的持续增长将导致空域拥堵，这会增加飞机运行中出现冲突的可能性。”

The control sector is the basic unit of air traffic control, and the service capacity of the control sector is usually limited by the controller’s workload. When an increasing flight flow gradually saturates the sector, the traditional measure is to share the workload of controllers by dividing the sector. However, the size of the sector has a critical state. When its size is too small, the control handoff load will be greater than the control command load, so that the effectiveness of sector segmentation is lost. According to the current trend of air traffic development, the existing sector operation mode will not be sustained in the future. At present, the controller’s workload mainly comes from the identification and resolution of flight conflicts. There is much auxiliary equipment for conflict identification, while conflict resolution depends mainly on the controller’s decision. At the same time, with the rapid development of Unmanned Aircraft System (UAS), the number of aircraft in the airspace continues to increase. How to ensure the safe and efficient flight of unmanned and manned aircraft in limited airspace is also important. [2] This virtually increases the workload of the controller. To cope with the continuous increase of flight flow and reasonably reduce the controller’s workload, intelligent conflict resolution methods should be studied to provide the controller with decision-making support in line with actual operation. In actual operation, there are both two-aircraft conflict and multi-aircraft conflict in the control sector. In the multi-aircraft flight conflict resolution problem, the aircraft in the sector are in rapid motion, which requires the time to obtain the conflict resolution strategy as quickly as possible, and the joint cooperation of multiple aircraft is required to avoid conflict. Therefore, conflict resolution strategy must be efficient and coordinated on the basis of satisfying control regulations.

控制扇区是空中交通管制的基本单元，控制扇区的服务能力通常受限于管制员的负荷。当不断增加的航班流量逐渐使扇区饱和时，传统的做法是通过分割扇区来分担管制员的负荷。然而，扇区的大小存在一个临界状态。当扇区过小时，控制交接负荷将大于控制指令负荷，从而导致扇区分割的有效性丧失。根据当前航空交通发展趋势，现有的扇区运行模式在未来将无法持续。目前，管制员的负荷主要来自识别和解决飞行冲突。虽然有许多辅助设备用于识别冲突，但冲突的解决主要依赖于管制员的决策。同时，随着无人机系统 (UAS) 的快速发展，空中的飞机数量持续增加。如何在有限的空域内确保无人机和有人机的安全高效飞行也至关重要。这实际上增加了管制员的负荷。为了应对航班流量的持续增加并合理地减轻管制员的负荷，应研究智能冲突解决方法，以在实际运行中为管制员提供符合实际操作的决策支持。在实际操作中，控制扇区内既有两机冲突也有多机冲突。在多机飞行冲突解决问题中，扇区内的飞机处于快速运动状态，这要求尽快获取冲突解决策略，并且需要多架飞机的协同合作以避免冲突。因此，冲突解决策略在满足控制规定的基础上必须高效且协调。

By analyzing the characteristics of the multi-aircraft flight conflict problem and incorporating the actual control operational regulations, the problem is modeled as a Markov Decision Process (MDP). The Independent Deep Q Network (IDQN) algorithm combined with the Deep Q Network (DQN) algorithm and an independent learning framework allows multiple conflicting aircraft to interact with the environment at the same time and obtain feedback, and the optimal resolution strategy can be obtained through continuous learning. At the same time, a flexible ’downward-compatible’ conflict resolution framework is proposed to support the dynamic variation in the number of conflicting aircraft. Finally, the validity and applicability of the model are verified by data, experiments and related indicators. The rest of this paper proceeds as follows: Section 2 introduces the research basis, including related works on the study of flight conflict resolution, the definition of multi-aircraft flight conflict and control regulations. Section 3 analyzes multi-aircraft flight conflicts, and the problem is modeled and described. Section 4 introduces the principle of the IDQN algorithm, the design of the resolution mechanism, the method of solving models and the result of model training. Section 5 is the result analysis of the conflict test data, including the analysis of related indicators such as successful conflict resolution rate, conflict resolution trends, the delay time after resolution, calculation time and distribution of successful conflict resolution rate. Section 6 presents the conclusions and research prospects.

通过分析多机飞行冲突问题的特点，并结合实际控制运行规定，将该问题建模为马尔可夫决策过程 (MDP)。结合深度 Q 网络 (DQN) 算法的独立深度 Q 网络 (IDQN) 算法和独立学习框架，使得多架冲突飞机能够同时与环境交互并获得反馈，通过持续学习可以获得最优解决策略。同时，提出了一个灵活的“向下兼容”的冲突解决框架，以支持冲突飞机数量的动态变化。最后，通过数据、实验和相关指标验证了模型的有效性和适用性。本文其余部分安排如下: 第 2 节介绍研究基础，包括关于飞行冲突解决的研究、多

* Corresponding author.
”* 通讯作者。”
E-mail address: xwping@nuaa.edu.cn (W. XU).
” 电子邮箱地址:xwping@nuaa.edu.cn (W. XU)。”
Peer review under responsibility of Editorial Committee of CJA.
” 同行评审由《中国航空学会学报》编辑委员会负责。”

机飞行冲突的定义和控制规定。第 3 节分析多机飞行冲突，并对问题进行建模和描述。第 4 节介绍 IDQN 算法的原理、解决机制的设计、模型求解方法以及模型训练结果。第 5 节是冲突测试数据的分析结果，包括成功冲突解决率、冲突解决趋势、解决后的延迟时间、计算时间以及成功冲突解决率的分布等相关指标的分析。第 6 节呈现结论和研究展望。

2. Research basis

2. 研究基础

Research on flight conflict management has been ongoing since its existence. Many scholars have studied flight conflict detection and resolution methods. At present, the methods of flight conflict detection have been developed comprehensively, but the methods of flight conflict resolution still attract much attention. This paper summarizes works related to the existing flight conflict resolution methods, puts forward research objectives, and gives a definition of multi-aircraft flight conflict.

自飞行冲突管理存在以来，对其的研究一直在进行。许多学者研究了飞行冲突检测和解决方法。目前，飞行冲突检测方法已经全面发展，但飞行冲突解决方法仍然受到广泛关注。本文总结了与现有飞行冲突解决方法相关的研究工作，提出了研究目标，并给出了多机飞行冲突的定义。

2.1. Related works

2.1. 相关研究工作

There are three common methods of flight conflict resolution: the swarm intelligence optimization algorithm, the optimal control theory and the hybrid system model. In recent years, a fourth method has been derived, which uses Reinforcement Learning (RL) methods to solve flight conflicts. The swarm intelligence optimization algorithm is often used in conflict resolution. The conflict process is usually divided into a series of discrete segments according to equal lengths of time or distance and then optimized by the algorithm. In 1996, Durand et al. [3] used a Genetic Algorithm (GA) to solve the flight conflict problem for the first time by considering the velocity error of aircraft and classifying multi-aircraft conflicts. On this basis, Durand [4] later proposed a conflict resolution method based on a neural network which learned through GA, and this method can effectively reduce the calculation time. However, it is difficult to extend to 3D conflict scenarios, and as the scale of neural networks expands, learning will be more difficult. To solve the problem with the shortest fuel consumption and the least resolution time, Stephane and Conway [5] used GA for conflict resolution considering the constraint conditions. However, the algorithm ran slowly and was not suitable for real-time solving. Later, Ma et al. [6] used GA to solve the conflict between three aircraft under the condition of free flight, and also considered the problem of aircraft flying according to air route, but the model can't be used when the number of aircraft increases. Hereafter, Guan et al. [7] proposed a global optimization method combining Memetic Algorithms (MA) and GA to resolve conflicts between multiple aircraft and reduce calculation time. In addition to GA, Particle Swarm Optimization (PSO) algorithm is also used to resolve flight conflicts, and these swarm intelligence optimization algorithms have similar principles in implementation. In recent studies, researchers have combined these algorithms with multi-agent concepts, such as Emami and Derakhshan. [8] used PSO algorithm to resolve conflicts and verified that the resolution strategy can reduce the delay time and fuel consumption. Zhou et al. [9] applied a distributed Multi-Agent System (MAS) to the flight conflict resolution problem and integrated a distributed algorithm and adaptive GA to solve the problem so that the multi-aircraft conflict problem could be solved from a global perspective. However, as the number of aircraft in the scenario increases, the resolution time increases significantly. Liu et al. [10] proposed an improved mechanism on the basis of PSO to plan time coordinated and conflict avoidance paths for multiple Unmanned Aerial Vehicles (UAVs), taking ETA, separation maintenance and performance constraints into consideration.

有三种常见的飞行冲突解决方法: 群体智能优化算法、最优控制理论和混合系统模型。近年来，衍生出第四种方法，即使用强化学习 (RL) 方法解决飞行冲突。群体智能优化算法常用于冲突解决。冲突过程通常根据相等的时间或距离段划分为一系列离散的段落，然后通过算法进行优化。1996 年，Durand 等人 [3] 首次使用遗传算法 (GA) 通过考虑飞机的速度误差并对多机冲突进行分类来解决飞行冲突问题。在此基础上，Durand [4] 后续提出了一种基于神经网络的冲突解决方法，该方法通过 GA 进行学习，可以有效地减少计算时间。然而，这种方法难以扩展到三维冲突场景，并且随着神经网络规模的扩大，学习将变得更加困难。为了解决燃油消耗最少和解决时间最短的问题，Stephane 和 Conway [5] 使用了考虑约束条件的 GA 进行冲突解决。但是，算法运行缓慢，不适合实时求解。后来，Ma 等人 [6] 使用 GA 解决了自由飞行条件下三架飞机之间的冲突，并也考虑了飞机按照航线飞行的问题，但该模型在飞机数量增加时无法使用。此

后，Guan 等人 [7] 提出了一种结合遗传算法 (MA) 和 GA 的全局优化方法，以解决多架飞机之间的冲突并减少计算时间。除了 GA，粒子群优化 (PSO) 算法也用于解决飞行冲突，这些群体智能优化算法在实施原理上相似。在最近的研究中，研究人员将这些算法与多代理概念相结合，例如 Emami 和 Derakhshan。[8] 使用 PSO 算法解决冲突，并验证了解决策略可以减少延迟时间和燃油消耗。Zhou 等人 [9] 将分布式多代理系统 (MAS) 应用于飞行冲突解决问题，并整合了分布式算法和自适应 GA 来解决问题，从而可以从全局角度解决多机冲突问题。然而，随着场景中飞机数量的增加，解决时间显著增加。Liu 等人 [10] 在 PSO 的基础上提出了改进机制，为多架无人机 (UAVs) 规划时间协调和冲突避免路径，考虑了预计到达时间 (ETA)、间隔保持和性能约束。

The purpose of applying optimal control theory is mainly to plan the optimal path of aircraft. In 1998, Bicchi et al. [11] studied how to apply a robotic optimal control model to flight conflict resolution. Many constraints are set in the algorithm, the calculation is complicated, and the applicability is poor. Later, Menon et al. [12] used a discrete waypoint model and ellipsoid conflict model to study flight conflict resolution, and the problem of optimizing the flight path is transformed into that of optimizing the parameters. Ghosh and Tomlin. [13] combined the dynamic constraint problem with game theory and used dynamic system analysis to solve the flight conflict problem. All the above attempts aimed at solving the conflict problem of flights in 2D scenarios. Narkawicz et al. [14] proposed a 3D geometric algorithm extended from the 2D optimal set algorithm on the basis of optimization theory to calculate the set of all points whose trajectories are tangent to the aircraft protection area and then reverse this process to obtain the resolution strategy. Hu et al. [15] used the Riemann manifold method to solve the conflict problem, and formation flight was optimized. Later, Liu et al. [16] obtained a controllable Markov chain by discretizing the flight conflict process, and the solution of the stochastic optimal control problem was obtained by finding the optimal control discipline. Han et al. [17] constructed the optimal conflict resolution model of aircraft on the same flight level by using optimal control theory under the condition of flying along a fixed route, and the operation instructions of the aircraft in each substage were obtained according to the optimal conflict-free track. Tang et al. [18] proposed a dynamic optimal resolution strategy based on rolling time-domain optimization on the basis of a static single optimal resolution strategy considering the heading and low speed adjustment, including uncertain factors such as velocity disturbance that may exist during aircraft flight. Li et al. [19] proposed a distributed conflict resolution method based on game theory and Internet of Things (IoT) technology for UAVs. Using game theory, it is possible to theoretically obtain a balanced optimal conflict-free trajectory suitable for multiple UAVs, and this method can also be applied to civil aircraft flight [20] to obtain an optimal and balanced resolution strategy.

应用最优控制理论的主要目的是规划飞机的最优路径。1998 年，Bicchi 等人 [11] 研究了如何将机器人最优控制模型应用于飞行冲突解决。算法中设置了多种约束，计算复杂，适用性差。后来，Menon 等人 [12] 使用离散航点模型和椭球冲突模型研究飞行冲突解决，将优化飞行路径的问题转化为优化参数的问题。Ghosh 和 Tomlin [13] 将动态约束问题与博弈论相结合，并使用动态系统分析解决飞行冲突问题。以上所有尝试都是为了解决二维场景中飞行的冲突问题。Narkawicz 等人 [14] 在优化理论的基础上，提出了一种从二维最优集合算法扩展到三维的几何算法，计算所有轨迹与飞机保护区相切的点的集合，然后逆转此过程以获得解决策略。Hu 等人 [15] 使用黎曼流形方法解决冲突问题，并对编队飞行进行了优化。后来，Liu 等人 [16] 通过离散化飞行冲突过程获得可控马尔可夫链，并通过寻找最优控制策略解决了随机最优控制问题。Han 等人 [17] 在飞机沿固定航线飞行条件下，使用最优控制理论构建了同一飞行高度上飞机的最优冲突解决模型，并根据最优无冲突轨迹获得了飞机在每个子阶段的操作指令。Tang 等人 [18] 在考虑航向和低速调整的静态单一最优解决策略的基础上，提出了一种基于滚动时间域优化的动态最优解决策略，包括飞行中可能存在的不确定因素，如速度扰动。Li 等人 [19] 提出了一种基于博弈论和物联网 (IoT) 技术的无人机分布式冲突解决方法。使用博弈论，理论上可以获得适合多无人机的平衡最优无冲突轨迹，并且这种方法也可以应用于民用飞机飞行 [20] 以获得最优且平衡的解决策略。

Conflict resolution based on a hybrid system model decentralizes the control function to each aircraft for conflict resolution. A hybrid system is a dynamic system, which is a special case of optimal control theory. Pappas et al. [21] performed conflict resolution by switching flight states, and the motion parameters of aircraft at different flight phases were calculated to control the motion states to ensure that the flight of aircraft was within the envelope of flight safety restrictions. This is regarded as a conflict resolution strategy. Tang et al. [22] used a hybrid system model to detect and solve flight conflicts in real time, and the changes in the distance or speed of the aircraft over time are shown in the form of a curve, and based on which, conflict detection and resolution were carried out. In 2017, Soler et al. [23] transformed the conflict-free track planning problem into the problem of hybrid optimal control. The discrete mode of the hybrid system was used for conflict resolution, and an actual seven-aircraft conflict scenario was simulated to find the conflict-free track with the lowest fuel consumption.

基于混合系统模型的冲突解决将控制功能去中心化，分配给每架飞机以解决冲突。混合系统是一种动态系统，它是最优控制理论的一种特殊情况。Pappas 等人 [21] 通过切换飞行状态进行冲突解决，并计算了不同飞行阶段飞机的运动参数，以控制运动状态，确保飞机的飞行在飞行安全限制的范围内。这被认为是一种冲突解决策略。Tang 等人 [22] 使用混合系统模型实时检测和解决飞行冲突，并展示了随着时间的推移飞机距离或速度的变化曲线，基于这些曲线，进行了冲突检测和解决。在 2017 年，Soler 等人 [23] 将无

冲突航迹规划问题转化为混合最优控制问题。使用混合系统的离散模式进行冲突解决，并对一个实际七架飞机的冲突场景进行了模拟，以找到燃油消耗最低的无冲突航迹。

The method of RL takes an aircraft as an agent with functions such as consciously receiving instructions, executing instructions and changing states. The aircraft learns by constantly interacting with the environment of a conflict scenario, and it obtains the optimal resolution strategy through the guidance of the reward function. In flight conflict resolution, the data dimension that needs to be recorded is large, so the powerful recording capability of neural networks is needed, that is, the Deep Reinforcement Learning (DRL) method. The DRL method began to be developed in approximately 2015, and it is mainly applied in robotics, strategy games and other fields. In recent years, researchers have tried to use DRL method to solve the problem of flight conflict resolution. In 2018, Wang et al. [24] proposed K -Control Actor-Critic (KCAC) algorithm to choose the random position for aircraft to avoid conflict. Considering the turning radius of the aircraft, they obtained the optimal or sub-optimal conflict-free flight trajectory by limiting the number of control times and heading changes, which proved the computational efficiency of DRL algorithm. In 2019, Pham et al. [25] used the Deep Deterministic Policy Gradient (DDPG) algorithm of DRL to establish an automatic flight conflict resolution model, and the test showed that the method under conditions of uncertainty can effectively solve conflicts between aircraft. Under different uncertainties, the precision of the model is approximately 87% , showing that the DRL method of solving flight conflicts is a promising approach, but it is only used for two-aircraft flight conflicts and does not extend to multi-aircraft flight conflict scenarios. Then, Pham’ s team [26] incorporated the experience of an actual controller, and the model trained through DRL could learn from the controller’s experience. The experimental results showed that the matching degree of the resolution scheme obtained by the training model and the controller’s manual solution was 65% . In 2019, Wang et al. [27] applied DRL to flight conflict detection and resolution strategies and developed a suitable training and learning environment for aircraft conflict detection and resolution by changing 2D continuous speed actions and headings to select a strategy. The study showed the possibility of applying DRL to the flight conflict resolution problem, which has obvious advantages in terms of calculation efficiency, but the research used only 2D continuous action adjustments without considering changes in altitude or other control instructions.

强化学习 (RL) 方法将飞机作为具有自觉接收指令、执行指令和改变状态等功能的智能体。飞机通过与冲突场景环境不断交互进行学习，并在奖励函数的指导下获得最优解决策略。在飞行冲突解决中，需要记录的数据维度很大，因此需要神经网络强大的记录能力，即深度强化学习 (DRL) 方法。DRL 方法大约从 2015 年开始发展，主要应用于机器人、策略游戏等领域。近年来，研究人员尝试使用 DRL 方法解决飞行冲突问题。2018 年，王等人 [24] 提出了 K -控制演员-评论家 (KCAC) 算法，为飞机选择随机位置以避免冲突。考虑到飞机的转弯半径，他们通过限制控制次数和航向变化，获得了最优或次优的无冲突飞行轨迹，证明了 DRL 算法的计算效率。2019 年，Pham 等人 [25] 使用 DRL 的深度确定性策略梯度 (DDPG) 算法建立自动飞行冲突解决模型，测试表明该方法在不确定性条件下能有效解决飞机间的冲突。在不同不确定性条件下，模型的精度约为 87% ，表明 DRL 方法解决飞行冲突是一种有前景的方法，但它仅适用于双机飞行冲突，并未扩展到多机飞行冲突场景。随后，Pham 团队 [26] 引入了实际控制员的经验，通过 DRL 训练的模型能够学习控制员的经验。实验结果表明，训练模型得到的解决方案与控制员手动解决方案的匹配度为 65% 。2019 年，王等人 [27] 将 DRL 应用于飞行冲突检测与解决策略，通过改变二维连续速度动作和航向选择策略，为飞机冲突检测与解决开发了一套合适的训练与学习环境。研究表明，DRL 应用于飞行冲突解决问题是可能的，其在计算效率方面具有明显优势，但研究仅使用了二维连续动作调整，未考虑高度或其他控制指令的变化。

Based on the analysis above, there are many methods to solve conflicts. At the same time, all kinds of more complicated and practical conditions are gradually being taken into consideration, but this problem still has research value. To compare the features and limitations of various research methods more clearly, the methods mentioned above are displayed in Table 1, which contains the method category, whether the method is based on the assumption of free flight, the number of conflicting aircraft, the types of conflict scenarios, and whether the method is based on actual control regulations.

基于上述分析，有许多方法可以解决冲突。同时，各种各样的更复杂、更实际的条件正在逐渐被考虑进去，但这个问题仍然具有研究价值。为了更清楚地比较各种研究方法的特征和局限性，上述方法在表 1 中展示，其中包含方法类别、方法是否基于自由飞行的假设、冲突飞机的数量、冲突场景的类型以及方法是否基于实际控制规定。

The characteristics of the research methods in Table 1 are summarized as follows: (A) Most conflict resolution methods are based on the assumption of free flight. (B) The methods that can resolve a multi-aircraft conflict problem can be used to resolve two-aircraft conflicts, but the reverse situation is not necessarily feasible. (C) Most conflict resolution methods only consider 2D scenarios, that is, adopt speed or heading adjustments to resolve a conflict in the same plane. (D) Few conflict resolution methods consider the restrictions of control operational regulations. Although these studies can resolve the flight conflict problem to a certain extent, some methods have some shortcomings. For example, methods 1, 2, 3, 4 and 7 have low calculate efficiency. They may have been limited by the equipment at the time and the low efficiency of the GA in decoding. Methods 10, 12, 16 and 22 have the problem of scenario specialization, that is, these methods can only address smaller scenarios, and some can only resolve specific two-aircraft or three-aircraft flight conflicts.

表 1 中研究方法的特征概括如下:(A) 大多数冲突解决方法是基于自由飞行的假设。(B) 能够解决多飞机冲突问题的方法可以用来解决双飞机冲突，但反过来则不一定可行。(C) 大多数冲突解决方法仅考虑二维场景，即采用速度或航向调整来解决同一平面内的冲突。(D) 很少有冲突解决方法考虑控制操作规定的限制。尽管这些研究在一定程度上可以解决飞行冲突问题，但某些方法存在一些不足。例如，方法 1、2、3、4 和 7 的计算效率较低。它们可能受到了当时设备和遗传算法解码效率低下的限制。方法 10、12、16 和 22 存在场景专用化的问题，即这些方法只能处理较小的场景，有些只能解决特定的双飞机或三飞机飞行冲突。

Table 1 Features of flight conflict resolution methods.

表 1 飞行冲突解决方法特征。

Category	No.	Scholars	Methods	Free flight	Two	Multiple	2D scenario	3D scenario	Control regulations
Swarm intelligence optimization algorithm	1	Durand3	GA	×	◦	◦	0	×	0
	2	Durand4	Neural Network	×	0	◦	0	×	0
	3	Stephane5	GA	×	◦	×	◦	×	×
	4	Ma6	GA	0	◦	◦	×	0	×
	5	Guan7	MA, GA	×	0	◦	0	0	×
	6	Emami8	PSO	◦	◦	◦	×	◦	×
	7	Zhou9	Adaptive GA	×	◦	◦	×	0	◦
	8	Liu^{10}	Improved PSO	◦	0	◦	0	◦	×
Optimal control theory method	9	Bicchi1	Optimal Control Model	◦	0	×	0	×	×
	10	Menon ^{12}	Parametric Optimal Model	◦	0	◦	×	0	×
	11	Tomlin 13	Game Theory	o	0	◦	×	0	×
	12	Dowek1 4	3-D Geometric Algorithm	0	◦	×	×	0	×
	13	Hu^{15}	Riemann Manifold Method	◦	0	0	×	◦	×
	14	Liu ^{16}	Optimal Control Theory	◦	◦	◦	×	◦	×
	15	Han^{17}	Optimal Control Theory	×	◦	X	0	×	0
	16	Tang^{18}	Optimal Dynamic Mixing	×	◦	X	◦	×	◦
	17	Li^{19}	Game Theory, IoT	◦	◦	◦	◦	◦	×
Hybrid system model algorithm	18	Pappas ^{21}	Hybrid System Model	◦	◦	0	0	×	×
	19	Tang2	Hybrid System Model	×	0	×	o	×	◦
	20	Soler ^{23}	Hybrid Optimal Control	0	◦	0	◦	×	×
RL methods	21	Li^{24}	KCAC	◦	0	0	o	×	×
	22	Pham^{25}	DDPG	0	◦	×	◦	×	×
	23	Pham2 6	DDPG	◦	◦	×	◦	×	0
	24	Wang^{27}	DRL	0	0	×	0	×	×

类别	编号	学者	方法	自由飞行	二	多个	二维场景	三维场景	控制规定
群智能优化算法	1	Durand3	遗传算法 (GA)	×	◦	◦	0	×	0
	2	Durand4	神经网络	×	0	◦	0	×	0
	3	Stephane5	遗传算法 (GA)	×	◦	×	◦	×	×
	4	Ma6	遗传算法 (GA)	0	◦	◦	×	0	×
	5	关 7	MA, GA	×	0	◦	0	0	×
	6	Emami8	PSO	◦	◦	◦	×	◦	×
	7	周九	自适应 GA	×	◦	◦	×	0	◦
	8	Liu^{10}	改进 PSO	◦	0	◦	0	◦	×
最优控制理论方法	9	Bicchi1	最优控制模型	◦	0	×	0	×	×
	10	Menon ^{12}	参数化最优模型	◦	0	◦	×	0	×
	11	Tomlin 13	博弈论	o	0	◦	×	0	×
	12	Dowek1 4	三维几何算法	0	◦	×	×	0	×
	13	Hu^{15}	黎曼流形方法	◦	0	0	×	◦	×
	14	Liu ^{16}	最优控制理论	◦	◦	◦	×	◦	×
	15	Han^{17}	最优控制理论	×	◦	X	0	×	0
	16	Tang^{18}	最优动态混合	×	◦	X	◦	×	◦
	17	Li^{19}	博弈论，物联网	◦	◦	◦	◦	◦	×
混合系统模型算法	18	Pappas ^{21}	混合系统模型	◦	◦	0	0	×	×
	19	Tang2	混合系统模型	×	0	×	o	×	◦
	20	Soler ^{23}	混合最优控制	0	◦	0	◦	×	×
强化学习方法	21	Li^{24}	KCAC	◦	0	0	o	×	×
	22	Pham^{25}	DDPG	0	◦	×	◦	×	×
	23	Pham2 6	DDPG	◦	◦	×	◦	×	0
	24	Wang^{27}	深度强化学习	0	0	×	0	×	×

Note: ◦ indicates that the conditions are met, and × indicates that the conditions are not met.

备注: ◦ 表示条件满足，× 表示条件不满足。

In actual operation, flight conflict resolution needs to meet the requirements of flying along the air route, solving the 3D conflict problem, and adjusting based on the control operational regulations. Among the methods above, only the method proposed by Zhou Jian (No.7) can meet these three requirements at the same time. This method improved the traditional GA, but with the increase in the number of aircraft, the algorithm programming needs to be modified accordingly, the programming will become complicated, the convergence speed will become slower, and the time needed to obtain the resolution strategy will increase. Therefore, for flight conflict resolution, on the basis of meeting three basic requirements, it is also necessary to achieve the research objectives of high efficiency, resolving multiple types of conflict scenarios and addressing dynamic changes in the number of conflicting aircraft. In this paper, combined with the basic requirements and research objectives, the IDQN algorithm of DRL is proposed to solve the multi-aircraft flight conflict problem, which can be greatly improved by offline training and online use. At the same time, an extensible framework is designed to support dynamic changes in the number of conflicting aircraft.

实际运行中，航班冲突解决需要满足沿航线飞行要求，解决三维冲突问题，并根据控制操作规程进行调整。在上述方法中，只有周健 (第 7 号) 提出的方法能够同时满足这三个要求。该方法改进了传统的遗传算法 (GA)，但随着飞机数量的增加，算法编程需要相应修改，编程将变得复杂，收敛速度将变慢，获取解决策略所需的时间将增加。因此，对于航班冲突解决，在满足三个基本要求的基础上，还有必要实现高效、解决多种冲突场景以及应对冲突飞机数量动态变化的研究目标。在本文中，结合基本要求和研究目标，提出了基于深度强化学习 (DRL) 的 IDQN 算法来解决多飞机航班冲突问题，通过离线训练和在线使用可以大大提高。同时，设计了一个可扩展框架，以支持冲突飞机数量的动态变化。

2.2. Multi-aircraft flight conflict

2.2. 多飞机航班冲突

A flight conflict is a situation in which the distance between aircraft is less than the specified minimum radar separation during flight. During en-route flight, the horizontal separation should not be less than 10 km , and the vertical separation should be in accordance with the flight level. If the distance between two aircraft is less than the minimum radar separation in both horizontal and vertical directions, it is considered a flight conflict. The moment when the distance between two aircraft first reaches the minimum separation and the aircraft continue to approach each other is regarded as the moment of flight conflict.

航班冲突是指飞行中飞机间的距离小于规定的最小雷达间隔的情况。在航路飞行中，水平间隔不应小于 10 km ，垂直间隔应符合飞行高度层。如果两架飞机在水平和垂直方向上的距离都小于最小雷达间隔，则被认为是航班冲突。两架飞机之间的距离首次达到最小间隔且飞机继续接近对方的时刻被认为是航班冲突的时刻。

There is no clear definition of multi-aircraft flight conflicts in the official documents of organizations such as ICAO. Conceptually, a multi-aircraft conflict is a situation in which an aircraft conflicts with two or more other aircraft in a certain time and space. However, this definition cannot cover all kinds of situations and is not precise enough, so it is necessary to supplement and clarify the definition of multi-aircraft flight conflict.

官方文件如国际民航组织 (ICAO) 中并没有对多机飞行冲突的明确定义。从概念上讲，多机冲突是指一架飞机在特定时间和空间内与两架或更多其他飞机发生冲突的情况。然而，这一定义无法涵盖所有情况，且精确性不足，因此有必要补充和明确多机飞行冲突的定义。

A multi-aircraft flight conflict is a situation in which multiple aircraft have flight conflicts within a certain space and a certain period of time. It is also necessary to ensure that at least one aircraft can be connected to other aircraft through their conflict relationships. To illustrate the definition of multi-aircraft flight conflicts more visually, several scenarios are shown in Fig. 1. There are three constraints in the definition, and a situation cannot be defined as a multi-aircraft flight conflict if it does not satisfy all three constraints at once. In the later training and testing parts, samples of conflict scenarios used are satisfied with all constraints as shown in Fig. 1(d).

多机飞行冲突是指多架飞机在特定空间和特定时间内存在飞行冲突的情况。同时，还需要确保至少有一架飞机能够通过它们的冲突关系与其他飞机连接。为了更直观地说明多机飞行冲突的定义，图 1 中展示了几个场景。该定义中有三个约束条件，如果一种情况不能同时满足这三个约束条件，则不能将其定义为多机飞行冲突。在后续的训练和测试部分，使用的冲突场景样本满足图 1(d) 所示的所有约束条件。

Fig. 1(a): The two conflict points meet the time constraints, and the three aircraft can be connected at the same time, but they exceed the space limit, so this is considered as a two-aircraft conflict rather than a multi-aircraft flight conflict.

图 1(a): 两个冲突点满足时间约束，三架飞机可以同时连接，但它们超出了空间限制，因此这被视作双机冲突，而非多机飞行冲突。

Fig. 1(b): The two conflict points meet the space limit, and three aircraft can be connected at the same time, but they exceed the time limit, so this is considered as a two-aircraft conflict rather than a multi-aircraft flight conflict.

图 1(b): 两个冲突点满足空间限制，三架飞机可以同时连接，但它们超出了时间限制，因此这被视作双机冲突，而非多机飞行冲突。

Fig. 1(c): The two conflict points meet the constraints of space and time, but they are isolated from each other and cannot be connected. Therefore, this is regarded as two two-aircraft conflicts instead of one multi-aircraft flight conflict.

图 1(c): 两个冲突点满足时间和空间的约束条件，但它们相互孤立，无法连接。因此，这被视为两个双机冲突，而非一个多机飞行冲突。

Fig. 1(d): The space and time constraints are satisfied, and the two independent conflicts can be connected through aircraft E, so this can be defined as a multi-aircraft flight conflict.

图 1(d): 空间和时间约束得到满足，两个独立的冲突可以通过飞机 E 连接，因此这可以定义为一个多机飞行冲突。

Therefore, only when all three constraints are met can a multi-aircraft flight conflict be defined. A multi-aircraft flight conflict can be considered as a combination of multiple two-aircraft conflicts whose intervals are defined in accordance with the radar control interval above. The limited time period should not be set too long. If the time is too long, it will be more efficient to resolve the problem of the multi-aircraft flight conflict as multiple two-aircraft conflicts. Through experimental research, it is found that setting Time = 4 min is an appropriate value.

因此，只有当所有三个约束都满足时，才能定义为一个多机飞行冲突。多机飞行冲突可以被视为多个双机冲突的组合，其间隔按照上述雷达控制间隔来定义。有限的时间段不应设置得太长。如果时间过长，将多个双机冲突作为多个两机冲突来解决问题会更加高效。通过实验研究，发现设置时间 = 4 min 是一个合适的值。

2.3.Air traffic control regulations

2.3. 空中交通管制规定

After the controller identifies the flight conflict, the controller continuously monitors the status of all aircraft in the sector in charge through the airspace dynamic information provided by the Air Traffic Control (ATC) automation system and coordinates the flight conflicts among aircraft. According to ATC regulations, controllers' experience and dynamic airspace environment, controllers usually adopt three kinds of resolution methods: altitude adjustments, speed adjustments and heading adjustments.

控制员识别出飞行冲突后，通过空中交通管制 (ATC) 自动化系统提供的空域动态信息，持续监控负责扇区内所有飞机的状态，并在飞机之间协调飞行冲突。根据 ATC 规定、控制员的经验和动态空域环境，控制员通常采用三种解决方法: 高度调整、速度调整和航向调整。

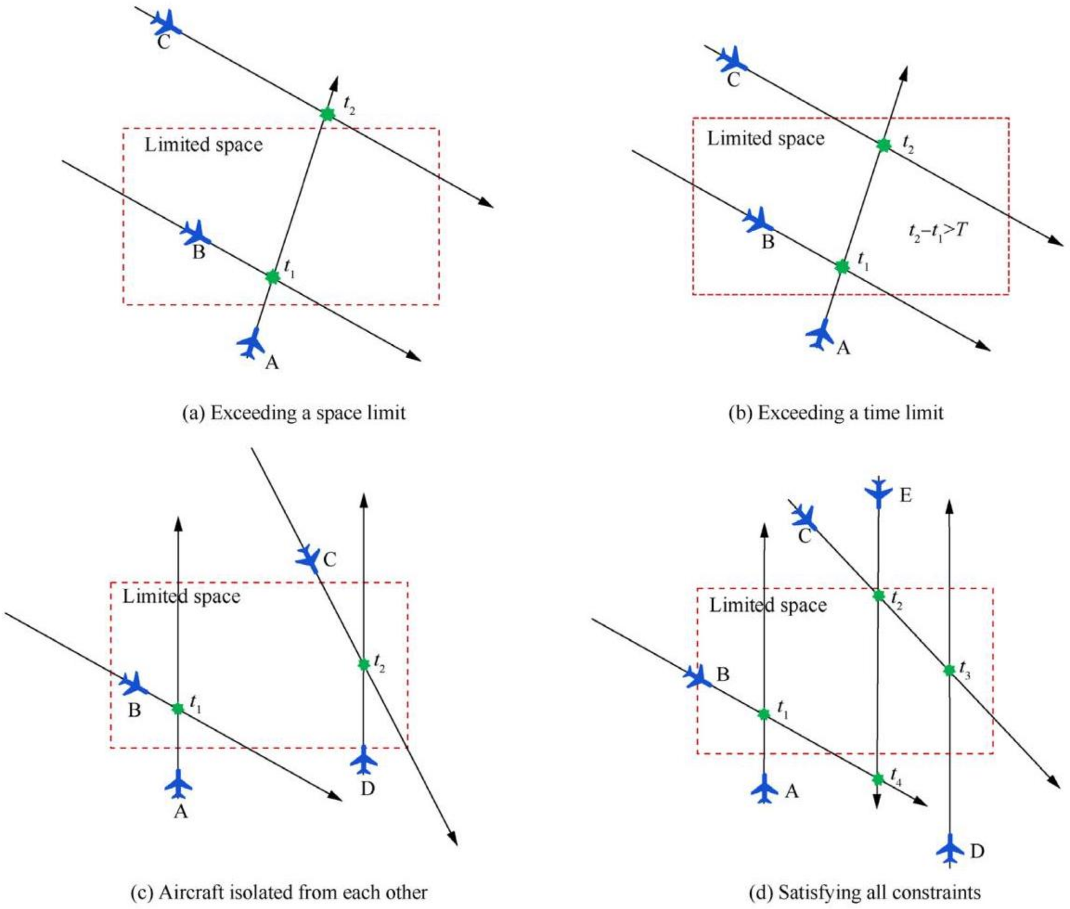


Fig. 1 Schematic diagram of multi-aircraft flight conflicts.

图 1 多机飞行冲突示意图。

(1) Altitude adjustments. This is the most frequently used and most effective adjustment. The aircraft en-route fly in accordance with the flight level provided. China is currently using the latest flight level configuration standard, which was implemented at 0:00 on November 22, 2007. At the same time, the RVSM is implemented in the range of 8900 m – 12500 m . The altitude ranges from 900 m to 12500 m , and every 300 m is a different flight level. Above 12500 m , there is a new flight level every 600 m . When the controller uses altitude adjustments, it is necessary to abide by the flight level configuration standard.

(1) 高度调整。这是最常使用且最有效的方法。途中飞机按照提供的飞行高度层飞行。中国目前使用最新的飞行高度层配置标准，该标准于 2007 年 11 月 22 日 0:00 开始实施。同时，在 8900 m – 12500 m 范围内实施 RVSM。高度从 900 m 到 12500 m，每个 300 m 是不同的飞行高度层。在 12500 m 以上，每 600 m 就有一个新的飞行高度层。当控制员使用高度调整时，必须遵守飞行高度层配置标准。

(2) Speed adjustments. This is a common resolution adjustment. At the cruising level, generally, aircraft fly according to the economical cruising speed stipulated by the airline to which it belongs. However, in certain circumstances, it is necessary to intervene in the aircraft speed, such as in the case of an over-waypoint speed limit or a conflict on the air route. When using speed adjustments, the controller should follow the corresponding regulations: at levels at or above 7600 m , speed adjustments should be expressed in multiples of 0.01 Mach. At levels below 7600 m , speed adjustments should be expressed in multiples of 10kt based on the Indicated Air Speed

(2) 速度调整。这是一种常见的解析调整。在巡航高度，通常，飞机按照所属航空公司规定的经济巡航速度飞行。然而，在某些情况下，需要干预飞机速度，例如在航路点速度限制或航路冲突的情况下。使用速度调整时，管制员应遵循相应的规定: 在 7600 m 及以上高度，速度调整应以 0.01 马赫的倍数表示。在 7600 m 以下高度，速度调整应以 10kt 为基础的指示空速的倍数表示。(IAS).

(3) Heading adjustments. The heading refers to the angle measured clockwise from the north end of the meridian to the extension line ahead of the aircraft's longitudinal axis. The specific means of making heading adjustments include radar guidance and offset. The method of radar guidance requires a large airspace range, and it commonly uses the 'dog-leg' maneuver and direct flight to the next waypoint, which is usually used in the terminal area or to deal with severe weather such as thunderstorms during an air route flight. The offset method is widely used in air route flights. The left or right offset method is used to offset a distance to the left or right parallel to the centerline

of the air route to widen the lateral space between aircraft. The offset is usually adjusted in nautical miles (nm), such as 6 nm .

(3) 航向调整。航向是指从子午线北端顺时针测量到飞机纵轴延长线的角度。进行航向调整的具体方法包括雷达引导和偏置。雷达引导方法需要较大的空域范围，通常使用“折返”机动和直接飞往下一个航路点，这通常用于终端区域或在航路飞行中处理严重天气，如雷暴。偏置方法在航路飞行中广泛使用。使用左或右偏置方法，将飞机偏置到航路中心线的左侧或右侧一定距离，以增加飞机之间的横向空间。偏置通常以海里 (nm) 为单位调整，例如 6 nm 。

3. Modeling of multi-aircraft flight conflict

3. 多机飞行冲突建模

From the analysis above, it can be seen that the key part of flight conflict resolution is to build the problem model and then use relevant methods to solve and calculate the model to obtain the theoretically optimal conflict resolution strategy. Therefore, it is necessary to analyze the problem of multi-aircraft flight conflict and select an appropriate method to model the problem.

从上述分析可以看出，解决飞行冲突的关键部分是构建问题模型，然后使用相关方法求解和计算模型，以获得理论上最优的冲突解决策略。因此，有必要分析多机飞行冲突问题，并选择适当的方法来建模问题。

3.1. Problem analysis

3.1. 问题分析

When determining the resolution strategy, the selection of actions for multiple aircraft needs to be considered. An aircraft has autonomous functions, such as receiving instructions and changing the flight status. It can be abstracted as an agent, so an aircraft agent is established, whose internal structure is shown in Fig. 2. The agent has the functions of generating an aircraft trajectory according to the flight plan and database files, receiving control instructions and status query signals, transmitting status information and so on. In a multi-aircraft flight conflict scenario, multiple aircraft agents are involved, and these aircraft agents are related to each other. Therefore, the multi-aircraft conflict scenario can be regarded as a MAS.

确定解决策略时，需要考虑多架飞机的行为选择。飞机具有自主功能，例如接收指令和改变飞行状态。它可以被抽象为一个代理，因此建立一个飞机代理，其内部结构如图 2 所示。该代理具有根据飞行计划和数据库文件生成飞机轨迹的功能，接收控制指令和状态查询信号，传输状态信息等。在多架飞机飞行冲突场景中，涉及多个飞机代理，这些飞机代理相互关联。因此，多架飞机冲突场景可以被视为一个多代理系统 (MAS)。

The typical modeling method for a MAS is to model the problem as an MDP or stochastic game process. Which modeling method should be adopted depends on the characteristics of the scenario. From actual control resolution regulations, when potential conflicts are detected, the controllers send resolution instructions according to the current state of aircraft and the future airspace trends, without considering the earlier states before the aircraft executed the instructions. Therefore, after the aircraft executes the conflict resolution instruction, the next environment feedback is only related to the current state and action, which is in line with the precondition of MDP, that is, Markov property. The dynamic process of MDP is shown in Fig. 3. The agent’s initial state is s_0 , an action a_0 is selected for execution, and the agent is randomly transferred to the next state s_1 according to probability. Then perform another action a_1 transfer to the next state s_2 repeat the process and continue.

多代理系统 (MAS) 的典型建模方法是将问题建模为马尔可夫决策过程 (MDP) 或随机博弈过程。应该采用哪种建模方法取决于场景的特点。从实际控制解决规定来看，当检测到潜在冲突时，控制器会根据飞机的当前状态和未来空域趋势发送解决指令，而不考虑飞机执行指令前的早期状态。因此，在飞机执行冲突解决指令后，下一个环境反馈仅与当前状态和动作相关，这符合 MDP 的前提条件，即马尔可夫性。MDP 的动态过程如图 3 所示。代理的初始状态为 s_0 ，选择一个动作 a_0 执行，代理根据概率随机转移到下一个状态 s_1 。然后执行另一个动作 a_1 转移到下一个状态 s_2 ，重复此过程并继续。

A reward r_{t+1} is given for the transition from state s_t to state s_{t+1} throughout the process. As the state changes, the value of the cumulative reward given can be calculated using the Fig. 4, which shows the MDP process diagram with rewards.

在整个过程中，从状态 s_t 转移到状态 s_{t+1} 会给予一个奖励 r_{t+1} 。随着状态的改变，可以使用图 4 所示的带有奖励的 MDP 过程图来计算给出的累积奖励值。

According to the above introduction, Temizer et al. [28] described that when there is no information uncertainty, the optimal control problem of stochastic systems can be described as MDP. In this paper, since the research is to provide decision-making support to the controller, all aircraft information is assumed to be available through tools like ATC automation system, so there is no question of information uncertainty and communications between aircraft can be ignored, so the multi-aircraft flight conflict resolution problem can be modeled as an MDP.

根据上述介绍，Temizer 等人 [28] 描述了在没有信息不确定性的情况下，随机系统的最优控制问题可以描述为马尔可夫决策过程 (MDP)。在本文中，由于研究旨在为控制器提供决策支持，因此假设所有飞机信息都可以通过类似于空中交通管制自动化系统 (ATC) 的工具获得，所以不存在信息不确定性的问题，飞机之间的通信可以忽略不计，因此多飞机飞行冲突解决问题可以被建模为一个 MDP。

3.2. Model description

3.2. 模型描述

Through the above analysis, the multi-aircraft flight conflict scenario is modeled as a MAS, in which each conflicting aircraft is regarded as an agent that has the ability to receive the controller's instructions, execute the resolution instructions, and fly according to the track prediction model. These aircraft agents are independent of each other and do not have any other corresponding behaviors, such as communication. An independent learning framework is adopted for training and learning. Therefore, each aircraft in the conflict environment has an MDP model, which is specifically expressed in the following four parts: $\mathrm{MDP} = (S, A, P_{\mathrm{sa}}, R)$.

通过上述分析，将多飞机飞行冲突场景建模为一个多代理系统 (MAS)，其中每个冲突飞机被视为一个能够接收控制器指令、执行解决指令并按照轨迹预测模型飞行的代理。这些飞机代理相互独立，没有其他相应的行为，例如通信。采用一个独立的学习框架进行训练和学习。因此，冲突环境中的每架飞机都有一个 MDP 模型，具体表示为以下四个部分: $\mathrm{MDP} = (S, A, P_{\mathrm{sa}}, R)$ 。

(1) State space

(1) 状态空间

The state space is expressed as $S = \{S_1, S_2, ..., S_n\}$, and $S_i\ (i \in n)$ represents the state of one small grid. The state space is the collection of the position and state information of all aircraft in the control sector at a certain time. The aircraft performs flight trajectory prediction 5 min in advance. The point where the aircraft is 10 km apart for the first time and continues to approach is considered the Minimum Safe Interval Point (MSIP), as shown in Fig. 5(a). Aircraft A and aircraft B reach the MSIP at time t_1 . If the aircraft continues to fly and maintains the minimum safety interval to aircraft C at time t_2 , the positions of aircraft A and B at time t_2 and aircraft C at time t_2 constitute a polygon, with the polygon’s center of gravity as the center of the state space. If there are other aircraft causing multi-aircraft flight conflicts, the expansion continues in accordance with the above method.

状态空间表示为 $S = \{S_1, S_2, ..., S_n\}$ ，其中 $S_i\ (i \in n)$ 表示一个小格子的状态。状态空间是某一时刻控制区域内所有飞机的位置和状态信息的集合。飞机提前进行飞行轨迹预测 5 min 。飞机首次相距 10 km 并继续接近的点被认为是图 5(a) 所示的最低安全间隔点 (MSIP)。飞机 A 和飞机 B 在时间 t_1 达到 MSIP。如果飞机继续飞行并在时间 t_2 保持与飞机 C 的最小安全间隔，那么时间 t_2 的飞机 A 和 B 的位置以及时间 t_2 的飞机 C 的位置构成一个多边形，该多边形的重心作为状态空间的中心。如果有其他飞机导致多机飞行冲突，则按照上述方法继续扩展。

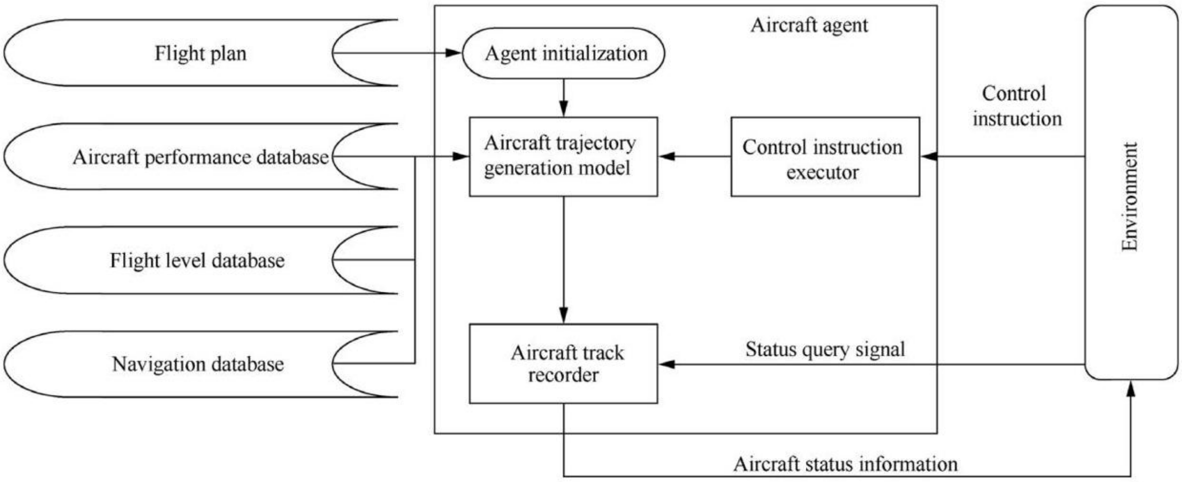


Fig. 2 Internal structure of an aircraft agent.
图 2 飞机代理的内部结构。

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \cdots$$

Fig. 3 MDP dynamic process diagram.
图 3 MDP 动态过程图。

The center of the state space extends 100 km transversely, 100 km longitudinally, and 3 km vertically around the center of the state space, forming a 200 km × 200 km × 6 km cuboid, as shown in Fig. 5(b). To facilitate the recording of state information, the cuboid is discretized into 8000 small grids with a size of $10\ \mathrm{km} \times 10\ \mathrm{km} \times 0.3\ \mathrm{km}$. The state in each small grid contains 8 pieces of information: the aircraft call sign, type, longitude, latitude, altitude, heading, vertical velocity and horizontal velocity. The whole space set is set is $S = \{S_1, S_2, \ldots, S_{8000}\}$ and each S_i contains $S_i = \{\mathrm{acft}_{\mathrm{id}}, \mathrm{acft}_{\mathrm{type}}, \mathrm{acft}_{\mathrm{ing}}, \mathrm{acft}_{\mathrm{lat}}, \mathrm{acft}_{\mathrm{heading}}, \mathrm{vvel}, \mathrm{hvel}\}$. If there are no aircraft in the grid, all the values are 0 .

状态空间的中心向中心横向扩展 100 km， 纵向扩展 100 km， 垂直扩展 3 km， 形成如图 5(b) 所示的 200 km × 200 km × 6 km 立方体。为了方便记录状态信息，将立方体离散成 8000 个小格子，每个小格子的尺寸为 10 km × 10 km × 0.3 km 。每个小格子中的状态包含 8 条信息: 飞机呼号、类型、经度、纬度、高度、航向、垂直速度和水平速度。整个空间集设置为 $S = \{S_1, S_2, \ldots, S_{8000}\}$， 每个 S_i 包含 $S_i = \{\mathrm{acft}_{\mathrm{id}}, \mathrm{acft}_{\mathrm{type}}, \mathrm{acft}_{\mathrm{ing}}, \mathrm{acft}_{\mathrm{lat}}, \mathrm{acft}_{\mathrm{heading}}, \mathrm{vvel}, \mathrm{hvel}\}$ 。如果格子中没有飞机，则所有值均为 0。

(2) Action space

(2) 动作空间

The action space is expressed as A . The action space consists of the conflict resolution adjustments that the controller can make at each moment. In this paper, speed adjustments, altitude adjustments and heading adjustments are mainly used.

动作空间表示为 A 。动作空间由控制器在每一时刻可以进行的冲突解决调整组成。在本文中，主要使用了速度调整、高度调整和航向调整。

Speed adjustments include accelerations and decelerations that are integer multiples of 10kt with a maximum range of 30kt , acceleration is indicated by ’+’ and deceleration by ’-’ . The speed action set is expressed as $A_{\mathrm{spd}} = \{+10, +20, +30, -10, -20, -30\}$.

速度调整包括加速和减速，它们是 10kt 的整数倍，最大范围为 30kt， 加速用’+’ 表示，减速用’-’ 表示。速度动作集表示为 $A_{\mathrm{spd}} = \{+10, +20, +30, -10, -20, -30\}$ 。

Altitude adjustments include ascents and descents in integer multiples of 600 m with a maximum range of 1200 m , ascent is indicated by ’ + ’ and descent by ’-’. The altitude action set is expressed as $A_{\mathrm{alt}} = \{+600, +1200, -600, -1200\}$.

高度调整包括上升和下降，它们是 600 m 的整数倍，最大范围为 1200 m， 上升用’+’ 表示，下降用’-’ 表示。高度动作集表示为 $A_{\mathrm{alt}} = \{+600, +1200, -600, -1200\}$ 。

Heading adjustments include offset and direct flight to the next waypoint, where the offset is 6 nm , a right offset is indicated by '+' and a left offset by '-'. The heading action set is expressed as $A_{\text{heading}} = \{$ DirectToNext, $+6, -6\}$.

航向调整包括偏移和直接飞向下一个航点，其中偏移是 6 nm，向右偏移用'+' 表示，向左偏移用'-' 表示。航向动作集表示为 $A_{\text{heading}} = \{$ DirectToNext, $+6, -6\}$。

So, the action space is expressed as $A = \{A_{\text{spd}}, A_{\text{alt}}, A_{\text{heading}}$, null$\}$, and there are 14 actions in total. If an aircraft does not need to change its state, it will receive a null instruction and continue to fly in its original state. If the above action instructions are received, the specified instructions need to be completed within the specified time.

因此，动作空间表示为 $A = \{A_{\text{spd}}, A_{\text{alt}}, A_{\text{heading}}$ ，null$\}$，总共有 14 个动作。如果飞机不需要改变其状态，它将接收到一个空指令并继续在其原始状态下飞行。如果接收到上述动作指令，指定的指令需要在规定的时间内完成。

(3) State transfer function

(3) 状态转移函数

The state transfer function is expressed as P_{sa} , which refers to the process by which the state of a certain aircraft is changed after the states s of all aircraft in the airspace at the current moment change to the next state s' after the instruction action a issued by the controller is taken, which is denoted as $p(s' \mid s, a)$ or $p(s', r \mid s, a)$ if the reward r is obtained. In this paper, deterministic state transfer is adopted, that is, after the action a is executed, the next state that the aircraft reach is uniquely determined according to the track prediction program. During the state transition, the track prediction program based on nominal data is used to calculate the flight path of aircraft quickly. The nominal data of commonly used aircraft types can be obtained from the Base of Aircraft Data (BADA) database.

状态转移函数表示为 P_{sa}，它指的是在当前时刻，空域中所有飞机的状态 s 发生变化后，某一特定飞机的状态如何经过控制器发出的指令动作 a 转变为下一状态 $p(s' \mid s, a)$ 或 $p(s', r \mid s, a)$ (如果获得了奖励 r)。在本文中，采用确定性的状态转移，即执行动作 a 后，飞机达到的下一状态是根据轨迹预测程序唯一确定的。在状态转换过程中，使用基于标称数据的轨迹预测程序来快速计算飞机的飞行路径。常用飞机类型的标称数据可以从飞机数据基础数据库 (BADA) 中获得。

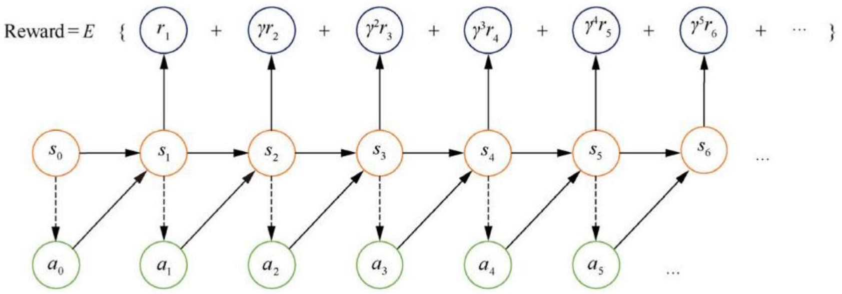
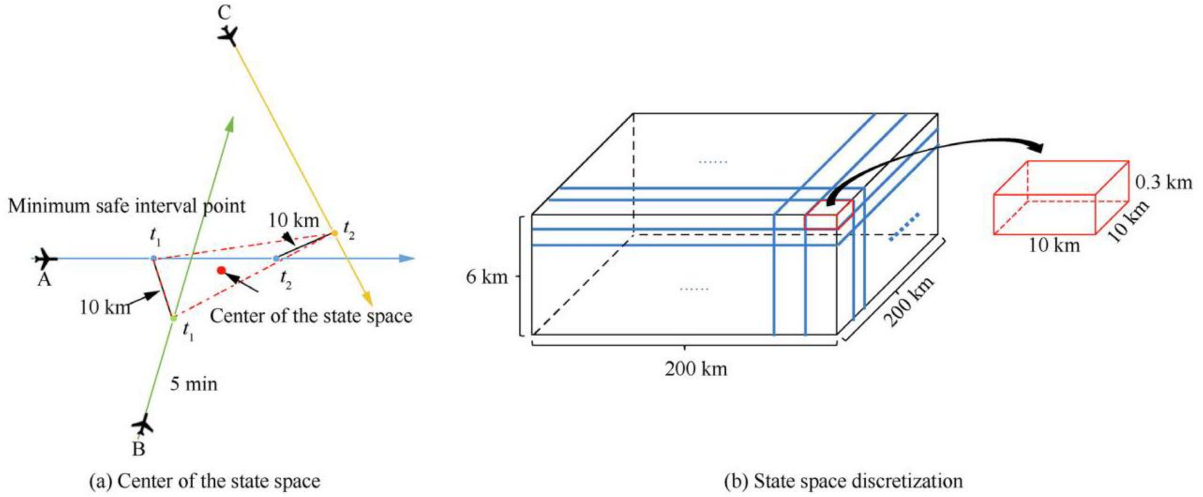

Reward = E { r_1 + γr_2 + $\gamma^2 r_3$ + $\gamma^3 r_4$ + $\gamma^4 r_5$ + $\gamma^5 r_6$ + ⋯ }
s_0
s_1
s_2
s_3
s_4
s_5
s_6
…
a_0
a_1
a_2
a_3
a_4
a_5
…

Fig. 4 MDP diagram with rewards.
图 4 带有奖励的马尔可夫决策过程 (MDP) 图。



(a) Center of the state space

(b) State space discretization

Fig. 5 Schematic diagram of the state space set.
图 5 状态空间集的示意图。

(4) Reward function

(4) 奖励函数

The reward function is expressed as R , it is an important tool to guide the intelligent behavior of agent trained by RL. During each state transfer process, the agent executes the selected action and moves to the next state, obtaining the corresponding reward value. Assuming the current state is s , the next state is s' that the agent reaches after executing action a , then the environment returns a reward r for the action a to the agent. The agent indicates the quality of the selected action through the reward function to guide the agent not to choose the action that yields a low accumulative reward. Due to the complete cooperation between multiple aircraft, the form of the reward function for each aircraft is the same. To measure the overall resolution effect and the execution effect of a single instruction, the reward function is divided into two parts: the overall reward and the individual reward. The overall reward measures the overall resolution effect after the whole conflict resolution process, and the individual reward measures a specific resolution instruction.

奖励函数表示为 R，它是指导通过强化学习训练的智能体智能行为的重要工具。在每个状态转移过程中，智能体执行所选动作并转移到下一个状态，获得相应的奖励值。假设当前状态为 s，智能体执行动作 a 后达到的下一个状态为 s'，然后环境为动作 a 返回一个奖励 r 给智能体。智能体通过奖励函数指示所选动作的质量，以指导智能体不要选择产生低累积奖励的动作。由于多架飞机之间的完全合作，每架飞机的奖励函数形式相同。为了衡量整体解决效果和单个指令的执行效果，奖励函数分为两部分: 整体奖励和个体奖励。整体奖励衡量整个冲突解决过程后的整体解决效果，个体奖励衡量特定的解决指令。

In the multi-aircraft flight conflict resolution model, the resolution time is stipulated to be 5 min of the complete process, and the timeline of the resolution instructions is as shown in Fig. 6. In the conflict scenario, the entire resolution time is 5 min. After the resolution, 5 min is added as the observation time to ensure the resolution result. The whole process lasts 10 min . Multi-aircraft conflict scenarios can be divided into multiple two-aircraft conflicts, and there are m conflict pairs in total. The time interval between the first and last conflict pairs is no more than 4 min according to the definition. In the resolution phase, when the aircraft detects the first conflict, the resolution begins. The first group of action instructions is sent to each conflicting aircraft at 30 s , and the aircraft executes the action instruction within the interval $30-120$ s . If the conflict is not resolved during the execution or a new conflict arises during this period, the scenario will be terminated and the next training episode will be started. If there is no new conflict until the end of execution, a certain reward will be given and the flight conflict scenario will end. If the work is carried out within $30-120$ s , no new conflict is generated, but a new conflict occurs after 120 s , the second group of action instructions is assigned to each conflicting aircraft at 120 s , and then the aircraft execute the action within the interval $120-210$ s . The execution is the same as in the above process. The controller has the right to issue three groups of instructions in total.

在多机飞行冲突解决模型中，解决时间规定为整个过程的 5 min，冲突解决指令的时间线如图 6 所示。在冲突场景中，整个解决时间为 5 分钟。解决后，5 min 作为观察时间加入，以确保解决结果。整个过程

持续 10 min 。多机冲突场景可以分为多个双机冲突，总共有 m 个冲突对。根据定义，第一个和最后一个冲突对之间的时间间隔不超过 4 min 。在解决阶段，当飞机检测到第一个冲突时，解决工作开始。第一组行动指令在 30 s 发送给每个冲突飞机，飞机在 30 – 120 s 的时间间隔内执行行动指令。如果在执行期间冲突未解决或在此期间出现新的冲突，场景将被终止并开始下一个训练环节。如果在执行结束时没有新的冲突，将会给予一定的奖励，飞行冲突场景结束。如果在 30 – 120 s 内完成工作，没有生成新的冲突，但在 120 s 之后出现新的冲突，则在 120 s 将第二组行动指令分配给每个冲突飞机，然后飞机在 120 – 210 s 的时间间隔内执行行动。执行过程与上述过程相同。控制器总共有权发出三组指令。

On the basis of the conflict resolution instruction timeline, a certain reward function should be given. As an important research object and a hyper-parameter in RL, the reward function needs to be determined before the whole training begins. At present, although there are studies on reward function learning, in most RL application cases, the reward function is still set according to the training target and modified according to many trials, so it is arbitrary. Initial attempts to use RL to resolve multi-aircraft flight conflicts have focused on the effectiveness, flexibility and practicality of strategies, and reward function set still uses artificial settings. When setting the reward function, the following basic principles should be followed:

在冲突解决指令时间线的基础上，应给予一定的奖励函数。作为强化学习 (RL) 中的一个重要研究对象和超参数，奖励函数需要在整个训练开始之前确定。目前，尽管有关于奖励函数学习的研究，但在大多数 RL 应用案例中，奖励函数仍然根据训练目标设置并经过多次尝试进行修改，因此具有任意性。最初尝试使用 RL 解决多机飞行冲突的研究主要集中在策略的有效性、灵活性和实用性上，奖励函数的设置仍然采用人工设定。在设置奖励函数时，应遵循以下基本原则:

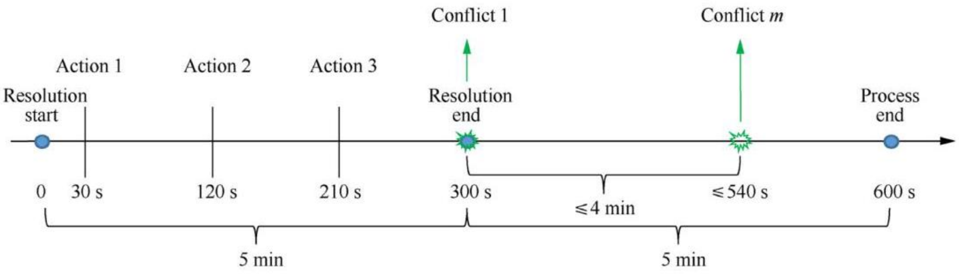


Fig. 6 Timeline of the IDQN-based multi-aircraft flight conflict resolution instructions.
图 6 IDQN 基于的多机飞行冲突解决指令时间线。

- The distribution of the reward function should show the recognition or the disapproval of the target taken by the agent, and the reward function usually increases monotonically as the expected value of the action increases.

- 奖励函数的分布应显示智能体对采取的目标的认可或反对，奖励函数通常随着动作预期值的增加而单调增加。

- The reward function can adapt to the change of environment caused by the action.

- 奖励函数可以适应由动作引起的环境变化。

- If the reward function contains several components, the magnitude of the reward function value of each component should be uniform, so as to make the evaluation of actions more balanced.

- 如果奖励函数包含多个组成部分，每个组成部分的奖励函数值的大小应保持一致，以便对动作的评价更加平衡。

- The reward function can be divided into the round reward, which only gets one reward at the end of the scenario, and the one-step reward, which gets one reward after each action.

- 奖励函数可以分为回合奖励，即在场景结束时只获得一次奖励，和单步奖励，即每次动作后获得一次奖励。

- Usually in RL, the reward function is finally unified into a mathematical expression.

- 通常在 RL 中，奖励函数最终统一为一个数学表达式。

Following the above principles, the setting of reward function in flight conflict problem needs to meet the following objectives: (A) Complete flight conflict resolution without causing new flight conflicts. (B) The amplitude of instructions should be as small as possible. (C) The resolution time is as short as possible. In order to get better training result, the design of reward function must take into account several factors. Firstly, the reward function is divided into individual reward $r_{\text{individual}}$ and overall reward r_{overall} . The individual reward is the evaluation of the actions taken by a single agent, while the overall reward is the evaluation of the overall resolution effect after the actions taken in each conflict scenario. Second, the solution of the problem can be divided into infeasible solution and feasible solution. The infeasible solution refers to the solution beyond the scope of aircraft performance or in violation of actual control habits, such as the climbing action followed by the descending action. The feasible solution is the solution other than the infeasible solution. In this part, the reward function is needed to distinguish the two in order to make the aircraft more likely to take more optimal solution within the range of feasible solutions. Then, in the process of seeking for more optimal solution, the constraints such as the amplitude of adjustment instructions, the time of resolution, the success of failure of resolution and whether new conflicts will be caused need to be taken into account. Thus, through the process of ’training-observation-modifying’, the reward function can be formed like the following.

遵循以上原则，飞行冲突问题中奖励函数的设置需要满足以下目标:(A) 完全解决飞行冲突，而不产生新的飞行冲突。(B) 指令的幅度应尽可能小。(C) 解决时间应尽可能短。为了获得更好的训练结果，奖励函数的设计必须考虑几个因素。首先，奖励函数分为个体奖励 $r_{\text{individual}}$ 和总体奖励 r_{overall} 。个体奖励是对单个代理采取的行动的评价，而总体奖励是在每个冲突场景采取行动后的整体解决效果的评价。其次，问题的解决方案可以分为不可行解决方案和可行解决方案。不可行解决方案是指超出飞机性能范围或违反实际控制习惯的解决方案，例如下降动作之后的爬升动作。可行解决方案是指除了不可行解决方案之外的其他解决方案。在这一部分，需要奖励函数区分这两种方案，以使飞机更可能在可行解决方案范围内采取更优的方案。然后，在寻求更优解决方案的过程中，需要考虑调整指令的幅度、解决时间、解决的成功与否以及是否会引发新的冲突等约束。因此，通过“训练-观察-修改”的过程，可以形成如下所示的奖励函数。

(1) From the perspective of a single agent, the reward function can be set in the form of Eq. (1), mainly considering whether it is a feasible solution and the adjustment range of the instruction.

(1) 从单个代理的角度来看，奖励函数可以设置为方程 (1) 的形式，主要考虑是否为可行解决方案以及指令的调整范围。

$$r_{\text{individual}} = \begin{cases} r_{\text{I}} & \text{, infeasible solution} \\ r_{\text{A}} + r_{\text{S}} + r_{\text{H}} & \text{, feasible solution} \end{cases} \tag{1}$$

(A) If the action falls within the range of infeasible solutions, the reward r_{I} is given according to Eq. (2), where p_{I} is the variable parameter, usually negative.

(A) 如果行动落在不可行解决方案的范围内，则根据方程 (2) 给予奖励 r_{I} ，其中 p_{I} 是变量参数，通常为负值。

$$r_{\text{I}} = p_{\text{I}} \tag{2}$$

(B) If the action falls within the range of feasible solution, the amplitude of the action is taken into account, and the greater the adjustment, the lower the reward. According to the different types of action taken, it can be divided into altitude reward r_{A} , speed reward r_{S} and heading reward r_{H} , and the rewards are given according to Eqs. (3)-(5) respectively.

(B) 如果行动在可行解的范围内，将考虑行动的幅度，调整越大，奖励越低。根据采取的不同类型的行动，可以分为高度奖励 r_{A} 、速度奖励 r_{S} 和航向奖励 r_{H} ，并根据公式 (3)-(5) 分别给予奖励。

(a) Altitude reward. The reward for an excessive altitude adjustment is shown like Eq. (3), where AltCmd represents a specific altitude instruction value, p_{A} represents the maximum reward value in the altitude adjustment, q_{A} is parameter that adjusts the magnitude.

(a) 高度奖励。过高调整高度所获得的奖励如公式 (3) 所示，其中 AltCmd 表示特定的海拔指令值，p_{A} 表示海拔调整中的最大奖励值，q_{A} 是调整幅度的参数。

$$r_{\text{A}} = p_{\text{A}} - |\text{AltCmd} \mid / q_{\text{A}}| \tag{3}$$

(b) Speed reward. The reward for an excessive speed adjustment is shown like Eq. (4), where SpdCmd represents a specific speed instruction value, p_{S} represents the maximum reward value in the speed adjustment, q_{S} is a parameter that adjusts the magnitude.

(b) 速度奖励。过高调整速度所获得的奖励如公式 (4) 所示，其中 SpdCmd 表示特定的速度指令值，p_{S} 表示速度调整中的最大奖励值，q_{S} 是调整幅度的参数。

$$r_{\mathrm{S}} = p_{\mathrm{S}} - |\mathrm{SpdCmd}| / q_{\mathrm{S}} \tag{4}$$

(c) Heading reward. The reward for a heading adjustment is shown like Eq. (5), p_{H} and q_{H} correspond to the reward value of different heading actions, which can be customized according to the user's preferences.

(c) 航向奖励。航向调整的奖励如公式 (5) 所示，p_{H} 和 q_{H} 分别对应不同航向行动的奖励值，可以根据用户的偏好进行自定义。

$$r_{\mathrm{H}} = \begin{cases} p_{\mathrm{H}}, & \text{direct to next waypoint} \\ q_{\mathrm{H}}, & \text{right or left offset} \end{cases} \tag{5}$$

(2) From the perspective of the effect of the overall conflict resolution process and combined with the instruction assignment timeline in Fig. 6, there may be four situations after an action is taken, and the rewards under different situations are given according to Eq. (6).

(2) 从整体冲突解决过程的效果角度来看，并结合图 6 中的指令分配时间线，采取行动后可能有四种情况，根据公式 (6) 给出不同情况下的奖励。

$$r_{\text{overall}} = \begin{cases} p_1 - (\text{nowtime} - \text{starttime}) / q_1 & \text{, Situation } 1 \\ p_2 & \text{, Situation } 2 \\ p_3 & \text{, Situation } 3 \\ p_4 & \text{, Situation } 4 \end{cases}$$

(6)

(A) Situation 1. Conflict resolution is successful. Calculate the reward according to the first equation in Eq. (6). The equation takes conflict resolution time into account, and the shorter the resolution time, the higher the reward value. The nowtime represents the time when the conflict is resolved and starttime represents the time when the resolution starts, p_1 represents the maximum possible reward, q_1 is a parameter that adjusts the magnitude.

(A) 情况 1. 冲突解决成功。根据公式 (6) 中的第一个公式计算奖励。该公式考虑了冲突解决时间，解决时间越短，奖励值越高。nowtime 表示冲突解决的时间，starttime 表示开始解决的时间，p_1 表示可能的最大奖励，q_1 是调整幅度的参数。

(B) Situation 2. Conflict resolution is failure and there are new conflicts. Calculate the reward according to the second equation in Eq. (6). In this case there is a negative reward $p_2 < 0$. In general, the penalty in this case is high, so the value of p_2 is relatively high.

(B) 情境 2. 冲突解决失败并且出现新的冲突。根据方程 (6) 中的第二个方程计算奖励。在这种情况下会有负奖励 $p_2 < 0$ 。通常，这种情况下的惩罚较高，因此 p_2 的值相对较高。

(C) Situation 3. Conflict resolution is failure and there are no new conflicts. Calculate the reward according to the third equation in Eq. (6). In this case there is a negative reward $p_3 < 0$. This punishment is slightly smaller than the punishment in situation 2, that is $|p_3| \leq |p_2|$.

(C) 情境 3. 冲突解决失败且没有新的冲突。根据方程 (6) 中的第三个方程计算奖励。在这种情况下会有负奖励 $p_3 < 0$ 。这种惩罚比情境 2 中的惩罚稍小，即 $|p_3| \leq |p_2|$ 。

(D) Situation 4. The execution is not successful within a certain period of time but the resolution is not complete. Calculate the reward according to the fourth equation in Eq. (6). In this case there is a reward $p_4 > 0$. Usually, this is a relatively small reward.

(D) 情境 4. 在一定时间内执行不成功但解决不完整。根据方程 (6) 中的第四个方程计算奖励。在这种情况下会有奖励 $p_4 > 0$ 。通常，这是一个相对较小的奖励。

Due to the particularity of the MAS, the overall resolution strategy should be optimized through cooperation among various aircraft. Therefore, an agent with a lower reward should be rewarded as much as possible through learning. The total reward value R is shown in Eq. (7), where $i \in n$ represents aircraft i .

由于多智能体系统 (MAS) 的特殊性，整体解决策略应通过各飞行器之间的合作进行优化。因此，应通过学习尽可能多地奖励奖励较低的代理。总奖励值 R 显示在方程 (7) 中，其中 $i \in n$ 代表飞行器 i 。

$$R = \min \left\{ r^{i}_{\text{individual}} \right\} + r_{\text{overall}} \tag{7}$$

4. Resolution scheme based on an IDQN

4. 基于 IDQN 的解决方案

After the multi-aircraft flight conflict problem is modeled, an appropriate method is selected to solve the model. The multi-aircraft flight conflict resolution strategy has dynamic and real-time requirements. Conflict scenarios contain high-dimensional data and require a large storage space and running memory. Therefore, the traditional RL algorithm cannot effectively address this problem. The DRL method developed in recent years makes use of the strong recording ability of neural networks, which can solve the dimensional problem well. Meanwhile, the method of offline training and online use greatly increases the solving speed. Therefore, the DRL method is adopted to solve the problem model. DRL was invented in 2013, and Google DeepMind designed the DQN algorithm. After years of development, the DQN, DDPG, Bic-Net, COMA and other algorithms have been proposed, but many aspects of the algorithm application ability in solving practical problems remain to be verified, so this paper chooses the DQN algorithm as the basic algorithm, and by using the framework of independent learning extensions in the field of multi-agent problems, the IDQN algorithm for the multi-aircraft flight conflict model is derived.

在多机飞行冲突问题建模之后，选择适当的方法来求解模型。多机飞行冲突解决策略具有动态和实时性的要求。冲突场景包含高维数据，需要大量的存储空间和运行内存。因此，传统的强化学习算法无法有效地解决这个问题。近年来发展的深度强化学习 (DRL) 方法利用了神经网络的强大记忆能力，可以很好地解决维度问题。同时，离线训练和在线使用的方法大大提高了求解速度。因此，采用 DRL 方法来求解问题模型。DRL 于 2013 年被发明，Google DeepMind 设计了 DQN 算法。经过多年的发展，提出了 DQN、DDPG、Bic-Net、COMA 等算法，但算法在解决实际问题方面的应用能力还有很多方面需要验证，所以本文选择 DQN 算法作为基本算法，并利用多代理问题领域的独立学习扩展框架，推导出适用于多机飞行冲突模型的 IDQN 算法。

4.1. Fundamentals of the algorithm

4.1. 算法基础

The RL process mainly consists of agent, environment, states, actions and rewards. The agent obtains the state through the environment, observes the state, uses the strategy to take an action, and gives feedback to the environment. After the environment executes the action, it transmits the new state as the new observation, and at the same time transmits the reward of the action to the agent. The agent continuously interacts with the environment through the above process and learns the best strategy by maximizing the cumulative reward.

强化学习过程主要由智能体、环境、状态、动作和奖励组成。智能体通过环境获得状态，观察状态，使用策略采取行动，并将反馈给环境。环境执行动作后，将新状态作为新的观察结果传递，同时传递动作的奖励给智能体。智能体通过上述过程与环境持续互动，并通过最大化累积奖励来学习最佳策略。

The DQN algorithm is a mature DRL algorithm, and the IDQN is the DQN algorithm using an independent learning framework, from a single-agent problem to a MAS. Before using the IDQN algorithm to solve the multi-aircraft flight conflict problem, the relevant fundamentals of the DQN algorithm and the independent learning framework are briefly introduced.

DQN 算法是一种成熟的深度强化学习算法，IDQN 是采用独立学习框架的 DQN 算法，从单一智能体问题到多智能体系统 (MAS)。在使用 IDQN 算法解决多机飞行冲突问题之前，先简要介绍 DQN 算法和独立学习框架的相关基础知识。

(1) DQN algorithm

(1)DQN 算法

The DQN algorithm is a DRL algorithm, and it combines Q-learning algorithms and neural networks which have powerful representation ability. It uses the information of the environment as the state of RL and as the input of the neural network model. Then, the neural network model outputs the value (Q value) corresponding to every action, and actions are executed. Therefore, the focus of the algorithm is to train the neural network model so that it can obtain the optimal mapping relationship between the environmental information and the action. There are a series of neurons in the neural network, and the weights of the neurons are constantly updated in the process of training, so it is necessary to set an objective function for them, which is generally a loss function. The DQN algorithm constructs the network optimized loss function through the Q -learning algorithm. According to the literature, [29] the update equation of the Q -learning algorithm is:

DQN 算法是一种深度强化学习算法，它结合了具有强大表征能力的 Q 学习算法和神经网络。它使用环境信息作为强化学习的状态和神经网络的输入。然后，神经网络模型输出每个动作对应的价值 (Q 值), 并执行动作。因此，算法的重点是训练神经网络模型，使其能够获得环境信息与动作之间的最优映射关系。神经网络中有一系列神经元，神经元的权重在训练过程中不断更新，因此需要为它们设置一个目标函

数，通常是一个损失函数。DQN 算法通过 Q 学习算法构建了优化损失函数的网络。根据文献 [29]，Q 学习算法的更新方程为:

$$Q*(s,a) \leftarrow Q(s,a)+\alpha\left[r+\gamma\max_{a'}Q(s',a')-Q(s,a)\right] \tag{8}$$

where α represents the learning rate, r represents the reward value of the action, γ represents the discount factor, $Q(s,a)$ represents the state action Q value at time step $t, Q(s',a')$ represents the state action Q value at time step $t+1$, and $r+\gamma\max_{a'}Q(s',a')$ represents the target Q value.

其中 α 代表学习率，r 代表动作的奖励值，γ 代表折扣因子，$Q(s,a)$ 代表时间步 $t, Q(s',a')$ 的状态动作 Q 价值，$t+1$ 代表时间步 $t, Q(s',a')$ 的状态动作 Q 价值，$r+\gamma\max_{a'}Q(s',a')$ 代表目标 Q 价值。

According to Eq. (8), the loss function of the DQN algorithm is defined as follows:

根据方程 (8)，DQN 算法的损失函数定义如下:

$$L(\theta)=E\left[(Q-Q(s,a,\theta))^2\right] \tag{9}$$

where θ represents the weight of the neural network model.

其中 θ 代表神经网络模型的权重。

Since the loss function in the DQN algorithm is determined based on the update equation of the Q value in the Q -learning algorithm, Eqs. (8) and (9) have the same meaning, and both approach the target Q value based on the current predicted Q value. After the loss function is obtained, the weights θ of the loss function $L(\theta)$ of the neural network model can be determined by the gradient descent algorithm.

由于 DQN 算法中的损失函数是基于 Q 学习算法中的 Q 值更新方程确定的，因此公式 (8) 和 (9) 具有相同的意义，都是基于当前预测的 Q 值来逼近目标 Q 值。获得损失函数后，可以通过梯度下降算法确定神经网络模型损失函数 $L(\theta)$ 的权重 θ 。

The DQN algorithm uses two key technologies: (A) a dual network structure. The DQN algorithm uses two neural networks for learning. The prediction network $Q\left(s,a,\theta_i^-\right)$ is used to evaluate the value function of the current state-action pair. The target network $Q\left(s,a,\theta_i^-\right)$ is used to generate the target Q value in Eq. (8). The algorithm updates the parameters θ in the prediction network according to the loss function in Eq. (9). After each N rounds of iteration, the parameters θ of the prediction network are copied to the parameters θ_i^- in the target network. (B) Experience replay. The DQN algorithm introduces the experience replay mechanism and stores the experience sample data (s,a,r,s',T) obtained from the interactions between agents and the environment at each time step into the experience pool. When network training is needed, a small batch of data is randomly selected from the experience pool for training. Using this mechanism, the reward data can be backed up, and the dependency and correlations among samples can be removed, which makes the model converge faster.

DQN 算法使用了两个关键技术:(A) 双网络结构。DQN 算法使用两个神经网络进行学习。预测网络 $Q\left(s,a,\theta_i^-\right)$ 用于评估当前状态-动作对的值函数。目标网络 $Q\left(s,a,\theta_i^-\right)$ 用于生成公式 (8) 中的目标 Q 值。算法根据公式 (9) 中的损失函数更新预测网络的参数 θ 。每经过 N 轮迭代后，预测网络的参数 θ 会被复制到目标网络的参数 θ_i^- 中。(B) 经验回放。DQN 算法引入了经验回放机制，并将每个时间步代理与环境交互获得的经验样本数据 (s,a,r,s',T) 存储到经验池中。当需要进行网络训练时，会从经验池中随机选取一小批数据进行训练。使用这种机制，可以备份奖励数据，并消除样本之间的依赖性和相关性，从而使模型更快收敛。

(2) Independent learning framework

(2) 独立学习框架

The DQN algorithm itself is designed for a single-agent problem. When the number of agents in the environment is greater than one, it becomes a multi-agent problem, the relationship among the agents becomes complicated, and any changes in the agent actions may influence other agents. Therefore, the problem of a MAS is a problem that many researchers have studied in recent years, and many research directions have developed from it. For example, agent communication behavior and the mutual modeling of other agents have been studied. [30] The independent learning framework is one of the research directions, which mainly concerns the behavior relations between agents, that is, whether agents in the MAS have cooperative relations, competitive relations or complex mixed relations, their relations are usually reflected by reward functions.

DQN 算法本身是针对单智能体问题设计的。当环境中的智能体数量大于一时，它变成了多智能体问题，智能体之间的关系变得复杂，任何智能体行为的改变都可能影响其他智能体。因此，多智能体系统 (MAS) 的问题是近年来许多研究者研究的课题，并从中发展出了许多研究方向。例如，研究了智能体的

通信行为和与其他智能体的相互建模。[30] 独立学习框架是其中的一个研究方向，主要关注智能体之间的行为关系，即 MAS 中的智能体是否具有合作关系、竞争关系或复杂的混合关系，它们的关系通常通过奖励函数来体现。

In the independent learning framework, each agent is regarded as an independent individual in the MAS, each agent has a neural network for training and studying, and there is no communication between them. This is a simple learning framework that extends the single-agent method to MAS problems. The IDQN directly extends the DQN algorithm to a MAS. Each Q value network controls a single agent. For systems with n agents, nQ value networks are required that correspond to them. Independence is assumed, which means that there is no coupling relationship between the Q value networks and no communication behavior between the networks, and data are independently sampled from the environment to update the network. The structure of the IDQN does not explicitly express cooperation between agents but allows agents to infer how cooperation should be conducted in the course of training based on changes in the state and reward functions. The framework of the IDQN algorithm is shown in Fig. 7.

在独立学习框架中，每个智能体被视为 MAS 中的一个独立个体，每个智能体都有一个神经网络用于训练和学习，它们之间没有通信。这是一个将单智能体方法扩展到 MAS 问题的简单学习框架。IDQN 直接将 DQN 算法扩展到 MAS。每个 Q 值网络控制一个智能体。对于有 n 个智能体的系统，需要 nQ 个与之对应的值网络。假设独立性，即 Q 值网络之间没有耦合关系，网络之间没有通信行为，数据独立地从环境中采样以更新网络。IDQN 的结构没有明确表达智能体之间的合作，但允许智能体在训练过程中基于状态和奖励函数的变化推断应该如何进行合作。IDQN 算法的框架如图 7 所示。

The independent learning framework is based on the assumption of independence while ignoring the problem of environmental instability. Although there is no guarantee of convergence in theory, many examples have shown that the algorithm can converge effectively in practice. In addition, in this algorithm, the neural networks of all agents share the same parameters. This technique of parameter sharing is widely used in RL, and it can improve the learning efficiency of agents, ensure the unity of the agent network structure, and improve the scalability of the algorithm.

独立学习框架基于独立性假设，同时忽视了环境不稳定的问题。尽管在理论上无法保证收敛性，但许多实例已经表明算法在实践中可以有效地收敛。此外，在该算法中，所有代理的神经网络共享相同的参数。这种参数共享技术在强化学习 (RL) 中被广泛使用，它可以提高代理的学习效率，确保代理网络结构的统一性，并提高算法的可扩展性。

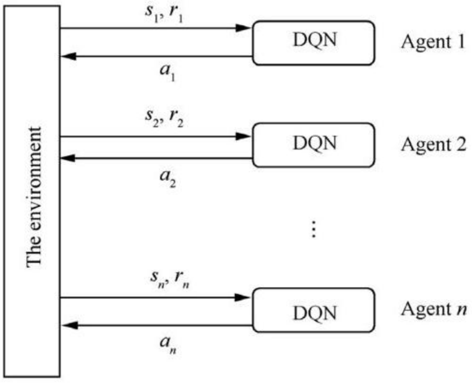


Fig. 7 IDQN algorithm framework.
图 7 IDQN 算法框架。

4.2. Aircraft cooperation mode

4.2. 飞机合作模式

Aircraft in multi-aircraft flight conflict scenarios are considered cooperative agents. Aircraft agents need to cooperate to resolve the situation. There is no possibility of sacrificing the interests of one aircraft agent to maximize the interests of other aircraft agents. The IDQN algorithm does not explicitly show the cooperative relationships among agents, and the relationships among agents are completely controlled by the reward function. The reward function in RL indicates the goal of solving the problem to a certain extent. Therefore, to balance the interests of the various aircraft agents and make the aircraft agents work together to complete the task, in the reward function, the minimum value of the individual reward and the overall reward of all aircraft agents are selected as the reward values in Eq. (7). The goal of RL is to maximize the cumulative reward, which will make the minimum value as high as possible. Only when aircraft agents tend to choose cooperative actions can they obtain a large reward.

在多飞机飞行冲突场景中，飞机被视为合作代理。飞机代理需要合作以解决问题。不存在牺牲某一飞机代理的利益以最大化其他飞机代理利益的可能性。IDQN 算法没有明确显示代理之间的合作关系，代理之间的关系完全由奖励函数控制。在强化学习 (RL) 中，奖励函数在一定程度上表示解决问题的目标。因此，为了平衡各个飞机代理的利益，并使飞机代理共同完成任务，在奖励函数中，选择了所有飞机代理的个体奖励和整体奖励的最小值作为公式 (7) 中的奖励值。强化学习的目标是最大化累积奖励，这将使最小值尽可能高。只有当飞机代理倾向于选择合作行为时，它们才能获得较大的奖励。

4.3. Dynamic expansion structure

4.3. 动态扩展结构

Because of the independent learning framework, each agent has a separate set of neural networks. Therefore, if the number of conflicting aircraft increases, neural networks with the same network structure need to be added for the additional aircraft and connected with each other by a reward function. If the number of conflicting aircraft is reduced, it is not necessary to change the structure or number of neural networks. In the scheme of multi-aircraft flight conflict resolution based on the IDQN algorithm, the ’downward-compatible’ mode is adopted. The resolution framework using the RL algorithm is designed to solve the problem of a flight conflict of $n\,(n \in \mathbf{N}^*)$ aircraft, and this framework can also be used to solve the problem of a flight conflict of $m\,(1 < m < n, m \in \mathbf{N}^*)$ aircraft. In this way, conflict scenarios with different numbers of conflicting aircraft can be mixed together for training. When using the trained model for testing, whenever a conflict scenario is imported, first judge the number of conflicting aircraft in the conflict scenario. When an n - aircraft flight conflict scenario is selected, all neural networks are selected for training. When an m -aircraft flight conflict scenario is selected, only the first m neural networks need to be selected for training. In this way, the corresponding neural network is selected to train for scenarios with different numbers of conflicting aircraft. When the training model is used for testing, flight conflict resolution is still carried out in accordance with the above selection mode. When using the RL method, it is necessary to ensure that the scenario is relatively fixed. For example, when only training the three-aircraft conflict scenario, it can usually only solve the three-aircraft conflict scenario during use. To resolve conflict scenarios with different numbers of conflicting aircraft, it needs to be trained separately for various flight conflict scenarios, and eventually several conflict resolution models need to be constructed. The advantage of using ’downward-compatible’ mode is that conflict scenarios with different numbers of conflicting aircraft can be put together for training and integrated into one conflict resolution model. In this way, when using, only one model needs to be called to avoid the burden of constructing the relatively repeated model.

由于独立学习框架，每个代理都有独立的神经网络集合。因此，如果冲突飞机的数量增加，则需要为额外的飞机添加具有相同网络结构的神经网络，并通过奖励函数相互连接。如果冲突飞机的数量减少，则无需更改神经网络的结构或数量。在基于 IDQN 算法的多飞机飞行冲突解决方案中，采用了“向下兼容”模式。使用 RL 算法的解决框架旨在解决 $n\,(n \in \mathbf{N}^*)$ 架飞机的飞行冲突问题，该框架也可用于解决 $m\,(1 < m < n, m \in \mathbf{N}^*)$ 架飞机的飞行冲突问题。这样，不同数量冲突飞机的冲突场景可以混合在一起进行训练。当使用训练好的模型进行测试时，每当导入一个冲突场景，首先判断该冲突场景中的冲突飞机数量。当选择 n 架飞机的飞行冲突场景时，选择所有神经网络进行训练。当选择 m 架飞机的飞行冲突场景被选中时，只需要选择前 m 个神经网络进行训练。这样，就可以为不同数量冲突飞机的场景选择相应的神经网络进行训练。当使用训练模型进行测试时，飞行冲突的解决仍然按照上述选择模式进行。使用 RL 方法时，需要确保场景相对固定。例如，当只训练三机冲突场景时，通常在应用中只能解决三机冲突场景。要解决不同数量冲突飞机的冲突场景，需要分别针对各种飞行冲突场景进行训练，并最终构建几个冲突解决模型。使用“向下兼容”模式的优点在于，不同数量冲突飞机的冲突场景可以一起训练并集成到一个冲突解决模型中。这样，在使用时，只需要调用一个模型，避免构建相对重复模型的负担。

4.4. Solution process

4.4. 解决过程

The IDQN algorithm is used to solve the problem model established in the third section. The DQN algorithm is a combination of a Convolutional Neural Network (CNN) and a Q-learning algorithm. The CNN has obvious advantages in high-dimensional image processing, but because the input data are the flight information in a designated airspace at a certain time, the CNN is not applicable. Therefore, it is replaced by a Multi-Layer Perceptron (MLP), which is more suitable for information data storage and processing. The essence of the IDQN algorithm is the DQN algorithm. It combines the MLP with Q-learning. Each agent has its own independent learning and training process. This method can effectively solve the problem of multi-aircraft flight conflict resolution. The main steps are as follows:

使用 IDQN 算法解决第三节中建立的模型问题。DQN 算法是卷积神经网络 (CNN) 和 Q 学习算法的结合。CNN 在高维图像处理中具有明显优势，但由于输入数据是特定空域在某一时间的飞行信息，CNN 不适用。因此，用多层感知器 (MLP) 替代，它更适合信息数据的存储和处理。IDQN 算法的本质是 DQN 算法。它将 MLP 与 Q 学习相结合。每个代理都有自己的独立学习和训练过程。这种方法可以有效解决多飞机飞行冲突问题。主要步骤如下:

Step 1. The multi-aircraft conflict scenario is input, and the state s_0 is initialized.

步骤 1. 输入多飞机冲突场景，并初始化状态 s_0 。

Step 2. Each agent selects actions a_0^i from the action space by using the selection strategy, and the actions selected by multiple agents are combined into joint actions

步骤 2. 每个代理使用选择策略从动作空间中选择动作 a_0^i ，并将多个代理选择的动作组合成联合动作。$u_0 = \{a_0^i\}_{i \in n}$.

Step 3. The simulated flight platform is used to execute the joint actions u_0 selected by each aircraft, and the reward value r_i and the next state s_1 are obtained according to the observation.

步骤 3. 使用模拟飞行平台执行每架飞机选择的联合动作 u_0 ，并根据观察获得奖励值 r_i 和下一个状态 s_1 。

Step 4. It is judged whether the next state s_1 meets the end condition. If the end condition is not satisfied, Steps 2 and 3 are continued. If the end condition is met, the current conflict scenario is reset, the new conflict scenario is input, and the algorithm starts again from Step 1.

步骤 4. 判断下一个状态 s_1 是否满足结束条件。如果结束条件不满足，继续执行步骤 2 和 3。如果结束条件满足，重置当前冲突场景，输入新的冲突场景，并从步骤 1 重新开始算法。

In summary, the specific algorithm flow of the IDQN algorithm used to train and solve the MDP-based control conflict resolution model is shown in Table 2.

总结来说，用于训练和解决基于 MDP 的控制器冲突解决模型的 IDQN 算法的具体流程如表 2 所示。

4.5. Model training and learning

4.5. 模型训练与学习

In this paper, the short-term conflict detection algorithm based on R-tree space query technology proposed by Li[31] was used to construct flight conflict scenarios needed in the study. By improving the steps of correlation search, the algorithm can be extended to multi-aircraft flight conflict detection problem. Import the flight plan into the trajectory prediction module, get the information of all aircrafts, use the R-tree for conflict detection, and record conflicting aircraft pairs. Then, according to the definition in Section 2.2, the conflict pairs are combined and filtered according to space limit, time limit, and connection limit to obtain multi-aircraft flight conflict scenarios. In the multi-aircraft flight conflict scenario obtained through this process, the flight number of the conflicting aircraft, the specific position of the conflict point, the time of the conflict, the flight plan of other aircraft in the scenario and other information are recorded.

在本文中，使用了由 Li[31] 提出的基于 R-tree 空间查询技术的短期冲突检测算法来构建研究中所需的飞行冲突场景。通过改进相关性搜索的步骤，算法可以扩展到多机飞行冲突检测问题。将飞行计划导入轨迹预测模块，获取所有飞机的信息，使用 R-tree 进行冲突检测，并记录冲突飞机对。然后，根据第 2.2 节的定义，按照空间限制、时间限制和连接限制将冲突对进行组合和筛选，以获得多机飞行冲突场景。通过此过程获得的多机飞行冲突场景中，记录了冲突飞机的航班号、冲突点的具体位置、冲突时间、场景中其他飞机的飞行计划等信息。

The model was trained on a computer with a Windows 7, 64-bit operating system with 32 GB of processor RAM. The training process is based on Python 3.7 in the PyCharm software. The core part of the algorithm uses the open source code of DQN published by OpenAI, and expands it into the form of IDQN, based on OpenAI Gym 0.15.4 version and TensorFlow 1.14.0 version to realize its function. Before training, multi-aircraft flight conflict scenario data should be constructed. Flight conflict detection is carried out through the Air Traffic Operation Simulation System (ATOSS) developed by Intelligent Air Traffic Control Laboratory. ATOSS uses the B/S (Browser/Server) architecture as a whole, and the user interface uses the VUE architecture, which mainly uses programming languages such as TS (TypeScript) JS (JavaScript) and CSS (Cascading Style Sheets). Its main function is to perform air traffic simulation and data analysis, and generate it in the form of simulation graphs and data analysis charts. The system mainly includes technologies such as track prediction, con-

模型在装有 Windows 7，64 位操作系统的计算机上进行了训练，处理器内存为 32 GB 。训练过程基于 PyCharm 软件中的 Python 3.7。算法核心部分使用了 OpenAI 发布的 DQN 开源代码，并将其扩展为 IDQN 形式，基于 OpenAI Gym 0.15.4 版本和 TensorFlow 1.14.0 版本来实现其功能。训练之前，应构建多机飞行冲突场景数据。通过智能空中交通控制实验室开发的空中交通运行模拟系统 (ATOSS) 进行飞行冲突检测。ATOSS 采用 B/S(浏览器/服务器) 架构整体，用户界面使用 VUE 架构，主要使用 TS(TypeScript)、JS(JavaScript) 和 CSS(层叠样式表) 等编程语言。其主要功能是执行空中交通模拟和数据分析，并以模拟图表和数据分析图表的形式生成。系统主要包含轨迹预测、冲突检测以及到达和起飞排序等技术。本文的研究主要使用了系统中的智能冲突检测技术。

Table 2 IDQN algorithm flow for the conflict resolution model. Algorithm: IDQN algorithm for the conflict resolution model

表 2 冲突解决模型的 IDQN 算法流程。算法: 冲突解决模型的 IDQN 算法

```
1. Initialize the experience replay pool of each agent with a capacity of $D$ .
. Initialize the action state value function $Q_i$ and randomly generate the weights $\theta$ .
  Initialize the target $Q$ network with weight $\theta^{-} = \theta$ .
  Loop through episodes $1, 2, \ldots, M$ .
    Randomly select the conflict scenario and initialize the state $s_0$ .
  Loop through the steps $1, 2, \ldots, T$ .
7. Each aircraft adopts an $\varepsilon$ - greedy strategy to select instruction actions $a_t^i$ from the action space to form joint actions $u_0$ .
8. Execute the joint instruction action $u_0$ according to the reward $r_i$ received and the new state $s_{t+1}$ of the aircraft.
    Save the conflict samples $\left(s_t, a_t^i, r_t, s_{t+1}\right)$ into the experience playback pool $D$ .
    Randomly select a conflict sample $\left(s_j, a_j^i, r_j, s_{j+1}\right)$ from the experience pool $D$ .
        Determine whether the step $j+1$ is final. In case of termination, set $y_j = r_j$ , otherwise, $y_j = r_j + \gamma \max_{d_i} Q\left(s_{j+1}, d_i'; \theta^{-}\right)$ .
      Calculate the loss function $L_i\left(\theta_i\right) = E_{\left(s,a,r,s'\right)}\left[\left(y_i^{\mathrm{DQN}} - Q\left(s, a; \theta_i\right)\right)^2\right]$ .
      Update $\theta$ in $\left(y_j - Q\left(s_j, a_j; \theta\right)\right)^2$ using gradient descent.
      Update the target $Q$ network for each $C$ step, $\theta^{-} = \theta$ .
    Until the steps end.
  Until the episodes end.
```

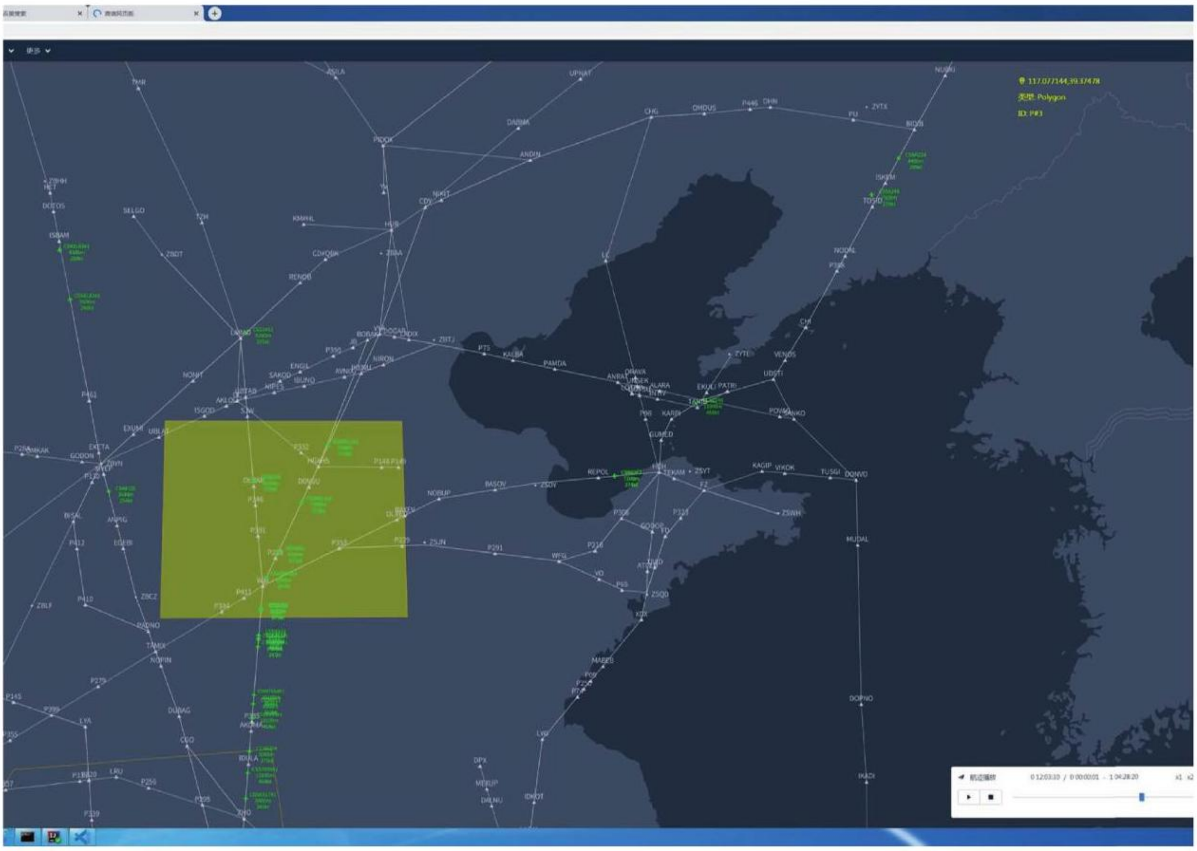


Fig. 8 Schematic diagram of conflict detection simulation scenario.

图 8 冲突检测模拟场景示意图。

flict detection, and arrival and departure sequencing. The research of this paper mainly uses the intelligent detection of conflicts technology in the system.

冲突检测、到达和起飞排序。

The flight plan data within China's airspace on June 1, 2018 are selected as the input, and random time changes are added to change the start times of the flight plans. The scenario size is set to 200 km × 200 km × 6 km during the detection process, as shown in Fig. 8. Each scenario contains a conflict of three, four or five aircraft, as well as other aircraft, and the number of aircraft is randomly distributed between 20 and 100. During the simulation, the multi-aircraft flight conflicts within the scope of the scenario are recorded. A total of 6700 multi-aircraft flight conflict scenarios are detected by using the conflict detection algorithm, among which three-aircraft conflict scenarios accounted for 70% , four-aircraft conflict scenarios accounted for 20% , and five-aircraft conflict scenarios accounted for 10% . The conflict scenarios are divided into training scenarios and test scenarios (6000 and 700, respectively), and the proportion of each type of conflict scenario is the same as above.

选择 2018 年 6 月 1 日中国空域内的飞行计划数据作为输入，并向飞行计划开始时间添加随机时间变化。如图 8 所示，在检测过程中，场景大小设置为 200 km × 200 km × 6 km 。每个场景包含三、四或五架飞机的冲突以及其他飞机，飞机数量在 20 到 100 之间随机分布。在模拟过程中，记录场景范围内的多架飞机飞行冲突。使用冲突检测算法共检测到 6700 个多架飞机飞行冲突场景，其中三架飞机冲突场景占 70% ，四架飞机冲突场景占 20% ，五架飞机冲突场景占 10% 。冲突场景被划分为训练场景和测试场景 (分别为 6000 个和 700 个)，每种类型冲突场景的比例与上述相同。

Multiple hyper-parameters are involved in the process of DRL training, such as learning rate, exploration rate and total time steps, etc., so the training parameters and their values required in the process of model training are sorted out as shown in Table 3. At the same time, set the values of the reward function parameters mentioned in Section 3.2, as shown in Table 4.

在深度强化学习 (DRL) 训练过程中涉及多个超参数，如学习率、探索率和总时间步等，因此，将模型训练过程中所需的训练参数及其值整理如表 3 所示。同时，设置第 3.2 节中提到的奖励函数参数的值，如表 4 所示。

It can be seen from Table 3 and Table 4 that model training involves multiple hyper-parameters, and different parameter values have different influences on the training effect of the model. Most of the parameter values are obtained through multiple tests and modifications. When the parameter values are different, it may affect the convergence speed of the model and the reward value. By referring to the research literature and relevant application research of DQN and combining with the test of flight conflict resolution, the values in Table 3 are finally determined. As can be seen from the analysis in Section 3.2, the first function of the reward is to distinguish the infeasible solution with feasible solution, and the second is to seek a better solution within the range of feasible solution. The setting of Eq. (2) can help the agent to avoid the action of choosing infeasible solution as soon as possible. It is a very important reward function. Since the parameter values of the reward function are set arbitrarily, the robust test was carried out here to illustrate with the parameter p_{I} of r_{I} in Eq. (2). Five parameter values were selected for the experiment, and the experimental results were shown in Table 5 and Fig. 9.

从表 3 和表 4 可以看出，模型训练涉及多个超参数，不同的参数值对模型的训练效果有不同的影响。大多数参数值是通过多次测试和修改得到的。当参数值不同时，可能会影响模型的收敛速度和奖励值。通过参考 DQN 的研究文献和相关应用研究，并结合飞行冲突解决的测试，最终确定了表 3 中的值。正如 3.2 节中的分析所示，奖励的第一个功能是区分不可行解和可行解，第二个是在可行解范围内寻求更好的解。式 (2) 的设置可以帮助代理尽可能快地避免选择不可行解的行为。这是一个非常重要的奖励函数。由于奖励函数的参数值是任意设置的，因此在这里进行了鲁棒性测试，以式 (2) 中的参数 p_{I} 的 r_{I} 为例。实验中选择了五个参数值，实验结果如表 5 和图 9 所示。

Table 3 Design values of the training parameters.

表 3 训练参数的设计值。

Training parameter	Value	Training parameter	Value
Number of agents	5	Exploration rate	0.1
Neural network	The connection layer	Exploration rate final episode	0.02
Number of hidden networks	64	Print frequency	100
Learning rate	1×10^{-3}	Batch size	64
Total time steps	3×10^{4}	Prioritized replay	TRUE
Buffer size	5×10^{4}	Parameter noise	TRUE

训练参数	价值	训练参数	价值
代理数量	5	探索率	0.1
神经网络	连接层	探索率最终剧集	0.02
隐藏网络的数目	64	打印频率	100
学习率	1×10^{-3}	批次大小	64
总时间步	3×10^{4}	优先重放	TRUE
缓冲区大小	5×10^{4}	参数噪声	TRUE

Table 4 Design values of the reward parameters.

表 4 奖励参数的设计值。

Reward parameter	Value	Reward parameter	Value
p_1	-1.0	q_{H}	0.25
p_{A}	1.0	p_1	1.0
q_{A}	2000	q_1	180
p_{s}	0.95	p_2	-3.0
q_{s}	100	p_3	-0.6
p_{H}	0.3	p_4	0.3

奖励参数	价值	奖励参数	价值
p_1	-1.0	q_{H}	0.25
p_{A}	1.0	p_1	1.0
q_{A}	2000	q_1	180
p_{s}	0.95	p_2	-3.0
q_{s}	100	p_3	-0.6
p_{H}	0.3	p_4	0.3

As can be seen from the first figure in Fig. 9, when the parameter value fluctuates slightly around -1.0, although the initial reward value is different and the convergence rate is slightly different, the reward value converges after the same timesteps of training, and it can be seen from Table 5 that the convergence value has little difference. When the parameter value is set to 0, it is equivalent to not having the constraint in Eq. (2). As can be seen from the second figure, the reward value is always fluctuating without a trend of convergence. When the parameter value is set to -10.0, the training effect is not good due to the great difference between the setting of this reward value and other reward values. In the training process of the same step number, the reward value does not converge, but also shows a trend of gradual decline. Meanwhile, due to the large deviation with other reward function values, it may lead to the problem of sparse reward, which is not conducive to the stability of the model. Therefore, when there is a small fluctuation between -0.5 and -1.5, the stability of the model will not be greatly affected, and the model has a certain robustness. The values of other parameters can also be analyzed by testing and comparing results to select the most appropriate value in a limited range.

从图 9 的第一个图中可以看出，当参数值在-1.0 附近轻微波动时，尽管初始奖励值不同，收敛速率略有差异，但在相同的训练步数后，奖励值会收敛，且从表 5 中可以看出，收敛值差异不大。当参数值设置为 0 时，相当于没有式 (2) 中的约束。从第二个图中可以看出，奖励值总是在波动，没有收敛的趋势。当参数值设置为-10.0 时，由于此奖励值与其他奖励值的设置差异较大，训练效果不佳。在相同步数的训练过程中，奖励值没有收敛，反而呈现出逐渐下降的趋势。同时，由于与其他奖励函数值偏差较大，可能导致稀疏奖励问题，不利于模型的稳定性。因此，当在-0.5 到-1.5 之间有小的波动时，模型的稳定性不会受到很大影响，模型具有一定的鲁棒性。通过测试和比较结果，也可以分析其他参数的值，以在有限范围内选择最合适的值。

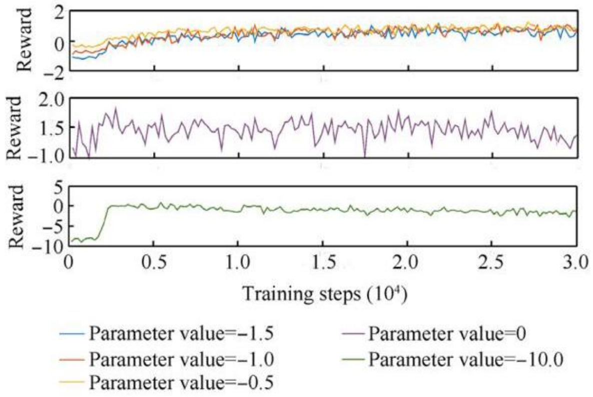
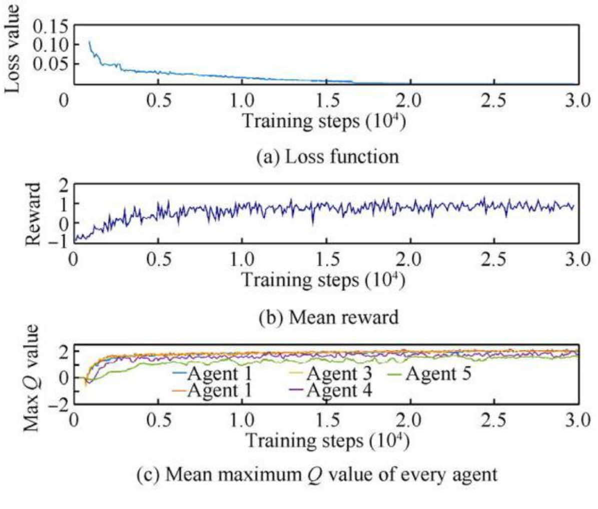


Fig. 9 Reward function curve of different parameter p_{I} value.
图 9 不同参数 p_{I} 值的奖励函数曲线。



(a) Loss function

(b) Mean reward

(c) Mean maximum Q value of every agent

Fig. 10 Chart of the model training data.
图 10 模型训练数据的图表。

After the training parameters are determined, the final model training is carried out. The model is trained 30,000 steps at a time, and each training time is 50 h . The trained model shows good stability and convergence. The graphs shown in Fig. 10 are its training curves, corresponding to the loss function value, the average reward value and the average maximum Q value of each agent. The loss function is used to evaluate the degree to which the predicted value of the model is different from the real value, which is usually non-negative. The smaller the loss function is, the better the model performance will be. The reward function is the evaluation of the action taken by an agent. The higher the reward, the more the action is encouraged to be taken. When the average reward is higher, it indicates that the agent can take the correct action most of the time. The Q value is updated according to the reward function. The agent acts according to the maximum Q value. The higher the maximum Q value, the more correct the agent can take the action.

确定训练参数后，进行最终的模型训练。模型每次训练 30,000 步，每次训练时间 50 h 。训练后的模型表现出良好的稳定性和收敛性。图 10 所示的图表是其训练曲线，分别对应损失函数值、平均奖励值以及每个代理的平均最大 Q 值。损失函数用于评估模型的预测值与实际值的差异程度，通常为非负值。损失函数越小，模型性能越好。奖励函数是对代理采取的行动的评价。奖励越高，越鼓励采取该行动。当平均奖励较高时，表明代理大多数时间能够采取正确的行动。Q 值根据奖励函数更新。代理根据最大 Q 值行动。最大 Q 值越高，代理能够采取的正确行动越多。

Table 5 Training results of different parameter p_1 value.

表 5 不同参数 p_1 值的训练结果。

p_{I} value	Initial reward value	Convergence	Convergence reward
0	1.16	No convergence	
-0.5	-0.2565	Convergent	0.8171
-1.0	-0.8420	Convergent	0.8663
-1.5	-1.0515	Convergent	0.6360
-10.0	-8.837	No convergence	

p_{I} 价值	初始奖励值	收敛	收敛奖励
0	1.16	未收敛	
-0.5	-0.2565	收敛的	0.8171
-1.0	-0.8420	收敛的	0.8663
-1.5	-1.0515	收敛的	0.6360
-10.0	-8.837	未收敛	

Fig. 10(a) shows that the loss function of the model starts to decline rapidly when the training reaches approximately 1000 steps and then gradually stabilizes at approximately 0 at approximately 15600 steps, indicating that the agent is learning. Fig. 10(b) shows that at the beginning of training, the average reward value of the model is low. With the increase in the number of training steps, the reward value gradually increases. Fig. 10(c) shows that the maximum Q value of an average single agent starts to rise after approximately 1000 steps and gradually stabilizes around a higher value after approximately 2000 steps. As seen from the figure, as the number of training steps increases, the agents gradually show good learning behavior. Although these five curves are different, their convergence properties show that the model proposed in this paper is effective in resolving conflicts equal to or less than five aircraft. The learning characteristics of the 'downward-compatible' model can also be seen in Fig. 10(c). The aircraft agent will select the neural network in order, and the number of the conflict scenarios will decrease with the increase of conflicting aircraft. Therefore, the first three are fully trained, converge quickly, and have a large maximum Q value. Although the convergence of the fourth and fifth aircraft agents is slow and the maximum Q value is small, they still show convergence results. Thus, 'downward-compatible' framework is valid.

图 10(a) 显示，当训练达到大约 1000 步时，模型的损失函数开始迅速下降，然后在大约 15600 步时逐渐稳定在大约 0，表明代理正在学习。图 10(b) 显示，在训练开始时，模型的平均奖励值较低。随着训练步数的增加，奖励值逐渐增加。图 10(c) 显示，平均单个代理的最大 Q 值在大约 1000 步后开始上升，并在大约 2000 步后逐渐稳定在一个较高的值附近。从图中可以看出，随着训练步数的增加，代理逐渐展现出良好的学习行为。尽管这五条曲线各不相同，但它们的收敛性表明本文提出的模型在解决等于或小于五架飞机的冲突中是有效的。在图 10(c) 中也可以看到“向下兼容”模型的学习特性。飞机代理将按顺序选择神经网络，随着冲突飞机数量的增加，冲突场景的数量将减少。因此，前三个完全训练，快速收敛，并且具有较大的最大 Q 值。尽管第四和第五架飞机代理的收敛速度慢，最大 Q 值小，但它们仍然显示出收敛结果。因此，“向下兼容”框架是有效的。

The above analysis shows that the model is stable, robust and convergent after sufficient training and learning of conflict scenario samples and training steps, which can be used for multi-aircraft flight conflict scenario tests. To further improve the training effect of the model, its performance is improved by increasing the number of training steps and adjusting the relevant hyper-parameters in Table 3 to make the model more stable.

上述分析表明，在足够的训练和学习冲突场景样本以及训练步数之后，模型是稳定、健壮且收敛的，可以用于多飞机飞行冲突场景测试。为了进一步提高模型的训练效果，通过增加训练步数和调整表 3 中的相关超参数来改进模型的性能，使模型更加稳定。

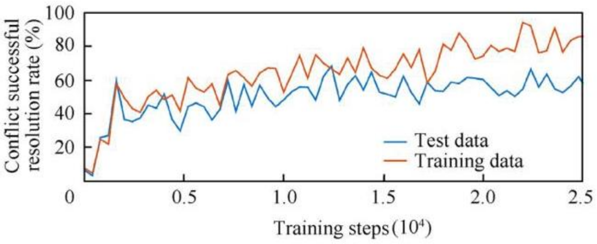


Fig. 11 Change curve of the resolution rate with the number of training steps.
图 11 展示了解决率随训练步数变化的曲线。

5. Result analysis

5. 结果分析

For the trained model, a test scenario should be used to verify its validity. The data used in the test should be different from the data used in the training. Therefore, the 700 reserved conflict scenarios should be used for testing. In the analysis of the test results, the indicators like successful conflict resolution rate, conflict resolution trends, the delay time after resolution, calculation time and distribution of successful conflict resolution rate are primarily considered.

对于训练好的模型，应使用测试场景来验证其有效性。测试中使用的数据应与训练中使用的数据不同。因此，应使用保留的 700 个冲突场景进行测试。在测试结果分析中，主要考虑了成功冲突解决率、冲突解决趋势、解决后的延迟时间、计算时间以及成功冲突解决率的分布等指标。

(1) Successful conflict resolution rate

(1) 成功冲突解决率

The successful conflict resolution rate represents the ratio of the number of flight conflict successful resolution scenarios to all test scenarios. It is the most direct indicator to measure the effect of the model, the higher the value, the better the resolution effect of the model. Fig. 11 shows a positive proportion between the successful conflict resolution rate and the number of training steps. At the same time, the training data and test data are used to test the model, indicating the generalization ability of the model. When the number of training steps is sufficient, the conflict resolution rate of the model is higher, and the reserved test data can be used to test it.

成功冲突解决率表示成功解决冲突场景数量与所有测试场景数量的比例。它是衡量模型效果的最直接指标，值越高，模型的解决效果越好。图 11 显示了成功冲突解决率与训练步数之间的正比关系。同时，使用训练数据和测试数据来测试模型，表明了模型的泛化能力。当训练步数足够时，模型的冲突解决率较高，可以使用保留的测试数据来进行测试。

Among the 700 conflict scenarios, a total of 600 conflict scenarios were successfully resolved, and the successful conflict resolution rate was approximately 85.71% , including 460 three-aircraft conflicts, 110 four-aircraft conflicts and 30 five-aircraft conflicts, totaling 1970 sorties. The distribution of the reward obtained by the successful resolution scenarios is shown in Fig. 12. The reward value is in the range of 1.85- 2.75, and the number of scenarios in which the conflict was successfully resolved with a reward value of 2.05 is the largest. If the number of scenarios that are successfully resolved at a certain reward value is greater, it indicates that the probability that the model is stable at the reward value when conflict is successfully resolved. The test results show that the training of the model achieves some stability, but its distribution is relatively discrete. In the future, to improve the stability of the model, the number of training steps and training samples can be increased. At the same time, this can also improve the successful resolution rate of the model.

在 700 个冲突场景中，共有 600 个冲突场景得到了成功解决，成功解决冲突的比例约为 85.71% ，包括 460 个三机冲突，110 个四机冲突和 30 个五机冲突，总计 1970 架次。成功解决冲突场景所获得的奖励分布如图 12 所示。奖励值的范围在 1.85-2.75 之间，其中以 2.05 的奖励值成功解决冲突的场景数量最多。如果在某一奖励值上成功解决的场景数量较多，则表明模型在该奖励值下解决冲突时具有稳定性。测试结果表明，模型的训练达到了一定的稳定性，但其分布相对离散。在未来，为了提高模型的稳定性，可以增加训练步数和训练样本数量。同时，这也可以提高模型的成功解决率。

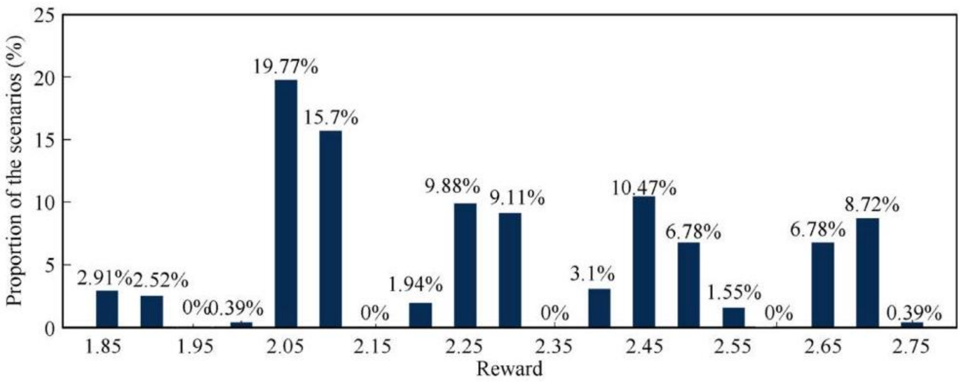


Fig. 12 Reward for successfully resolved scenarios.

图 12 成功解决场景的奖励。

In the test, 100 scenarios were not successfully resolved. Most were due to new conflicts with other aircraft in the environment and a small number were due to the failure of their own conflict resolution. The reasons for the failures may be analyzed as follows: (A) a model using RL training will not achieve a 100% successful resolution rate because of the difference between the training samples and the test samples, (B) the setting of the hyper-parameters and reward function in the algorithm impact the training effect of the model, and the reward function is not comprehensive enough to avoid the flights of other aircraft in the environment, (C) the training steps of the model are insufficient. Through the training of existing steps, the model obtains convergent and stable results, but the characteristics of all training scenarios have not been learned. Through the analysis above, the successful resolution rate of the model can be improved by changing the hyper-parameter settings, reward function and training steps.

在测试中，有 100 个场景未能成功解决。大多数是因为与环境中其他飞机的新冲突，少数是因为自身冲突解决失败。失败的原因可以分析如下:(A) 使用强化学习训练的模型由于训练样本与测试样本之间的差异，无法达到 100% 成功解决率；(B) 算法中超参数的设置和奖励函数影响了模型的训练效果，且奖励函数不足以避免环境中其他飞机的飞行；(C) 模型的训练步骤不足。通过现有步骤的训练，模型获得了收敛和稳定的结果，但并未学习到所有训练场景的特征。通过以上分析，可以通过改变超参数设置、奖励函数和训练步骤来提高模型的成功解决率。

(2) Conflict resolution trends

(2) 冲突解决趋势

In successful conflict resolution scenarios, it is necessary to verify the requirements of various aspects of the scenario. First, it is necessary to verify whether the resolution strategy adopted complies with the actual controllers' habits. The distribution of the resolution actions taken by the 1970 aircraft is shown in Fig. 13. The abscissa corresponds to the action instructions in the action space. Fig. 13 shows that the action with the highest selection probability is ascending 600 m . In each category, aircraft tend to choose the action with a small-amplitude adjustment. Comprehensive statistical results showed that 12.4% of the aircraft did not change, 33.33% of the aircraft chose speed adjustments, 29.53% of the aircraft chose altitude adjustments, 24.71% of the aircraft chose heading adjustments, and the overall order of priority selection was speed > altitude > heading, which is consistent with most of the area controllers' habits.

在成功的冲突解决场景中，需要验证场景各方面的要求。首先，需要验证采用的解决策略是否符合实际控制员的习惯。1970 架飞机采取的解决行动分布如图 13 所示。横坐标对应动作空间中的动作指令。图 13 显示，选择概率最高的动作是上升 600 m 。在每个类别中，飞机倾向于选择小幅调整的动作。综合统计结果显示，12.4% 的飞机没有改变，33.33% 的飞机选择了速度调整，29.53% 的飞机选择了高度调整，24.71% 的飞机选择了航向调整，整体优先选择顺序为速度 > 高度 > 航向，这与大多数区域控制员的习惯相符。

In the test results, some actions are not selected at all because of the limitations of aircraft performance. The aircraft has minimum and maximum cruising speeds at cruising altitude, i.e., speed adjustment range limits. Taking the A320 aircraft type as an example, as shown in Fig. 14, as the flight level rises, its cruising speed adjustment range gradually decreases, eventually reaching approximately 60kt . The standard cruise speed is usually between the maximum and minimum cruise speeds, so it can be adjusted up and down by approximately 30kt . The selection range of the test resolution action shown in the model is 20kt up and down, so the resolution strategy meets the performance limit.

在测试结果中，由于飞机性能的限制，某些动作根本没有被选中。飞机在巡航高度具有最小和最大巡航速度，即速度调整范围限制。以 A320 机型为例，如图 14 所示，随着飞行高度的升高，其巡航速度调整范围逐渐减小，最终达到大约 60kt 。标准巡航速度通常在最大和最小巡航速度之间，因此可以大约调整 30kt 。模型中显示的测试分辨率动作的选择范围是 20kt 上下，因此分辨率策略满足性能限制。

In the resolution strategy, the aircraft tends to choose the action instruction with a smaller range. The result of this instruction resolution is more in line with actual control and resolution habits. Both controllers and pilots prefer to choose the simplest way to solve a conflict. At the same time, an action with a small adjustment range can make the aircraft arrive at the destination as close to the original planned arrival time as possible to reduce the impact of conflict resolution on its delay time.

在分辨率策略中，飞机倾向于选择调整范围较小的动作指令。这种指令解析的结果更符合实际控制和解析习惯。控制器和飞行员都倾向于选择最简单的方式来解决冲突。同时，调整范围小的动作可以使飞机尽可能接近原计划到达时间到达目的地，以减少冲突解析对其延误时间的影响。

(3) Delay time

(3) 延迟时间

In the process of conflict resolution, in addition to achieving the successful resolution of conflicts, the aircraft should be ensured to arrive at the destination in accordance with the planned arrival time as far as possible to reduce the impact on the flight plans of other aircraft. The delay times of the aircraft successfully resolved in the above conflicts were analyzed, and the flight time after resolution minus the original planned flight time was used as the delay time (a value greater than 0 indicates a delay, and less than 0 indicates advance arrival). The smaller the absolute value of the delay time, the smaller the impact on normal flight. To better display the distribution of the delay time, the time is divided into intervals of 1 min and expressed in the form of intervals, as shown in Fig. 15. The delay time approximately follows a normal distribution. Thirty-two percent of the aircraft arrive at the destination within 30 s before or after the original planned arrival time, and 71.51% of the aircraft can reach the destination within 150 s before or after the original planned arrival time. The reason for this result is related to the selection of the resolution action. The delay time is within the acceptable range, and it will not cause great interference to the flight plans of other aircraft, reducing the workload of the air traffic controllers in handling the aircraft operating on the airport surface. Therefore, using this model to resolve multi-aircraft flight conflicts will not affect the normal operation of flights to ensure the resolution rate.

在冲突解决过程中，除了实现冲突的成功解决外，还应确保飞机尽可能按照计划到达时间到达目的地，以减少对其他飞机飞行计划的影响。分析了上述冲突中成功解决的飞机的延误时间，将解决后的飞行时间减去原计划飞行时间作为延误时间 (大于 0 的值表示延误，小于 0 的值表示提前到达)。延误时间的绝对值越小，对正常飞行的影响越小。为了更好地显示延误时间的分布，将时间分为 1 min 的区间，并以区间形式表示，如图 15 所示。延误时间大致遵循正态分布。百分之三十二的飞机在原计划到达时间前后 30 s 内到达目的地，而 71.51% 的飞机可以在原计划到达时间前后 150 s 内到达目的地。这个结果与解决行动的选择有关。延误时间在可接受范围内，不会对其他飞机的飞行计划造成很大干扰，减少了空中交通管制员在处理机场地面运行的飞机时的工作量。因此，使用此模型解决多架飞机的飞行冲突不会影响航班的正常运营，以确保解决率。

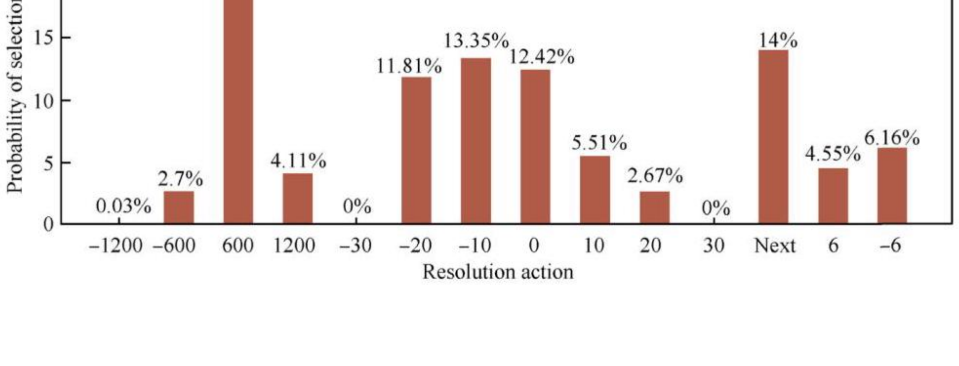


Fig. 13 Statistics of conflict resolution actions.
图 13 冲突解决行为的统计数据。

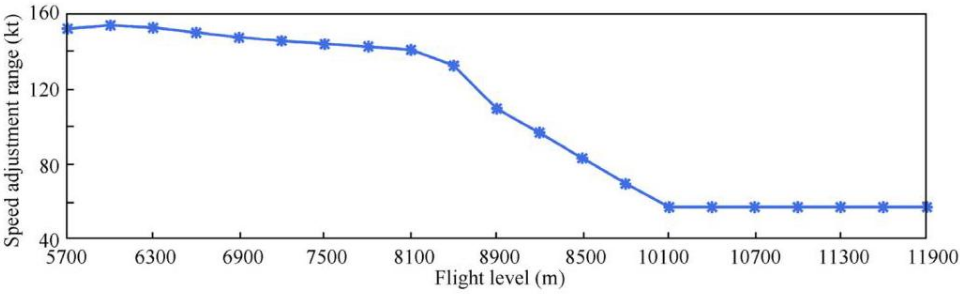


Fig. 14 Cruise speed adjustment ranges of A320 aircraft at different flight altitudes.
图 14 不同飞行高度下 A320 飞机的巡航速度调整范围。

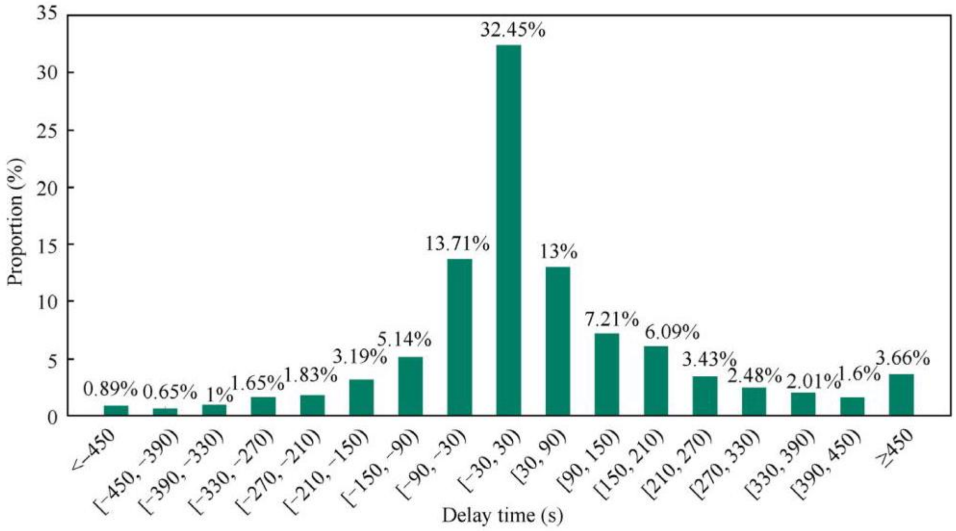


Fig. 15 Statistics of the aircraft delay time.
图 15 飞机延误时间统计。

(4) Comparative analysis of calculation time

(4) 计算时间的比较分析

One of the advantages of RL is that the method of offline training and online using can greatly increase the solving speed and improve the solving efficiency. In order to verify that the algorithm proposed in this paper has a high solving speed, the NO. 7 method in Table 1 is selected as a comparison algorithm. Through the test of 700 conflict scenarios, the same as IDQN, the calculation time distribution of GA algorithms is shown in Fig. 16. The solving time of GA is distributed between 12.6591 s and 141.5892 s , and its average solving time is 37.6003 s . The calculation time of the GA is related to the number of aircraft in the scenario and the generation. The more the number of aircraft, the more complex the airspace environment, the larger the iteration generation required, and the longer the calculation time. However, the IDQN algorithm is less affected by the number of aircraft. Due to the training of a large number of samples, it has better adaptability in actual solving. In the solution of the proposed IDQN algorithm, the calculation time distribution of the conflict resolution strategy is between $7.0004\mathrm{e}-3$ and $2.2001\mathrm{e}-2$, and the average calculation time is $1.1056\mathrm{e}-2$. The calculation time is at a different order of magnitude from that of GA. Through comparison, it can be found that DRL algorithms such as IDQN have great advantages in solving time, so it can improve the solving efficiency.

强化学习 (RL) 的一个优势在于，离线训练和在线使用的方法可以大大提高解题速度和改善解题效率。为了验证本文提出的算法具有较高的解题速度，选取表 1 中的第 7 种方法作为对比算法。通过 700 个

冲突场景的测试，与 IDQN 相同，遗传算法 (GA) 的计算时间分布如图 16 所示。GA 的解题时间分布在 12.6591 s 到 141.5892 s 之间，其平均解题时间为 37.6003 s 。GA 的计算时间与场景中飞机的数量和代数有关。飞机数量越多，空域环境越复杂，所需的迭代代数越大，计算时间越长。然而，IDQN 算法受飞机数量的影响较小。由于大量样本的训练，它在实际解题中具有更好的适应性。在所提出 IDQN 算法的解决方案中，冲突解决策略的计算时间分布在 7.0004e − 3 到 2.2001e − 2 之间，平均计算时间为 1.1056e − 2 。计算时间与 GA 的量级不同。通过比较可以发现，如 IDQN 这样的深度强化学习 (DRL) 算法在解题时间上具有很大优势，因此可以提升解题效率。

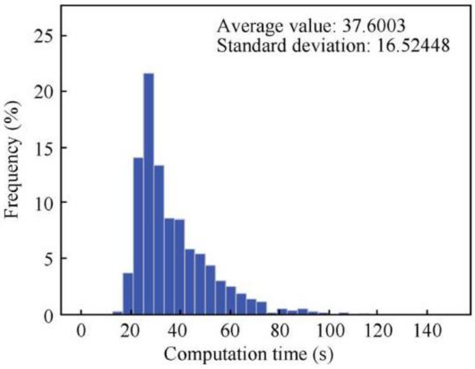


Fig. 16 Distribution of GA calculation time.
图 16 遗传算法计算时间分布。

(5) Distribution of successful conflict resolution rate

(5) 成功冲突解决率的分布

From the first indicator, it can be seen that not all multi-aircraft conflict scenarios can obtain an effective resolution strategy during the test of the proposed algorithm. It has a certain failure rate, so it is necessary to add the test of the level of uncertainty that the algorithm fails. The level of uncertainty is defined as the probability range of successful conflict resolution rate in multiple tests, the smaller the range, the more stable the model. In this part, 70 flight conflict scenarios were randomly selected for each test, a total of 1000 times tests were conducted. In each round of testing, the successful conflict resolution rate was calculated by dividing the number of resolved scenarios by 70, expressed as a percentage. The results of 1000 tests were calculated and the data were divided into quartile and represented as frequency histograms like Fig. 17. It shows the results of 1000 tests, with the rate ranging from 70.00% to 88.57% . Using the distribution curve to fit, it is found that the random variable successful conflict resolution rate (represented by X , decimal form) approximately follows the normal distribution $X \sim N\left(0.81, 0.042^2\right)$. This shows that the results of the algorithm are generally stable. The level of uncertainty of the algorithm is acceptable.

从第一个指标可以看出，并非所有多机冲突场景在测试所提出算法时都能获得有效的解决策略。它有一定的失败率，因此需要增加算法失败时不确定性的测试等级。不确定性等级定义为多次测试中成功解决冲突率的概率范围，范围越小，模型越稳定。在这一部分，每次测试随机选取了 70 个飞行冲突场景，共进行了 1000 次测试。在每一轮测试中，通过将解决场景数除以 70 来计算成功解决冲突率，并以百分比表示。1000 次测试的结果被计算出来，并将数据分为四分位数，以频率直方图的形式表示，如图 17 所示。它显示了 1000 次测试的结果，成功率范围从 70.00% 到 88.57% 。使用分布曲线拟合，发现随机变量成功冲突解决率 (由 X 表示，小数形式) 近似遵循正态分布 $X \sim N\left(0.81, 0.042^2\right)$ 。这表明算法的结果通常比较稳定。算法的不确定性水平是可以接受的。

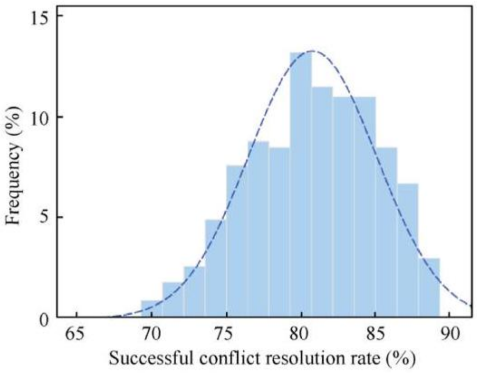


Fig. 17 Distribution of test results for level of uncertainty.
图 17 不确定性等级测试结果分布。

6. Discussion

6. 讨论

Through the above analysis, the flight conflict resolution model based on DRL can successfully resolve the most conflict scenarios. Under actual constrained flight conditions, flight conflicts among multiple aircraft are successfully resolved based on the control operational regulations. The overall resolution effect is measured by the delay time, and the resolution process has little impact on the normal flights of other aircraft. However, the model also has some shortcomings. In the actual control procedure, the successful resolution rate is required to be 100% . However, in the model, the rate does not meet this requirement, but the uncertainty of the algorithm is generally acceptable. So it can only be used as an assistant decision-making method for controllers. In addition, although the test results show that the action selection is more in line with control habits than previous methods, in actual area control, the priority resolution actions of controllers in different sectors are different. Conflict resolution actions often depend on the personal habits of the controller and the preferences of the aircraft crew. At the same time, the relative position, route structure and airspace situation of the aircraft should be considered. To meet the actual resolution habits of air traffic controllers, it is necessary to conduct empirical statistics for a sector or a controller and then modify the reward function to make the training results conform to their resolution habits. Generally, solving the problem of multi-aircraft flight conflicts by using the framework of DRL and independent learning is successful. By comparing with the basic algorithm, it is proved that the algorithm proposed has great advantages in solving efficiency. It can ensure safe and efficient of the high-density airspace in the future.

通过上述分析，基于深度强化学习 (DRL) 的飞行冲突解决模型可以成功解决大多数冲突场景。在实际受限的飞行条件下，根据控制操作规程，成功解决了多架飞机之间的飞行冲突。整体解决效果通过延迟时间来衡量，解决过程对其他飞机的正常飞行影响较小。然而，该模型也存在一些不足。在实际控制过程中，成功解决率需要达到 100% 。但在模型中，这一比率并未满足要求，尽管如此，算法的不确定性通常是可以接受的。因此，它只能作为控制器的一种辅助决策方法。此外，虽然测试结果表明，与之前的方法相比，动作选择更符合控制习惯，但在实际区域控制中，不同扇区的控制器优先解决动作是不同的。冲突解决动作往往取决于控制员的个人习惯和机组人员的偏好。同时，还应考虑飞机的相对位置、航线结构和空域情况。为了满足空中交通管制员的实际解决习惯，需要对一个扇区或一个控制器进行实证统计，然后修改奖励函数，使训练结果符合他们的解决习惯。总的来说，使用 DRL 框架和独立学习解决多架飞机飞行冲突问题是成功的。通过与基本算法的比较，证明了所提出的算法在解决效率上具有很大优势。它可以确保未来高密度空域的安全和高效。

7. Conclusions

7. 结论

(1) A method for multi-aircraft flight conflict resolution based on DRL is proposed in this paper, which combines the DQN algorithm in DRL and an independent learning framework to construct an IDQN algorithm. Through verification, this method has great advantages in solving time

(1) 本文提出了一种基于深度强化学习 (DRL) 的多飞机飞行冲突解决方法，该方法结合了 DRL 中的 DQN 算法和独立学习框架，构建了 IDQN 算法。通过验证，这种方法在解决时间上具有很大优势。

(2) An MDP-based multi-aircraft flight conflict resolution model is established, and IDQN based on MLP neural network is used to solve the problem, and a 'downward-compatible' framework is proposed to support the dynamic change of the number of conflicting aircraft in conflict scenarios.

(2) 建立了基于马尔可夫决策过程 (MDP) 的多机飞行冲突解决模型，并使用基于多层感知器 (MLP) 神经网络的 IDQN 算法来解决问题，提出了一种"向下兼容"的框架，以支持冲突场景中冲突飞机数量的动态变化。

(3) By using 700 multi-aircraft flight conflict scenarios constructed in ATOSS for testing, the successful resolution rate of the model can reach 85.71% , and 71.51% of the aircraft can reach their destinations within 150 s around original arrival times. At the same time, through 1000 tests, the successful conflict resolution rate of the algorithm is distributed between 70.00% and 88.57% , and it's generally a normal distribution. Next, the research can be carried out from the following aspects: extract the control experience from historical radar data to train the reward function to make it more in line with the actual operation, further improve the successful resolution rate of the model and reduce the probability of algorithm failure, add obstacle areas such as restricted areas and thunderstorm avoidance areas so that the aircraft can avoid obstacles.

(3) 通过使用在 ATOSS 中构建的 700 个多机飞行冲突场景进行测试，该模型的成功解决率可以达到 85.71% ，并且 71.51% 的飞机可以在原预计到达时间左右 150 s 内到达目的地。同时，通过 1000 次测试，算法的成功冲突解决率分布在 70.00% 和 88.57% 之间，通常呈正态分布。接下来，研究可以从以下几个方面进行: 从历史雷达数据中提取控制经验来训练奖励函数，使其更符合实际操作，进一步提高模型的成功解决率并降低算法失败的概率，添加限制区域和雷暴规避区域等障碍区域，以便飞机能够避开障碍。

Declaration of Competing Interest

竞争利益声明

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

作者声明，他们没有已知的经济利益或个人关系可能影响本文报告的工作。

Acknowledgements

致谢

This paper is jointly completed by the project team of the Intelligent Air Traffic Control Laboratory of NUAA and cannot be separated from the efforts of every member. And this work was supported by Safety Ability Project of Civil Aviation Administration of China (No. TM 2018-5-1/2), and the Open Foundation project of The Graduate Student Innovation Base, China (Laboratory) of Nanjing University of Aeronautics and Astronautics, China (No. kfjj20190720). At the same time, I would like to thank the other teachers of the laboratory where the project is located for their support and guidance, as well as the technical guidance of professors from the College of Computer Science and the advice of experts in air traffic control.

本文由南京航空航天大学智能空中交通控制实验室项目团队共同完成，离不开每一位成员的努力。该项工作得到了中国民用航空局安全能力项目 (编号:TM 2018-5-1/2) 的支持，以及南京航空航天大学研究生创新基地 (中国)(编号:kfjj20190720) 开放基金项目的资助。同时，感谢实验室的其他教师给予的支持和指导，以及计算机科学与技术学院教授的技术指导和空中交通控制领域专家的建议。

Appendix A. Supplementary data

附录 A. 补充数据

Supplementary data to this article can be found online at

本文的补充数据可以在以下网址在线找到:https://doi.org/10.1016/j.cja.2021.03.015.

References

参考文献

1. Civil Aviation Administration of China. 2019 Civil aviation industry development statistical bulletin. Beijing: Civil Aviation Administration of China; 2020.

2. Guan X, Lyu R, Shi H, et al. A survey of safety separation management and collision avoidance approaches of civil UAS operating in integration national airspace system. Chin J Aeronaut 2020;33(11):2851-63.

3. Durand N, Alliot J, Noailles J. Automatic aircraft conflict resolution using genetic algorithms. SAC '96-ACM symposium on applied computing; 1996. p. 289-98.

4. Durand N. Neural nets trained by genetic algorithms for collision avoidance. Appl Intell 2000;13(3):205-13.

5. Stephane M, Conway S. An airborne conflict resolution approach using a genetic. AIAA guidance, navigation, and control conference and exhibit; Montreal, Canada. Reston: AIAA; 2001. p. 1-22.

6. Ma Y, Ni Y, Liu P. Aircrafts conflict resolution method based on ADS-B and genetic algorithm. 2013 6th international symposium on computational intelligence and design; Hangzhou, China. Piscataway: IEEE Press; 2013. p. 121-4.

7. Guan X, Zhang X, Han D, et al. A strategic flight conflict avoidance approach based on a memetic algorithm. Chin *J* Aeronaut 2014;27(1):93-101.

8. Emami H, Derakhshan F. Multi-agent based solution for free flight conflict detection and resolution using particle swarm optimization algorithm. UPB Sci Bull, Ser C: Electr Eng 2014;76 (3):49–64.

9. Zhou J, Rahmani A, Liu X, et al. Application of distributed MAS in flight conflict avoidance. J Transp Syst Eng Inf Technol 2015;15:231-8 [Chinese].

10. Liu Y, Zhang X, Zhang Y, et al. Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach. Chin J Aeronaut 2019;32(6):1504-19.

11. Bicchi A, Marigo A, Pappas G, et al. Decentralized air traffic management systems: performance and fault tolerance. IFAC Proc Vol 1998;31(27):259-64.

12. Menon PK, Sweriduk GD, Sridhar B. Optimal strategies for free-flight air traffic conflict resolution. J Guid Ccontrol Dynam 1999;22 (2):202–11.

13. Ghosh R, Tomlin C. Maneuver design for multiple aircraft conflict resolution. Proceedings of the 2000 American control conference; Chicago, IL, USA. Piscataway: IEEE; 2000. p. 672-6.

14. Narkawicz A, Muñoz C, Dowek G. Provably correct conflict prevention bands algorithms. Sci Comput Program 2012;77(10- 11):1039-57.

15. Hu JH, Lygeros J, Prandini M, et al. Aircraft conflict prediction and resolution using brownian motion. Proceedings of the 38th IEEE conference on decision and control; Phoenix, AZ, USA. Piscataway: IEEE; 1999. p. 2438-43.

16. Liu XF, Barnes EC, Savolainen JE. Conflict detection and resolution for product line design in a collaborative decision making environment. Conference on computer supported cooperative work, 2012 Feb; New York, NY, USA. New York: ACM Press; 2012. p. 1327-36.

17. Han YX, Tang XM, Han SC. Conflict resolution model of optimal flight for fixation airway. J Traff Transp Eng 2012;12(01):115-20 [Chinese].

18. Tang XM, Han YX, Han SC. 4D trajectory based operation flight conflict supervisory control based on hybrid system theory. J Univ Electron Sci Technol China 2012;41(5):717-22 [Chinese].

19. Li Y, Du W, Yang P, et al. A satisficing conflict resolution approach for multiple UAVs. IEEE Internet Things 2019;6 (2):1866-78.

20. Xurui J, Minggong W, Xiangxi W, et al. A multi-aircraft conflict resolution method based on cooperative game. 2017 IEEE international conference on cybernetics and intelligent systems (CIS) and IEEE conference on robotics, automation and mecha-tronics (RAM); Ningbo, China. Piscataway: IEEE; 2017. p. 774-8.

21. Pappas GJ, Tomlin C, Sastry SS. Conflict resolution for multi-agent hybrid systems. Proceedings of 35th IEEE Conference on Decision and Control. Kobe, Japan. Piscataway: IEEE; 1996. p. 1184-9.

22. Tang XM, Chen P, Li B. Receding horizon optimization of en route flight conflict resolution strategy. J Traff Transp Eng 2016;16 (05):74-82 [Chinese].

23. Soler M, Kamgarpour M, Lloret J, et al. A hybrid optimal control approach to fuel-efficient aircraft conflict avoidance. IEEE Trans Intell Transp 2016;17(7):1826-38.

24. Wang Z, Li H, Wang J, et al. Deep reinforcement learning based conflict detection and resolution in air traffic control. IET Intell Transp Syst 2019;13(6):1041-7.

25. Pham D, Tran NP, Goh SK, et al. Reinforcement learning for two-aircraft conflict resolution in the presence of uncertainty. IEEE-RIVF international conference on computing and communication technologies; Danang, Vietnam. Piscataway: IEEE Press; 2019. p. 1-6.

26. Tran NP, Pham D, Goh SK, et al. An intelligent interactive conflict solver incorporating air traffic controllers' preferences using reinforcement learning. Integrated communications, navigation and surveillance conference; Herndon, VA, USA. Piscataway: IEEE; 2019. p. 1-8.

27. Wang Z, Li H, Wang J, et al. Deep reinforcement learning based conflict detection and resolution in air traffic control. IET Intell Transp Sy 2019;13(6):1041-7.

28. Temizer S, Kochenderfer M, Kaelbling L, et al. Collision avoidance for unmanned aircraft using Markov decision processes*. AIAA guidance, navigation, and control conference. Reston: AIAA; 2010.

29. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature 2015;518 (7540):529-33.

30. Tampuu A, Matiisen T, Kodelja D, et al. Multi-agent cooperation and competition with deep reinforcement learning. PLoS ONE 2015;12.

31. Li Y. Research on the ATC conflict identification and resolution based on machine learning [dissertation]. Nanjing: Nanjing University of Aeronautics and Astronautics; 2019 [Chinese].