ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2025, Vol. 86, No. 9, pp. 845–866.

REVIEWS

An Overview of Conflict Prevention Methods in Air Traffic Control Using Deep Reinforcement Learning

E. L. Kulida[*,a] **and V. G. Lebedev**[*,b]

[*] *Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
e-mail: [a] *elena-kulida@yandex.ru,* [b] *lebedev-valentin@yandex.ru*

Received March 4, 2025
Revised March 17, 2025
Accepted March 25, 2025

Abstract—An overview of the development of modern approaches to conflict prevention between aircraft based on deep reinforcement learning is given. The basic concept of reinforcement learning and some fundamental algorithms used for aircraft conflict prevention are reviewed. Models with discrete and continuous actions for conflict prevention in two-dimensional and three-dimensional airspaces during flight along fixed trajectories or in free flight are presented. Various approaches to representing information about the state of the airspace (using a state vector and as a graph) and different types of interaction between aircraft (based on information about the state of surrounding aircraft or through message exchange) are considered.

Keywords: air traffic management, conflict prevention, aircraft maneuver, deep reinforcement learning

DOI: 10.31857/S0005117925090041

1. INTRODUCTION

Maintaining a safe separation distance, both vertically and horizontally, between any two aircraft at any time is the most important function of an air traffic control system [1]. Loss of separation between aircraft is called a conflict. The growing density of air traffic leads to an increase in the number of potential conflicts, so prevention methods are important to reduce the risk of collisions. Strategic prevention of potential conflicts based on the global optimization model that generates conflict-free four-dimensional trajectories for all aircraft in advance fails to manage uncertainty arising in the dynamics of flights in real time [2]. Tactical real-time prevention of potential conflicts is critical for ensuring safe air traffic control since it allows for better management of uncertainty that arises in flight dynamics.

At present, aircraft mainly move along fixed trajectories, with conflict prevention being the responsibility of air traffic controllers. Tactical decisions are still made by air traffic controllers with very few changes as compared to decisions made 50 years ago [3]. With the increasing intensity of air traffic, the workload of air traffic controllers is constantly growing and may exceed human capabilities. The development of existing methods for assessing the dynamic air situation by an air traffic controller in order to reduce risk factors, including information overload and lack of time for functional operations, is considered in [4]. In [5–7], a conception is proposed for the development of automation tools to increase the capacity and safety of airspace operation in order to effectively and intelligently support air traffic controllers' decision-making to prevent conflicts. The prospective air traffic management involves using free flights when aircraft will move along arbitrary trajectories and conflict prevention will be ensured by an autonomous air traffic control

system. Theoretical studies confirm that free flight can potentially increase safety [8] and reduce fuel consumption [9]. To implement the free flight concept, one needs to "ensure control over aircraft separation using on-board systems in addition to ground systems. The higher reliability of such a structure will make it possible to clarify a number of scenarios for simulating the risk of aircraft collision and will contribute to the creation of safer, more flexible, and more capacious conditions for air traffic management" [10]. The challenge to develop decentralized autonomous conflict prevention tools is of fundamental importance for the implementation of the free flight concept [11]. Methods and algorithms for detecting and preventing dangerous approaches in the air within the framework of a promising air traffic management system, taking into account flight safety and efficiency requirements, were studied in [12, 13].

Methods have been developed to provide air traffic controllers with recommendations on conflict resolution based on optimal control [14] and mathematical programming [15–18], geometric optimization [19–21], evolutionary algorithms [22], and the Monte Carlo tree search algorithm [23]. These methods work with the existing air traffic density; however, when it comes to higher density, their insufficient computational efficiency becomes an issue. It takes them tens or even hundreds of seconds to come up with a decision. Taking into account the uncertainty inherent in air traffic significantly increases the computation time, making the methods less capable of quickly generating decisions. Most conventional approaches to ensuring separation fail to cope with stochastic environments and high air traffic density [24]. New approaches are needed that can effectively respond to the dynamics of the external environment in real time, for example, based on neural networks and machine learning [25–27].

Recently, deep reinforcement learning has been widely used in various fields of aviation since it can solve decision-making problems unavailable previously due to a combination of nonlinearity and high dimensionality [28]. The use of deep reinforcement learning will automatically provide safe and effective conflict prevention decisions to support air traffic controllers' decision-making and reduce their workload [29]. In the future, a fully automated control system will become the ultimate solution for handling high-density, complicated, and dynamic air traffic [30].

Reinforcement learning methods are applied in two stages, viz. the stage of training the model and the stage of applying the trained model in practice. While it can take long to train a model, the trained model helps generate decisions very quickly. The decision-making speed is an indicator of the efficiency and advantage of deep reinforcement learning as compared to conventional algorithms. Faster decision-making means earlier detection of conflicts and formulation of instructions to reduce the workload of air traffic controllers and pilots. In [31], the following figures are given (with the proviso that the data were obtained under different computation conditions)—it takes a mixed-integer linear programming algorithm 49 seconds to generate a decision, the average decision time using a genetic algorithm is 37.6 seconds, and a trained deep reinforcement learning agent requires less than 0.2 seconds. Reinforcement learning methods have a clear advantage over conventional methods in the speed of computing decisions and the ability to adapt to dynamics of the external environment, which is critically important when resolving air traffic conflicts.

Research on conflict resolution in air traffic using deep reinforcement learning has been continuously conducted since 2018, with many models and algorithms proposed during this period [31]. Conflict resolution models for both air route movement and free flight as well as models with both discrete and continuous actions are considered. At present, in most cases, two-dimensional models are proposed for conflict resolution using horizontal maneuvers, three-dimensional models for horizontal and vertical maneuvers being proposed much less frequently [32]. Two-dimensional models ignore vertical maneuvers due to the potential instability they can cause in air traffic [33]. The number of conflicting aircraft varies—conflicts between two aircraft and conflicts in groups of aircraft with a fixed or variable number of aircraft are considered. In [34–36], a hybrid approach

has been developed that combines the strengths of geometric and reinforcement learning methods to resolve conflicts. It is argued that the wide range of different optimal decisions found by the reinforcement learning method shows that the rules of the geometric method should be expanded to take into account different conflict geometries. In Russia, the problem of aircraft traffic control based on reinforcement learning is currently being actively studied by State Scientific Research Institute of Aviation Systems (GosNIIAS) experts, with a significant scientific and technical reserve already made [37].

Given a wide variety of conflict resolution problem statements, when constructing a reinforcement learning model, in many cases, one first develops an interactive environment for an agent to learn various strategies. An artificial intelligence agent learns using deep reinforcement learning algorithms through trial and error, applying various possible actions and receiving feedback from the environment in the form of rewards. The agent's goal is to study an action selection strategy that will maximize the mathematical expectation of the total discounted rewards over a long period of time. The convergence of the reinforcement learning model to the desired outcome is determined by the choice of the reward function, through which the agent learns to optimize the strategy for choosing actions in various situations. The reward function has an impact on the learning rate, convergence, and performance of agents. The principal thing that should be considered in the reward function is the success or failure of conflict resolution. Moreover, to increase the efficiency of the model, the reward should take into account the number of maneuvers and the time required to resolve the conflict. Using the reward function, it is possible to bring the agent's behavior closer to the existing rules for conflict resolution by air traffic controllers [38, 39].

Despite the success of reinforcement learning in research on solving air traffic control problems, two significant challenges remain for this method to be applied in real conditions. The first problem is the vulnerability of deep neural networks to adversarial attacks, the second is the problem of the explainability of "black box" models, viz. pilots and air traffic controllers do not understand how the models make certain decisions [40, 41]. In [42], approaches to resolving these issues in the autonomous resolution of aircraft conflicts are proposed.

2. REINFORCEMENT LEARNING

2.1. Basic Conception [43]

Machine learning is divided into supervised learning, unsupervised learning, and reinforcement learning. Supervised learning allows approximating any function; however, it requires labeled data sets to be available, which is not always so. Unsupervised learning relies on sets of unlabeled data. Reinforcement learning allows implementing a consistent decision-making process through trial and error, with the learning data synthesized as the agent interacts with the environment. When using reinforcement learning, each aircraft is simulated as an interactive agent whose actions are conflict prevention maneuvers.

Reinforcement learning is based on the Markov model of the decision-making process, in which the state of the system and the actions of the agent do not depend on how the system came to this state. The Bellman equations hold for the Markov decision-making process. The basis of the Markov model of the decision-making process is the environment and the agent operating in it. The environment is characterized by a set of parameters, and the state of the environment s is a specific set of values of these parameters. An agent is a program that can analyze the state of the environment and perform a specific set of actions $a \in A(s)$ in each state. As a result of the agent's action, the environment switches from the state s to the new state s' and receives feedback from the environment in the form of the reward $r = R(s, a, s')$. In the case of multistep interaction of the agent with the environment, starting from the state s at the step t to the end of the episode at

the step T, the total discounted benefit

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots + \gamma^{T-1} r_T = r_{t+1} + \gamma G_{t+1}$$

is determined, and $\gamma \in [0, 1]$ is the discount coefficient, which specifies the decrease of the value of the reward received at later steps.

The transition function $p(s'|s, a)$ specifies the probability of transition to the state s' at the step t given that the action a was chosen in the state s at the step $t - 1$

$$p(s'|s, a) = P(S_t = s'|S_{t-1} = s, A_{t-1} = a), \quad \sum_{s' \in S} p(s'|s, a) = 1, \quad \forall s \in S, \quad \forall a \in A(s).$$

A strategy (or a policy) is a function $\pi(a|s)$ that matches the action of the agent with each non-terminal state of the environment.

The expected benefit when an agent follows the strategy π in the state s is called the state value function

$$V_\pi(s) = E_\pi[r_{t+1} + \gamma G_{t+1}|S_t = s].$$

However, the state value function does not allow us to know the expected benefit from the agent performing the action a in the state s when it follows the strategy π; it is specified by the action value function

$$Q_\pi(s, a) = E_\pi[r_{t+1} + \gamma G_{t+1}|S_t = s, A_t = a].$$

The basic concept of reinforcement learning—generalized iteration with respect to strategies—is an iterative procedure. The step of this procedure involves two processes, viz. evaluating the current strategy to refine the current approximation of the value function followed by improving the strategy in accordance with the changed value function.

The value function can be represented as

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma V_\pi(s')], \forall s \in S.$$

The iteration step to refine the value function is to calculate it for the action with the highest value

$$V_{k+1}(s) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V_k(s')].$$

The action that delivers the maximum value to the Q-function is called greedy. The strategy can be improved by using a strategy optimization algorithm, which consists in choosing a greedy action with respect to the Q-function

$$\pi'(s) = \operatorname*{argmax}_a \sum_{s',r} p(s', r|s, a)[r + \gamma V_\pi(s')].$$

When studying the value function, it is very important to maintain a balance between choosing a greedy action and choosing a random action for exploration. There are many different approaches to solving this problem such as the epsilon-greedy strategy with a random action selected with the epsilon probability, the epsilon greedy strategy with decay with epsilon decreased as the agent learns, action selection strategies using knowledge gained up to the current step about the value and exploration degree of actions, etc.

The two processes considered stabilize when the value function matches the strategy, and the strategy is greedy with respect to the value function. The strategy and the value function are optimal if

$$V^*(s) = \max_\pi V_\pi(s).$$

2.2. Reinforcement Deep Learning Algorithms [44, 45]

For simple examples, the action value function is represented as a table. For important practical tasks, it is impossible to implement a table representation of value functions due to the large number

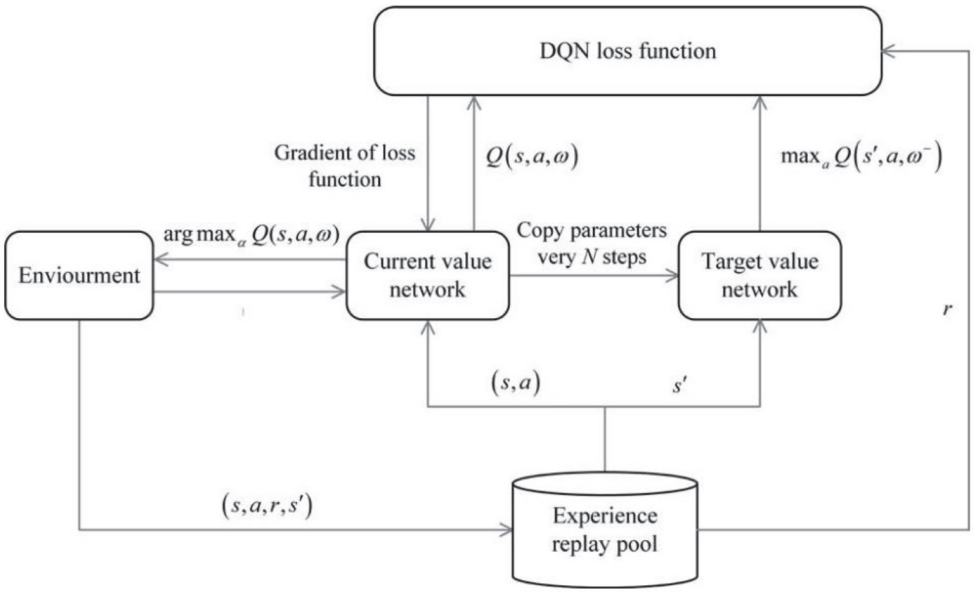

Fig. 1. The operation principle of the deep Q-learning algorithm.

of states, continuous variables or actions. This is the case when deep reinforcement learning is applied with deep learning algorithms with a teacher used to approximate value functions based on the samples (s, a, r, s') formed as the agent interacts with the environment.

At present, this approach is being rapidly developed in the field of aviation conflict prevention, and various algorithms for the value function approximation and strategy optimization are being proposed and studied. The deep Q-learning (deep Q-network, DQN) algorithm is widely used, which leverages two key technologies, viz. experience replay and the dual network structure. Figure 1 shows the operation principle of the DQN algorithm [46].

The experience replay consists in creating the replay buffer D that accumulates a large number of samples. The mini-sets $U(D)$ for network training are selected from the accumulated replay buffer uniformly and randomly and thus correspond to different trajectories and policies, increasing the network training stability. The dual network structure is based on using the same network with different sets of parameters. A dynamic network is used to approximate the current value $Q(s, a; \omega_i)$, and the parameters of this network ω_i are updated at each time step i. The target network is used to obtain a more stable target value Q, and the parameters of the target network ω^- are updated in N time steps. The loss function has the form

$$L_i(\omega_i) = E_{(s,a,r,s')\sim U(D)}\left[\left(r + \gamma \max_{a'} Q(s', a'; \omega^-) - Q(s, a; \omega_i)\right)\right].$$

The loss function is optimized by the gradient descent method

$$\nabla_{\omega_i} L_i(\omega_i) = E_{(s,a,r,s')\sim U(D)}\left[\left(r + \gamma \max_{a'} Q(s', a'; \omega^-) - Q(s, a; \omega_i)\right) \nabla_{\omega_i} Q(s, a; \omega_i)\right].$$

The difference between the double deep Q-network (DDQN) method [47] and the DQN method is that the dynamic network parameters are used instead of the target network parameters to select an action in the equation.

$$\nabla_{\omega_i} L_i(\omega_i) = E_{(s,a,r,s')\sim U(D)}\left[\left(r + \gamma Q(s', \operatorname*{argmax}_{a'} Q(s', a'; \omega_i); \omega^-) - Q(s, a; \omega_i)\right) \nabla_{\omega_i} Q(s, a; \omega_i)\right].$$

The DQN and DDQN algorithms are used in conflict prevention models with a discrete action space [48–52].

Methods that use a continuous action space to resolve conflicts between aircraft are also proposed [34, 53–55]. In this case, strategies are represented by parameterized stochastic functions $\pi_\theta(a, s)$ that are optimized using actor-critic algorithms. In these algorithms, in addition to the neural network used to evaluate the strategy (the critic), a second neural network is used to form a strategy based on optimizing the value function (the actor).

For the trajectory $\tau = S_0, A_0, R_1, S_1, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$, the function $G(\tau)$ is the total discounted benefit, and $\pi_\theta(A_t|S_t)$ is the probability of choosing the action A_t in the state S_t at the step t. The actor's network is updated according to the gradient of the value function

$$\nabla_\theta E_{\tau\sim\pi_\theta}[G(\tau)] = E_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} G_t(\tau)\nabla_\theta \log \pi_\theta(A_t|S_t)\right].$$

3. AIRCRAFT CONFLICT PREVENTION MODELS

3.1. Conflict Prevention Models for Two Aircraft

A conflict resolution strategy can include both two-dimensional (2D) maneuvers such as changing the course and speed in a plane airspace and three-dimensional (3D) maneuvers when there is also a change in altitude. Although a 2D model is not as effective in conflict resolution as a 3D model is, maneuvers in the two-dimensional airspace cause less discomfort for passengers and do not distort the vertically stratified structure of the airspace [56]. If they are too large, neural networks can be harmful to agent training due to an excessively big number of parameters. Therefore, smaller neural networks for 2D models have greater potential for the model to be expanded further to take into account more real-world factors [30].

The first studies on aircraft conflict prevention using reinforcement learning considered conflict resolution between two aircraft in a two-dimensional airspace. In one of the first works on aircraft conflict prevention, flights along routes were considered, with a hierarchical structure of deep reinforcement learning proposed [48]. The training environment used was the NASA Sector 33 software containing 35 air traffic control problems involving two to five aircraft. The hierarchical structure includes a parent agent designed to solve the problem of choosing aircraft routes at the beginning of the episode and the child agent that controls actions to change speeds on the selected routes. The hierarchical structure allows separating the route selection actions performed at the beginning of the episode and the speed control actions during the episode. A reinforcement learning algorithm based on a dual deep Q-network is used to train the agents. The first neural network (the target one) is used to select actions that are greedy with respect to the current Q-function, and the second neural network (the dynamic one) is used to adjust the Q-function based on evaluating the success of the actions performed. It was shown in [48] that a hierarchical deep reinforcement learning agent can choose optimal combinations of routes and speeds in order to avoid a conflict between two aircraft flying along routes.

One of the first works on reinforcement learning for conflict resolution in free flight deals with the case of two aircraft, with uncertainty taken into account [53]. An environment has been developed for simulating potential conflicts for agent training and testing. Figure 2 shows a conflict between two aircraft in a circular airspace with the radius R of 50 nautical miles and a maneuver to prevent it. The trajectory of the own aircraft is A_1B_1, the trajectory of the intruder is A_2B_2, and QP is the closest distance between the two aircraft at which they lose safe separation if none of them makes a maneuver. A single maneuver to change the course in the continuous two-dimensional space is used as a conflict prevention action. The maneuver A_1MNB_1 in Fig. 2 represents a series of actions performed by the own aircraft—deviating from the original trajectory at the point M by changing the course by the angle α, then moving along the vector MN, and turning to the point B_1 at the point N.

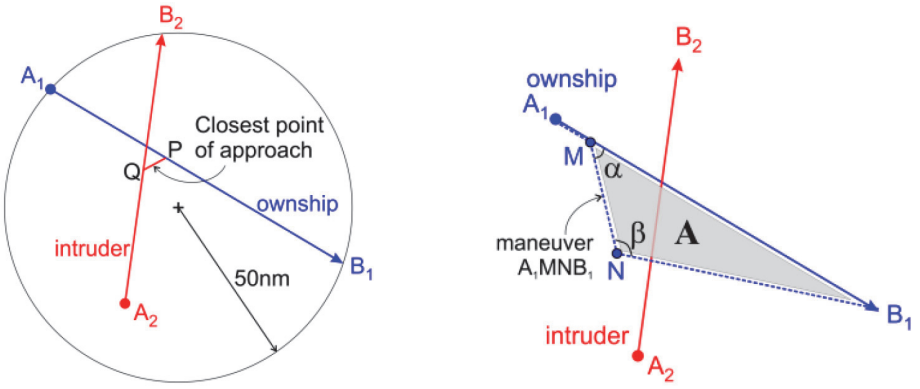

Fig. 2. The scenario of a conflict between two aircraft and a maneuver to prevent it.

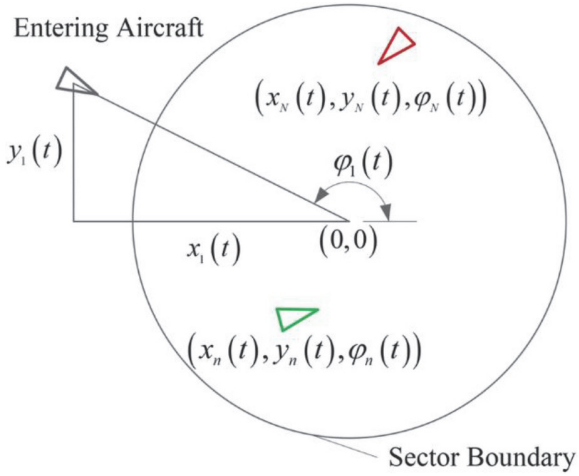

Fig. 3. Conflict scenario with several aircraft.

The reward of the agent being trained is calculated as the sum

$$R_{final} = 5 + R_{conflict} + R_{maneuver} + R_{deviation},$$

i.e., negative rewards are added to reward 5, viz. $R_{conflict} = -3$ if the maneuver fails to resolve the conflict, $R_{maneuver} = -2$ if it goes beyond the boundaries of the area or if the angle $\beta > 120°$, $R_{deviation} = -S_{\Delta MNB_1}$, the deviation from the initial trajectory is estimated by the area between the trajectory of the maneuver and the initial trajectory.

The deep deterministic policy gradient (DDPG) method, which is one of the advanced methods of deep reinforcement learning for continuous action space control problems, is used [44]. The algorithm uses two neural networks, viz. the critic's network to study the utility function of state-action pairs $Q(s, a)$ and the actor's network to map the state into a deterministic action based on the policy gradient. The performance of the DDPG algorithm for preventing air traffic conflicts is close to the performance of conventional methods, yet the calculation time is significantly reduced [57].

In [54, 56, 58, 59], the conflict prevention problem statement is generalized to the case when there are several other aircraft in the area in addition to the own aircraft and the intruder. In this case, secondary potential conflicts (the domino effect) may arise when maneuvering to prevent the conflict [60].

In [54], a two-dimensional environment for simulating free flights was developed, which can be applied to several aircraft in a sector (no more than 5) (Fig. 3).

It is assumed that there are no conflicts between these aircraft, and conflicts can be caused by an aircraft entering the sector that aim to fly from the starting point to the end one in minimal time without collisions with other aircraft. An actor-critic algorithm with the fixed number K of actions (control cycles) is proposed. To avoid conflicts, actions are generated to change the heading angle of the entering aircraft. At each time step, the agent selects the action $A = \{\rho, \varphi | \rho \in [0, L], \varphi \in [-\pi, \pi]\}$ described by a two-dimensional polar coordinate, where ρ, φ are the polar radius and the angle.

The reward function

$$R_t = \begin{cases} -1 & \text{if there is a conflict,} \\ 1 - \dfrac{1}{K} \times \dfrac{|\Delta\varphi_t|}{\pi}, & \text{otherwise.} \end{cases}$$

The value function is approximated using the neural network $\hat{V}(S_t, \omega) \approx V_\pi(S)$, where ω are the weight of neurons.

For the critic's network, δ is specified

$$\delta_t = R_t + \gamma \hat{V}(S_{t+1}, \omega) - \hat{V}(S_t, \omega),$$

where R_t is the immediate reward, $\hat{V}(S_{t+1}, \omega)$ is the value of the value function of the next state, and $\hat{V}(S_t, \omega)$ is the value of the value function of the current state. The least squares method is used to update the parameters $\omega \leftarrow \omega + \alpha^\omega \nabla \delta^2$; α is the learning rate.

The policy gradient method is used for the actor's network. The equation

$$\ln \pi(\rho_t, \varphi_t | S_t, \theta) = \ln \pi(\rho_t | S_t, \theta) + \ln \pi(\varphi_t | S_t, \theta)$$

is used for the action (ρ, φ), where $\pi(\rho_t, \varphi_t | S_t, \theta)$ is the probability of choosing ρ and φ in the state S_t with the parameters θ, $\pi(\rho_t | S_t, \theta)$ is the probability of choosing ρ in the state S_t with the parameters θ, and $\pi(\varphi_t | S_t, \theta)$ is the probability of choosing φ in the state S_t with the parameters θ. The parameters θ are updated by the formula

$$\theta \leftarrow \theta + \alpha^\theta \delta_t \nabla \ln \pi(\rho_t, \varphi_t | S_t, \theta).$$

The variable number of control cycles is not reflected in the reward function, and it cannot be adjusted dynamically. Since the computational efficiency of a well-trained agent is very high, several agents with different values K, for example $K = 1, 2, 3$, are proposed to be trained. In the real-world air traffic control process, one can quickly calculate several control options and select the best decision from several options. The simulation results confirmed that deep reinforcement learning can be used for conflict resolution and has advantage in computational efficiency as compared to known methods.

In [61], a model for two-aircraft conflict resolution was proposed and analyzed, taking into account wind-related uncertainty. The proposed conflict prevention method is applicable in the case when uncertainty associated with the direction and speed of the wind is unstable (stochastic) throughout the simulation.

In [46], a strategy for resolving two-aircraft conflicts in the three-dimensional space based on deep reinforcement learning is considered. One conflict between two aircraft is selected for resolution from an air traffic scenario that may contain several conflicts. The conflict resolution model is simulated as a discrete-time Markov decision-making process. The agent uses altitude adjustment, speed adjustment, or course correction commands to resolve the conflict. The preferences of the air traffic controller for choosing conflict resolution maneuvers are transmitted to the agent by adjusting the reward function.

3.2. Conflict Prevention Models for a Fixed Number of Aircraft

Multi-agent reinforcement learning considers a set of agents that interact with the same environment [62]. Each agent tries to achieve its goals, which are unknown to other agents. One of the strategies for solving problems in a multi-agent environment is represented by independent Q-learning with no communication between agents and other agents considered as part of the environment [63]. However, when an agent changes its policy, it affects the policies of other agents, resulting in training instability [64].

In [3, 65, 66], to ensure communication between agents, the state for each agent is proposed to include information about the state of N closest agents. The state space for an agent has a constant dimension since it depends only on N closest agents and does not scale to fit the increased number of agents in the environment. At the same time, it is highlighted that determining which N closest agents should be considered is very important for a good result to be obtained since adding irrelevant information to the state space complicates training [3].

In [67], a reinforcement learning algorithm combined with the Monte Carlo tree search (MCTS-UCT) is presented to solve the problem of self-maintenance of aircraft safe separation given high air traffic flows in the sector. All agents (aircraft) are at the same flight level, and their strategies include changing the course and the cruising speed.

The value of the reward for an agent, depending on the state s, is determined as

$$r(s) = \begin{cases} 1 & \text{if } s \text{ is the target state,} \\ 0 & \text{if } s \text{ is } LOS \text{ or out of the sector boundary,} \\ 1 - \dfrac{d(o,g)}{\max d(o,g)}, & \text{otherwise,} \end{cases}$$

where the state LOS is the loss of separation, $d(o,g)$ is the distance between the agent's current position and its target, and $\max d(o,g)$ is the greatest distance between the agent and its target.

Each agent state is considered as a node in the tree, and tree deployment is performed based on the state values calculated using the formula

$$UCT(S_j) = \bar{r}_j + 2C\sqrt{\frac{2\ln N}{n_j}},$$

where $\bar{r}_j$ is the average value of the reward of the action j for the current agent, N is the counter of node visits, n_j is the action selection counter j, and $C = 1\Big/\sqrt{2}$.

In the process of joint decision-making, all n agents $\{A_1, \ldots, A_n\}$ must share their intention when choosing each individual action. One iteration of the algorithm of the multi-agent Markov decision-making process is as follows. First, n agents $\{A_1, \ldots, A_n\}$ are initialized at the level $L-1$. All agents continue to follow the default cooperative action policy $a_{-j} = \{a_i$ from the default cooperative action policy $|i = 1, \ldots, n, i \neq j\}$. The agent A_j with the minimum index at the level $L-1$ uses the MCTS-UCT algorithm to select its optimal strategy of actions a_j^* using the following equation

$$a_j^* = \underset{a_j}{\operatorname{argmax}}\, r_j^*(s, a_j, a_{-j}), \quad j = 1, \ldots, n,$$

where $r_j^*(s, a_j, a_{-j})$ is the value of the reward of the agent A_j in the state s when performing the action a_j while the action strategies of other agents are represented as a_{-j}. When calculating a_j^*, other agents will continue to follow the default action strategy set as $a_{-j} = \{a_i$ from the default cooperative action policy $|i = 1, \ldots, n, i \neq j\}$. When the agent A_j receives its optimal action strategy a_j^*, A_j is upgraded to the level L and preserves the action strategy a_j^* to update the default

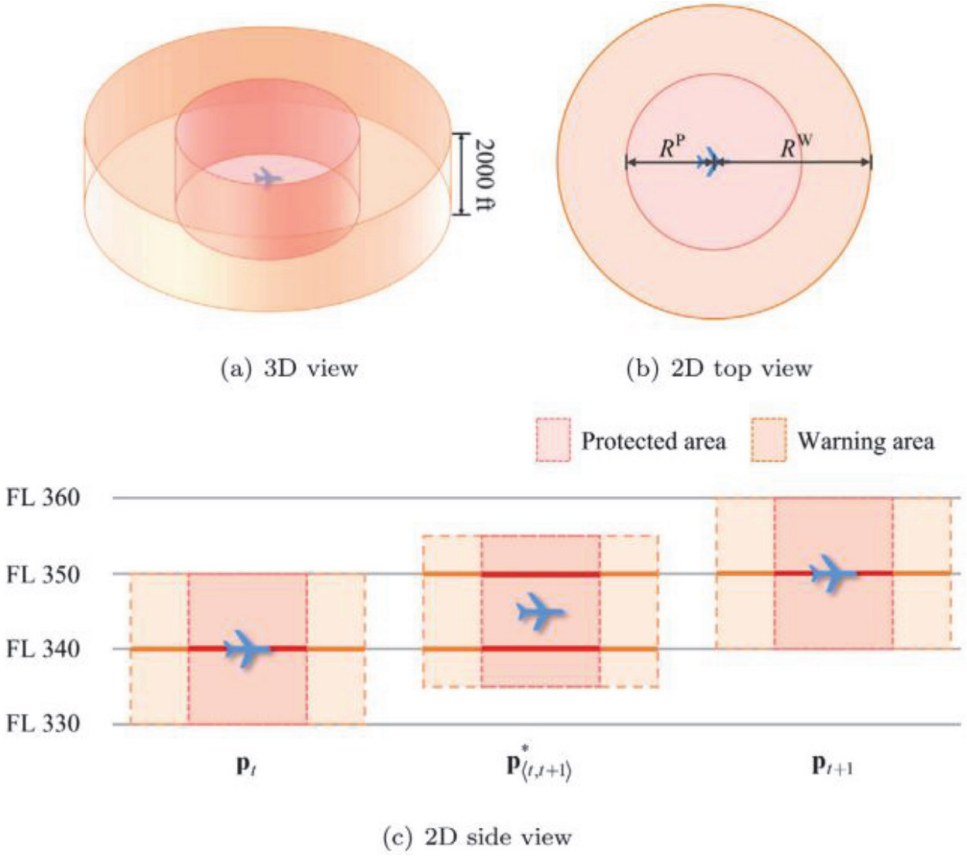

Fig. 4. Conflict buffer model.

joint decision-making strategy. Then, the next agent, with the minimum index at the level $L-1$ chooses its optimal action strategy. This process continues until all agents have their optimal action strategy $\{a_i^*, i = 1, \ldots, n\}$. The resulting joint decision-making strategies are used for the next time step Δt for all agents. Iterations are repeated until all agents reach the target state.

In [68], a method of deep ensemble multi-agent reinforcement learning was proposed for dynamic adjustment of the aircraft speed in real time. The authors claim that the extensive empirical results obtained using an open-source air traffic control model developed by Eurocontrol and based on real-world data involving thousands of aircraft demonstrate that the proposed method is significantly superior to other reference approaches.

In [69], an approach to multi-agent reinforcement learning is proposed for three-dimensional conflict resolution in a free route space, where agents use a common neural network. The trained network is deployed on each aircraft to form a distributed real-time decision-making system. In this case, communication between agents is reduced to informing other agents about the selected actions. The introduction of the three-dimensional space leads to an explosive increase in the scale of the neural network and, as a result, to an increase in the training complexity. To overcome this problem, it is proposed to consider conflicts in three planes rather than the three-dimensional space in the case of three-dimensional maneuvers, viz. the plane of the current flight level and two adjacent planes at the levels above and below it. This significantly reduces the agent training complexity. A conflict buffer model is proposed in which each aircraft is assigned a protective zone and a warning zone. Figure 4 [69] illustrates the conflict buffer model, where R^W and R^P stand for the radii of the warning zone and the protective zone, respectively. Intruders found inside the protective zone always receive a big fine while intruders found in the warning zone receive a small fine.

The number of conflicting aircraft that an agent can monitor is fixed in the neural network structure and should not change. To solve this problem, a partial surveillance model has been developed, which considers only a fixed number of aircraft that pose the greatest threat, for example, those closest to the agent. The DQN algorithm with improvements, called Rainbow, is used to implement reinforcement learning [70]. To train and evaluate the proposed approach, a simulation environment has been created that takes into account flight uncertainty (resulting, for example, from mechanical and navigational errors and wind). Experimental results show [69] that the proposed method can resolve conflicts in scenarios with a much higher traffic density than that in today's real-world situations. 2D and 3D models are compared. The training time of the 2D model is less than 1% of the training time of the 3D model; however, this does not affect the performance of the model during the decision-making due to the nature of reinforcement learning methods. In some scenarios, the 3D model can resolve conflicts more easily by changing the flight level. The 3D model is shown to be superior to the 2D model in terms of success rate and the rate of reduction of additional flight range.

With centralized air traffic control, when the air traffic controller transmits directives to the pilots, actions should be quite rare. However, in models with a continuous action space, agents can make small adjustments to their trajectories at each step. In [71], a multi-agent deep reinforcement learning method with a continuous action space is proposed, in which the number of actions is significantly reduced using a priority mechanism. At each time step, a maximum of one aircraft with the highest priority can perform actions. This approach significantly reduces the number of actions taken while maintaining a high level of conflict prevention performance. The resulting decisions are suitable for centralized air traffic control, where the number of directives that can be transmitted to the pilot is limited. In [72], the priority mechanism based on a dynamic assessment of the proximity of conflicts between aircraft is used in a model with a discrete action space.

3.3. Conflict Prevention Models for an Arbitrary Number of Aircraft

Above, we considered the models, in which the agent has access to information about the state of N nearest aircraft, where N is the hyperparameter selected during the experiments; however, this limits the adaptability of the model. Using the parameter N is a disadvantage since a small change in the location of the aircraft can change the set of N nearest aircraft and, thus, change the input data for the neural network. The neural network should understand that it is almost the same state of the airspace despite the aircraft rearrangements—however, this can be challenging [49]. The solution to this problem is to use more advanced neural network architectures, search for other ways to represent data, use data improvement algorithms, and select the most relevant neighboring aircraft [71]. One of the ways to solve the problem of a variable number of aircraft is to graphically encode information into fixed-size images and use convolutional neural networks (CNN) to extract useful information, similar to air traffic controllers' screens [73]. One can also use recurrent neural networks (RNN) with long short-term memory (LSTM) cells [74] or controlled recurrent blocks (GRU) [75] that work with the entire set of aircraft in the environment, encoding the relevant information into a hidden state of the fixed size.

In [30, 76], multi-agent reinforcement learning is considered to resolve conflicts on routes and intersections in a structured two-dimensional airspace between a variable number of aircraft. The state information is encoded using the LSTM neural network into a fixed-length vector. The agent has access to the encoded information on all aircraft in the sector; in this case, there is no need to determine the value N for each new environment. The BlueSky air traffic control simulator is used as a learning environment [77]. Centralized training and a decentralized execution scheme are used, in which one neural network is trained. This network is used by all agents to receive speed recommendations, and depending on the state, the actions of the agents may vary. The

environment is stochastic because of the uncertainty in the actions of other agents; therefore, the "actor-critic" algorithm called "proximal policy optimization" is used [78].

To ensure the safe separation requirements, an identical reward function is introduced for all agents

$$r_t = \begin{cases} -1 & \text{if } d_o^c < d^{LOS}, \\ -\alpha + \delta d_o^c & \text{if } d_o^c < 10 \ \& \ d_o^c \geqslant d^{LOS}, \\ 0, & \text{otherwise}, \end{cases}$$

where d^{LOS} is the minimum safe separation distance in nautical miles ($d^{LOS} = 3$), d_o^c is the distance from the own aircraft to the nearest aircraft in nautical miles, α and δ are small positive constants to fine agents as they approach losing the safe separation distance. Three practical air traffic scenarios demonstrate the ability to solve decision-making problems with a variable number of agents and uncertainty [30].

Recurrent neural networks process input data sequentially, and the output data depends on this sequence. This can lead to undesirable results in situations when the input sequence does not matter. Transformers were introduced as an alternative to recurrent neural networks for sequential processing of input data in order to ensure parallel learning [79]. Transformers calculate the relative importance of so-called tokens containing information about the states of aircraft using the attention mechanism. In [80], absolute states are used for observation—the coordinates and velocities of the aircraft in the reference system associated with the environment. In [81], relative states are used—the coordinates and velocities of the aircraft in the reference system of the own aircraft with the positive direction of the abscissa axis in the direction of the flight. However, studies have not yet shown the superior performance of the transformer network architecture as compared to feed-forward neural networks and recurrent neural networks to ensure safe aircraft navigation [81].

4. COOPERATIVE CONFLICT PREVENTION STRATEGIES BASED ON NEURAL COMMUNICATION NETWORKS

Communication is a key ability of cooperative multi-agent systems, in which agents can significantly benefit from the information exchange prior to performing joint actions [82]. A model based on neural communication networks, which allows agents to exchange information through a communication protocol, can allow agents to develop cooperative strategies for joint actions to prevent conflicts [83].

In [80], aircraft in the airspace are simulated as agents of a cooperative multi-agent system. The state $s_i = [x_i, y_i, v_i, \chi_i]$ of each agent $i \in N$ consists of coordinates in the Euclidean space x_i, y_i, the velocity v_i, and the course χ_i. The state changes according to the formulas

$$\begin{aligned} x_i(t+1) &= x_i(t) + v_i(t) \sin \chi_i(t) \Delta t, \\ y_i(t+1) &= y_i(t) + v_i(t) \cos \chi_i(t) \Delta t, \\ v_i(t+1) &= v_i(t) + \Delta v_i, \\ \chi_i(t+1) &= \chi_i(t) + \Delta \chi_i, \end{aligned}$$

where Δv_i and $\Delta \chi_i$ are the increments of the velocity and the course, and Δt is the simulation step.

The interaction between agents is represented as the graph $G = (V, E)$, each node corresponds to one agent $i \in N$, and agents that can communicate are connected by the edges e_{ij}. The observation

vector for the current state of the ith agent consists of five elements

$$o_i = \begin{bmatrix} d_i/D \\ \cos(\chi_i - \psi_i) \\ \sin(\chi_i - \psi_i) \\ \bar{v}_i \\ \bar{v}_{ei} \end{bmatrix}^T,$$

where d_i is the shortest distance to the exit point from the observation area, D is the normalizing coefficient, ψ_i is the bearing angle to the exit point, and the normalized velocity and the velocity deviation are determined as

$$\bar{v}_i = \frac{v_i - v_{\min_i}}{v_{\max_i} - v_{\min_i}}; \quad \bar{v}_{ei} = \frac{v_i - v_{opt_i}}{v_{\max_i} - v_{\min_i}}.$$

At each step, the agent encodes its state o_i into a hidden state using the neural network

$$h_i^{(0)} = f_h(o_i).$$

Then, the communication phase begins, consisting of C communication rounds. At each round $c = 0, 1, \ldots, C-1$, each agent's message is calculated as a weighted sum of the edges that connect it to its neighbors based on the attention mechanism,

$$m_i^{(c+1)} = \sum_{j \in N_i} a_{ij}^{(c+1)} e_{ij}^{(c+1)},$$

where N_i is the set of nodes connected by the edges with the node i.

The edge values are calculated using a neural network, taking into account the hidden states of the agents

$$e_{ij}^{(c+1)} = f_e^{(c)}\left([h_i^{(c)}, h_j^{(c)}, e_{ij}^{(c)}]\right).$$

The attention weights are calculated by the formula

$$a_{ij}^{(c+1)} = \frac{\exp\left(v_a^{(c)} f_a^{(c)}([h_i^{(c)}, h_j^{(c)}, e_{ij}^{(c)}])\right)}{\sum_{j \in N_i} \exp\left(v_a^{(c)} f_a^{(c)}([h_i^{(c)}, h_j^{(c)}, e_{ij}^{(c)}])\right)},$$

where $v_a^{(c)}$ is the vector of parameters.

Each node then updates its state using the updating function

$$h_i^{(c+1)} = U^{(c)}\left(h_i^{(c)}, m_i^{(c+1)}\right).$$

After C rounds of communication between the nodes, a probability distribution is generated over all possible actions for each agent

$$a_i = f_a\left([h_i^{(0)}, h_i^{(C)}]\right).$$

The expected reward, which is the same for all agents, is calculated using the read function

$$V^\pi = f_v\left(\sum_{i \in N} f_y([h_i^{(0)}, h_i^{(C)}])\right),$$

$f_h, f_e^{(c)}, f_a^{(c)}, f_a, f_v, f_y$ are feed-forward neural networks.

At each time step, each agent chooses an action a_i, followed by the environment giving a collective reward to the team.

The experiments on training the presented model showed that the total reward per episode increases, and the number of expected conflicts decreases as the number of episodes grows, i.e., agents can improve their policies based on their interaction with the environment.

5. USING GRAPH-BASED DEEP REINFORCEMENT LEARNING TO PREVENT CONFLICTS

In many studies, a multi-agent statement combines the state vectors of several aircraft into a multidimensional vector of the aggregate state using the concatenation operation [30, 84]. However, aggregating the states of all nearest neighboring aircraft, regardless of whether there are potential conflicts between them, can lead to processing redundant data and reduce the efficiency of the model. Such vectors cannot encapsulate spatiotemporal dynamics and distinguish between different levels of risk and urgency in conflict scenarios. Graph reinforcement learning (Graph RL) is designed to process data structured in the form of graphs [85]. Using graph-inherent properties, one can improve scalability, efficiency, and adaptability when working with multidimensional and dynamic environments [86]. Graph reinforcement learning allows using graph properties to represent relationships between planes [87–89]. Graph deep reinforcement learning methods are invariant to the order and number of planes.

In [49], a graph deep reinforcement learning method is proposed for air traffic control in the three-dimensional airspace. To prevent conflicts, the altitude, course, and speed of the aircraft are selected. Planes are represented by the graph vertices; nodes in this graph are connected if the distance between a pair of planes is below a certain threshold. Two approaches are compared, viz. graph neural networks with convolutional layers (GCN) [90] and graph neural networks with the attention mechanism (GAT, Graph Attention Network) [91] used to efficiently aggregate information from the neighboring nodes in the graph. With a normal traffic density, a model with the attention mechanism can prevent 100% of potential collisions and 89.8% of potential conflicts. However, performance deteriorates as the traffic density grows. With the increasing traffic flow density, both methods have difficulty overcoming congested airspace.

In [92], a graph convolutional network with LSTM cells is used to collect the space-time dependencies of flight data, and a graph neural network with increased attention is used to focus on the information characteristics of the key nodes.

In [93], a conflict graph that develops over time was proposed to be used, in which aircraft are represented by nodes, and the connections between them indicate the urgency of the conflict. The urgency of the conflict is determined by the time before the conflict if the current course and speed of the aircraft located at the points A and B (Fig. 5) are maintained. The observation is conducted from the point A. The point B is the center of a circle with the radius R (the radius of the protective zone). Tangents to this circle are drawn from the point A, forming an obstacle

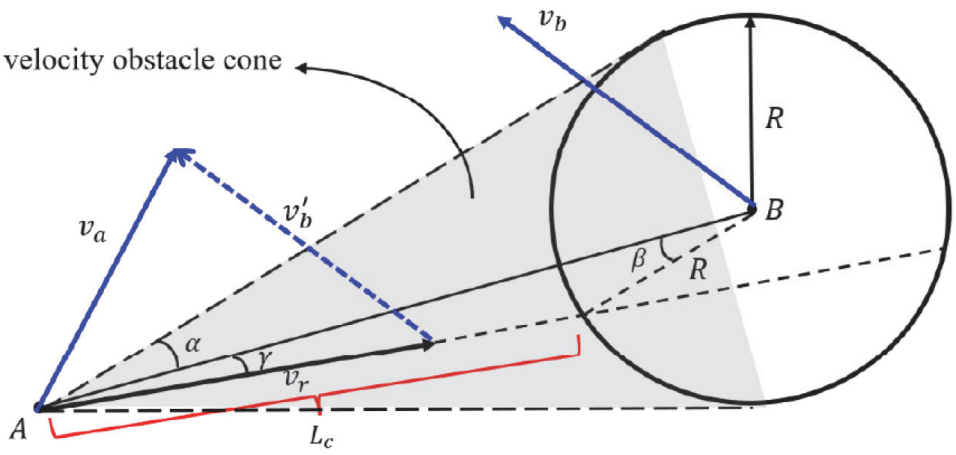

Fig. 5. Conflict geometry.

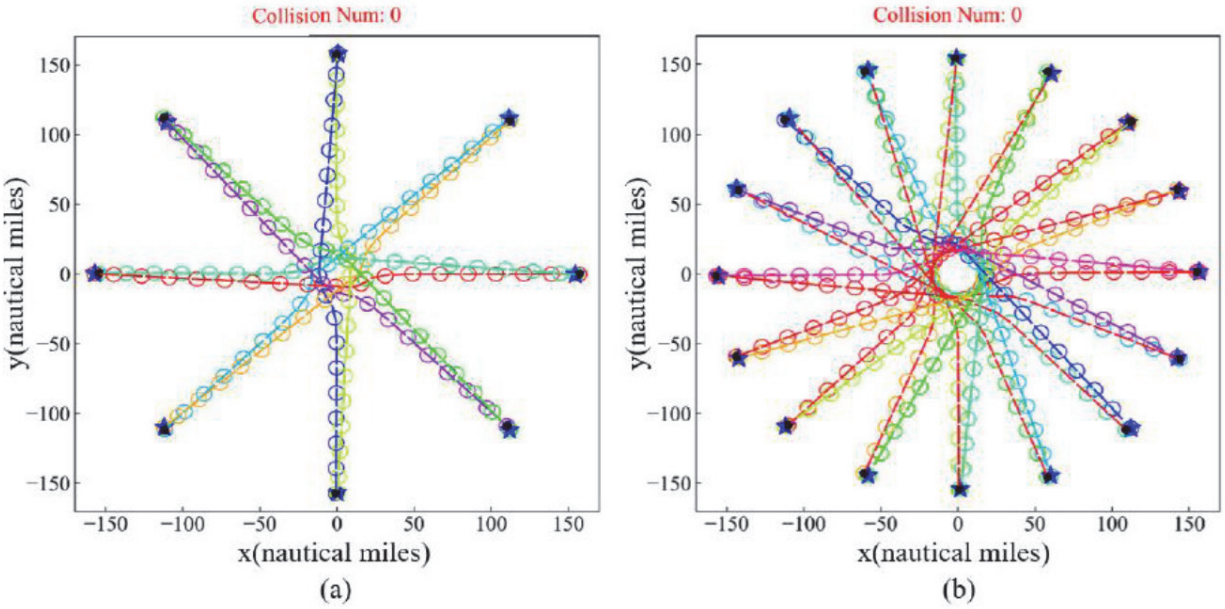

Fig. 6. Circular conflict scenario.

cone; v_A and v_B are the velocities of the aircraft A and B. A potential conflict exists if the relative velocity vector $v_r = v_A - v_B$ is inside the obstacle cone. The time before the conflict is determined based on a geometric model of the relative position and speeds of the pair of the aircraft A and B

$$t_c = \frac{L_c}{|v_r|},$$

where L_c is a straight line segment from the point A along the relative velocity v_r to the point of intersection with the protective zone of the aircraft B. The edge weight of the conflict graph ω_{AB} is normalized in the range $[0, 1]$ as follows

$$\omega_{AB} = e^{-t_c}.$$

If there is no potential conflict, then $\omega_{AB} = 0$; if the planes collided, then $\omega_{AB} = 1$.

Further, based on the conflict graph, information is aggregated using a multi-head neural attention network. The time regularization mechanism is used to increase the training stability. The efficiency of the proposed algorithm is demonstrated, among other things, by two visual scenarios [93]. Circular conflict scenario: In this setup, planes start flying at points in a circle with a radius of 160 nautical miles and fly in opposite directions. This configuration leads to the fact that each plane conflicts with all the others at the center of the circle. The experimental results for scenarios involving 8 and 16 aircraft are shown in Figs. 6a and 6b, respectively. These results demonstrate the ability of the proposed method to manage characteristic circular potential conflicts and prevent all potential collisions. Intersection conflict scenario: Aircraft are divided into two groups, each containing an equal number of them. These groups fly along the intersecting trajectories, causing conflicts at each intersection. Testing was carried out with 20 and 30 aircraft. As we can see in Figs. 7a and 7b, for 20 and 30 aircraft, respectively, the proposed method helps determine conflict-free route points at each time step with minimal deviations from the original trajectories.

Despite significant research successes, there are serious obstacles to the practical application of reinforcement learning methods in the field of air transport due to the tight certification standards

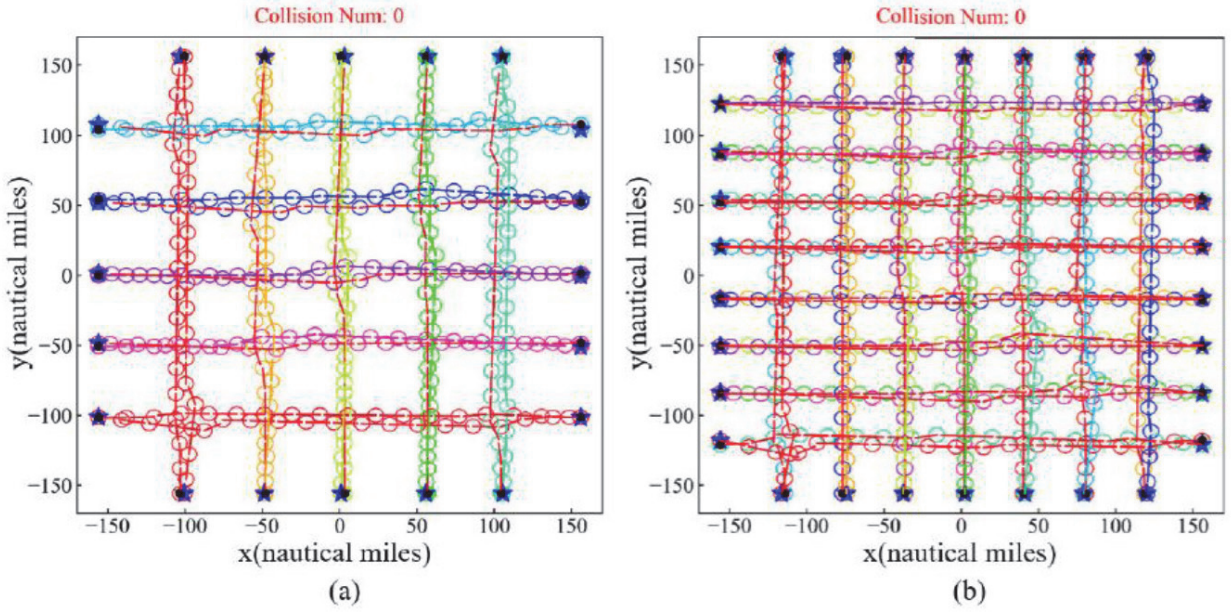

Fig. 7. Intersection conflict scenario.

in the aviation industry. The current regulatory and legal framework fails to provide adequate and acceptable means of meeting the requirements for reinforcement learning applications, and thus there is no legal framework yet in place for their safe use. It is necessary to develop certification recommendations for reinforcement learning models designed for air transport, so that these promising methods can be used in real conditions [94].

6. CONCLUSIONS

It follows from the review of publications that air traffic conflict prevention methods based on deep reinforcement learning are aimed at solving two principal tasks, viz. automatic generation of decision options to support air traffic controllers given the centralized air traffic control and support of autonomous conflict resolution systems in free flight. Models with discrete actions are mainly offered to support decision-making by air traffic controllers. Continuous-action models are designed for autonomous conflict resolution in free flight and allow all agents to perform trajectory correction actions at each time step.

The graph reinforcement learning approach seems to be most promising since the information represented in the form of a conflict graph that develops over time helps reduce the amount of redundant information processed and ensures the scalability of models for different numbers of aircraft. The attention mechanism allows singling out the most urgent information contained in the conflict graph, providing improved conflict prevention strategies in terms of security and efficiency.

With the development of research, the potential for practical application of conflict prevention methods between aircraft based on deep reinforcement learning is becoming more and more obvious. A review of publications shows that the studied reinforcement learning methods demonstrate promising results due to adaptive decision-making in real time to prevent air traffic conflicts. However, there are still serious unresolved issues preventing these methods from being applied in air traffic control practice, where safety is critical. It is impossible to train models in real-world conditions because of the potential damage, and it is impossible to perfectly simulate those conditions. The shift in distribution between the simulated environment and the reality may limit the efficiency of reinforcement learning models. Certification of such models should be one of the directions of research in this area.

REFERENCES

1. *International Civil Aviation Association. Doc 4444: Air Traffic Management, Procedures for Air Navigation Services,* 16th ed. ICAO: Montreal, QC, Canada. 2016.
2. Kulida, E.L. and Lebedev, V.G., Methods for Solving Some Problems of Air Traffic Planning and Regulation. PART I: Strategic Planning of 4D Trajectories, *Control Sciences*, 2023, no. 1, pp. 2–11. https://doi.org/10.25728/cs.2023.1.1
3. Brittain, M. and Wei, P., Autonomous Separation Assurance in a High-Density En Route Sector: A Deep Multi-Agent Reinforcement Learning Approach, *22nd IEEE Intelligent Transportation Systems Conference (ITSC).* Auckland, New Zealand, 2019. https://doi.org/10.109/ITSC.2019.8917217
4. Ponomarev, K.Yu., A Method for Assessing the Dynamic Air Situation for Conflict by Polychromatic Display of Objects in the Information Support of the Air Traffic Control Dispatcher, *Cand. Sci. (Tech.) Dissertation Synopsis*, St. Petersbusrg: Saint Petersburg State University of Civil Aviation, 2023. 24 p.
5. Erzberger, H., Automated Conflict Resolution for Air Traffic Control, *25th International Congress of the Aeronautical Sciences (ICAS)*, Germany, Hamburg. 2006.
6. Farley, T., Field, M., and Erzberger, H., Fast-time Simulation Evaluation of a Conflict Resolution Algorithm Under High Air Traffic Demand, https://www.researchgate.net/publication/255062615, 2007.
7. Erzberger, H. and Heere, K., Algorithm and Operational Concept for Resolving Short-Range Conflicts, *Proc. Inst. Mechan. Engin., Part G: J. Aerospac. Engin*, 2010, vol. 224, no. 2, pp. 225–243. https://doi.org/10.1243/09544100JAERO546
8. Hoekstra, J.M., van Gent, R.N.H.W., and Ruigrok, R.C.J., Designing for Safety: The 'Free Flight' Air Traffic Management Concept, *Reliab. Engin. Syst. Safet.,* 2002, vol. 75, no. 2, pp. 215–232. https://doi.org/10.1016/S0951-8320(01)00096-5
9. Clari, M.V., Ruigrok, R.C.J., Hoekstra, J.M., and Visser, H.G., Cost-Benefit Study of Free Flight with Airborne Separation Assurance, *Air Traffic Control Quarterly*, 2001, vol. 9, no. 4, pp. 287–309. https://doi.org/10.2514/atcq.9.4.287
10. Marjin, N.P., Prospect of Introducing the Concept of "Free Flight," *Problemy bezopasnosti poletov*, 2009, no. 5, pp. 42–55. (In Russ.)
11. Yang, Y., Zhang, J., Cai, K., and Prandini, M., Multi-aircraft Conflict Detection and Resolution Based on Probabilistic Reach Sets, *IEEE Transactions on Control Systems Technology*, 2016, vol. 25, no. 1, pp. 309–316. https://doi.org/10.1109/TCST.2016.2542046
12. Orlov, V.S., Development and Research of Algorithms for Detecting and Preventing Dangerous Approaches in the Air within the Framework of a Promising ATM System, *Cand. Sci. (Tech.) Dissertation*, Moscow: Moscow Aviation Institute. 2015. 116 p.
13. Burkin, V.S., Synthesis of Algorithms for the Detection and Resolution of Collision Conflicts Based on Data from the Automatic Dependent Surveillance System under Uncertainty, *J. Comput. Syst. Sci. Int.*, 2017, vol. 56, no. 3, pp. 492–504. https://doi.org/10.1134/S106423071703008X
14. Kumkov, S.I. and Pyatko, S.G., Fast Algorithms for Detecting Conflict Situations between Aircraft, *Theory of Optimal Control and Applications (OSTA 2022), Materials of the International Conference.* N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences (IMM UB RAS), Yekaterinburg, 2022, pp. 126–131.
15. Pelegrin, M. and D'Ambrosio, C., Aircraft Deconfliction via Mathematical Programming: Review and Insights, *Transportation Science*, 2022, vol. 56, no. 1, pp. 118–140. https://doi.org/10.1287/trsc.2021.1056
16. Cafieri, S., Conn, A.R., and Mongeau, M., Mixed-Integer Nonlinear and Continuous Optimization Formulations for Aircraft Conflict Avoidance via Heading and Speed Deviations, *Eur. J. Oper. Res.*, 2023, vol. 310, no. 2, pp. 670–679. https://doi.org/10.1016/j.ejor.2023.03.002
17. Dias, F. and Rey, D., Aircraft Conflict Resolution with Trajectory Recovery Using Mixed-integer Programming, *J. Global Optim.,* 2024, vol. 90, pp. 1031–1067. https://doi.org/10.1007/s10898-024-01393-1

18. Cecen, R.K. and Cetek, C., Conflict-Free En-Route Operations with Horizontal Resolution Manoeuvers Using a Heuristic Algorithm, *Aeronaut. J.*, 2020, vol. 124, pp. 767–785. https://doi.org/10.1017/aer.2020.5

19. Eby, M., A Self-Organizational Approach for Resolving Air Traffic Conflicts, *Lincoln Lab. J.*, 1994, vol. 7, no. 2, pp. 239–254.

20. Balasooriyan, S., Multi-Aircraft Conflict Resolution Using Velocity Obstacles, *MSc thesis*, Delft University of Technology, 2017, 126 p.

21. Durand, N., Constant Speed Optimal Reciprocal Collision Avoidance, *Transportation Research. Part C, Emerging Technologies*, 2018, pp. 366–379. https://doi.org/0.1016/j.trc.2018.10.004

22. Pan, W., Qin, L., He, Q., and Huang, Y., Three-Dimensional Flight Conflict Detection and Resolution Based on Particle Swarm Optimization, *Aerospace*, 2023, vol. 10, no. 9. https://doi.org/10.3390/aerospace10090740

23. Sui, D. and Zhang, K., A Tactical Conflict Detection and Resolution Method for En Route Conflicts in Trajectory-Based Operations, *J. Advanc. Transport*, 2022, no. 2, pp. 1–16. https://doi.org/10.1155/2022/9283143

24. Brittain, M. and Wei, P., Scalable Autonomous Separation Assurance with Heterogeneous Multi-agent Reinforcement Learning, *IEEE Transactions on Automation Science and Engineering*, 2022, vol. 19, no. 4, pp. 2837–2848. https://doi.org/10.1109/TASE.2022.3151607

25. Samoilov, V.A. and Dotsenko, D.V., The Possibility of Using Neural Networks to Find and Resolve Potential Conflict Situations between Aircraft When Flying in Upper Airspace, *Transport of Russia: Problems and Prospects – 2022. Materials of the International Scientific and Practical Conference*, N.S. Solomenko Institute of Transport Problems of the Russian Academy of Sciences, Saint Petersburg, 2022, pp. 180–184.

26. Wang, Z., Liang, M., and Delahaye, D., Data-Driven Conflict Detection Enhancement in 3D Airspace with Machine Learning, *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, Singapore, 2020. https://doi.org/10.1109/AIDA-AT48540.2020.904.9180

27. Pinto Neto, E.C., Baum, D., Almeida, J.R., et. al., Deep Learning in Air Traffic Management (ATM): Applications, Opportunities, and Open Challenges, *Aerospace 2023*, vol. 10, no. 4. https://doi.org/10.3390/aerospace10040358

28. Razzaghi, P., Tabrizian, A., Guo, W., et. al., A Survey on Reinforcement Learning in Aviation Applications, *Engineering Applications of Artificial Intelligence*, 2024, vol. 136, no. 3. https://doi.org/10.1016/j.engappai.2024.108911

29. Kulida, E.L. and Lebedev, V.G., Methods for Solving Some Problems of Air Traffic Planning and Regulation. PART II: Application of Deep Reinforcement Learning, *Control Sciences*, 2023, no. 2, pp. 2–14. https://doi.org/10.25728/cs.2023.2.2

30. Brittain, M.W. and Wei, P., One to Any: Distributed Conflict Resolution with Deep Multi-agent Reinforcement Learning and Long Short-Term Memory, *AIAA Scitech 2021 Forum*, Nashville, Tennessee, USA. https://doi.org/10.2514/6.2021-1952

31. Wang, Z., Pan, W., Li, H., et. al., Review of Deep Reinforcement Learning Approaches for Conflict Resolution in Air Traffic Control, *Aerospace*, 2022, vol. 9, no. 6. https://doi.org/10.3390/aerospace9060294

32. Groot, J., Ribeiro, M., Ellerbroek, J., et. al., Improving Safety of Vertical Manoeuvres in a Layered Airspace with Deep Reinforcement Learning, *International Conference on Research in Air Transportation (ICRAT)*, Tampa, Florida, USA, 2022, pp. 19–23.

33. Ribeiro, M., Ellerbroek, J., and Hoekstra, J., Review of Conflict Resolution Methods for Manned and Unmanned Aviation, *Aerospace*, 2020, vol. 7, no. 6. https://doi.org/10.3390/aerospace7060079

34. Ribeiro, M., Ellerbroek, J., and Hoekstra, J., Distributed Conflict Resolution at High Traffic Densities with Reinforcement Learning, *Aerospace*, 2022, vol. 9, no. 9. https://doi.org/10.3390/aerospace9090472

35. Ribeiro, M., Conflict Resolution at High Traffic Densities with Reinforcement Learning, *PhD Thesis,* 2023. https://doi.org/10.4233/uuid:a2979919-cb01-41d1-bbba-fefa9079463b

36. Ribeiro, M., Ellerbroek, J., and Hoekstra, J., Improving Algorithm Conflict Resolution Manoeuvres with Reinforcement Learning, *Aerospace*, 2022, vol. 9, no. 12. https://doi.org/10.3390/aerospace9120847

37. Vizilter, Yu.V., Vishnyakov, B.V., and Zheltov, S.Yu., Modern Artificial Intelligence Technologies and Their Application in Aviation Complexes, *16th All-Russian Multi-conference on Management Problems (MCPU-2023)*, Volgograd, Materials of the multi-conference in 4 volumes, vol. 3, pp. 13–16.

38. Sui, D., Ma, C., and Wei, C., Tactical Conflict Solver Assisting Air Traffic Controllers Using Deep Reinforcement Learning, *Aerospace*, 2023, vol. 10, no. 2. https://doi.org/10.3390/aerospace10020182

39. Bastas, A. and Vouros, G., Data-Driven Modeling of Air Traffic Controllers' Policy to Resolve Conflicts, Aerospace, 2023, vol. 10, no. 6. https://doi.org/10.3390/aerospace10060557

40. Kulida, E.L. and Lebedev, V.G., Problems in the Application of Machine Learning Methods in Aviation, *Proc. of the 16th International Conference "Management of Large-Scale Systems Development" (MLSD'2023, Moscow)*, Moscow, Institute of Control Sciences of RAS, pp. 1315–1320. https://doi.org/10.25728/mlsd.2023.1315

41. Degas, A., Islam, M.R., Hurter, C., et. al., A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory, *Appl. Sci.*, 2022, vol. 12, iss. 3. https://doi.org/10.3390/app12031295

42. Wang, L., Yang, H., Lin, Y., et. al., Enhancing Air Traffic Control: A Transparent Deep Reinforcement Learning Framework for Autonomous Conflict Resolution, *Expert Systems with Applications,* 2024, vol. 260(2). https://doi.org/10.06/j.eswa.2024.125389

43. Sutton, R.S. and Barto, A.G., *Reinforcement Learning: An Introduction*, MIT Press, 1998.

44. Graesser, L. and Keng, W.L., *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*, Addison-Wesley Professional, 2019.

45. Morales, M., *Grokking Deep Reinforcement Learning*, Manning, 2010.

46. Sui, D., Ma, C., and Dong, J., Conflict Resolution Strategy Based on Deep Reinforcement Learning for Air Traffic Management, *Aviation*, 2023, vol. 27, iss. 3, pp. 177–186. https://doi.org/10.3846/aviation.2023.19720

47. Hasselt, H.V., Double Q-Learning, *24th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2010, pp. 2613–2621.

48. Brittain, M. and Wei, P., Autonomous Aircraft Sequencing and Separation with Hierarchical Deep Reinforcement Learning, *International Conference for Research in Air Transportation*, Castelidefeil, Spain, 2018.

49. Mollinga, J. and Hoof, H., An Autonomous Free Airspace En-route Controller Using Deep Reinforcement Learning Techniques, *9th International Conference on Research in Air Transportation (ICRAT),* Tampa, Florida, USA, 2020.

50. Sui, D., Xu, W., and Zhang, K., Study on the Resolution of Multi-aircraft Flight Conflicts Based on an IDQN, *Chin. J. Aeronaut.*, 2021, vol. 35, no. 11, pp. 195–213. https://doi.org/10.06/j.cja.2021.03.015

51. Li, S., Egorov, M., and Kochenderfer, M.J., Optimizing Collision Avoidance in Dense Airspace Using Deep Reinforcement Learning, *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar*, Vienna, Austria. June 17–21, 2019. https://doi.org/10.48550/arXiv.1912.10146

52. Hermans, M.C., Towards Explainable Automation for Air Traffic Control Using Deep Q-Learning from Demonstrations and Reward Decomposition. *Master's Thesis*, Delft University of Technology, Delft, The Netherlands, May 2020.

53. Pham, D.-T., Tran, N.P., Goh, S.K., et. al., Reinforcement Learning for Two-aircraft Conflict Resolution in the Presence of Uncertainty, *IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, Singapore, 2019. https://doi.org/10.1109/RIVF.2019.8713624

54. Wang, Z., Li, H., Wang, J., et. al., Deep Reinforcement Learning Based Conflict Detection and Resolution in Air Traffic Control, *IET Intelligent Transport Systems,* 2019, vol. 13, iss. 6, pp. 1041–1047. https://doi.org/10.1049/iet-its.2018.5357

55. Badea, C.A., Groot, J., Morfin Veytia, A., et. al., Lateral and Vertical Air Traffic Control Under Uncertainty Using Reinforcement Learning, *Proc. of the 12th SESAR Innovation Days*, Budapest, Hungary, 2022.

56. Pham, D., Tran, N., Alam, S., et. al., A Machine Learning Approach for Conflict Resolution in Dense Traffic Scenarios with Uncertainties, *13th USA/Europe Air Traffic Management Research and Development Seminar*, Vienne, Austria, June 2019.

57. Wen, H., Li, H., and Wang, Z., Application of DDPG-based Collision Avoidance Algorithm in Air Traffic Control, *12nd International Symposium on Computational Intelligence and Design*, Hangzhou, China, 2019, pp. 130–133. https://doi.org/10.1109/ISCID.2019.00036

58. Pham, D.-T., Tran, P.N., Alam, S., et. al., Deep Reinforcement Learning Based Path Stretch Vector Resolution in Dense Traffic with Uncertainties, *Transportation Research, Part C*, 2022, vol. 135, no. 3. https://doi.org/10.1016/j.trc.2021.103463

59. Mukherjee, A., Guleria, Y., and Alam, S., Deep Reinforcement Learning for Air Traffic Conflict Resolution Under Traffic Uncertainties, *41st Digital Avionics Systems Conference (DASC)*, Portsmouth, USA, 2022. https://doi.org/10.1109/DASC55683.2022.9925772

60. Sunil, E., Ellerbroek, J., and Hoekstra, J. M., Camda: Capacity Assessment Method for Decentralized Air Traffic Control, *International Conference on Research in Air Transportation (ICRAT)*, Barcelona, Spain, 2018, pp. 26–29.

61. Dudoit, A., Rimsa, V., and Bogdevicius, M., Investigation of Aircraft Conflict Resolution Trajectories under Uncertainties, *Sensors*, 2024, vol. 24, no. 18. https://doi.org/10.3390/s24185877

62. Busoniu, L., Babuska, R., and De Schutter, B., A Comprehensive Survey of Multi-agent Reinforcement Learning, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2008, vol. 38, no. 2, pp. 156–172. https://doi.org/10.1109/TSMCC.2007.913919

63. Tan, M., Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents, *10th International Conference on Machine Learning (ICML)*, 1993, pp. 330–337. https://doi.org/10.1016/B978-1-55860-307-3.50049-6

64. Matignon, L., Laurent, G.J., and Le Fort-Piat, N., Independent Reinforcement Learners in Cooperative Markov Games: a Survey Regarding Coordination Problems, *Knowledge Engineering Review*, 2012, vol. 27, no. 1. https://doi.org/10.1017/S0269888912000057

65. Everett, M., Chen, Y.F., and How, J.P., Motion Planning Among Dynamic, Decision-making Agents with Deep Reinforcement Learning, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3052–3059. https://doi.org/10.48550/arXiv.1805.01956

66. Chen, Y., Hu, M., Yang, L., et. al., General Multi-agent Reinforcement Learning Integrating Adaptive Manoeuvre Strategy for Real-time Multi-Aircraft Conflict Resolution, *Transportation Research. Part C. Emerging Technologies*, 2023, vol. 151. https://doi.org/10.1016/j.trc.2023.104125

67. Xu, Q., Chen, Z., Li, F., et. al., An Efficient Aircraft Conflict Detection and Resolution Method Based on an Improved Reinforcement Learning Framework, *Int. J. Aerospac. Engin.*, 2023, vol. 1, pp. 1–16. https://doi.org/10.1155/2023/6643903

68. Ghosh, S., Laguna, S., Lim, S.H., et. al., A Deep Ensemble Method for Multi-Agent Reinforcement Learning: A Case Study on Air Traffic Control, *31st International Conference on Automated Planning and Scheduling*, SuiGuangzhou, China, 2021, pp. 468–476. https://doi.org/10.1609/icaps.v31i1.15993

69. Chen, Y., Xu, Y., Yang, L., et. al., General Real-Time Three-Dimensional Multi-Aircraft Conflict Resolution Method Using Multi-Agent Reinforcement Learning, *Transportation Research. Part C. Emerging Technologies*, vol. 157. https://doi.org/10.1016/j.trc.2023.104367

70. Hessel, M., Modayil, J., Van Hasselt, H., et. al., Rainbow: Combining Improvements in Deep Reinforcement Learning, *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 3. https://doi.org/10.1609/aaai.v32i1.11796

71. Nilsson, J., Unger, J., and Eilertsen, G., Self-Prioritizing Multi-Agent Reinforcement Learning for Conflict Resolution in Air Traffic Control with Limited Instructions, *Aerospace*, 2025, vol. 12, no. 2. https://doi.org/10.3390/aerospace12020088

72. Sui, D., Zhou, Z., and Cui, X., Priority-based Intelligent Resolution Method of Multi-Aircraft Flight Conflicts, *Aeronaut. J.,* 2024, vol. 129, pp. 326–350. https://doi.org/10.1017/aer.2024.75

73. Zhao, P. and Liu, Y., Physics Informed Deep Reinforcement Learning for Aircraft Conflict Resolution, *IEEE Transactions on Intelligent Transportation Systems*, 2021, vol. 23, no. 7, pp. 8288–8301.

74. Hochreiter, S. and Schmidhuber, J., Long Short-Term Memory, *Neural Computation*, 1997, vol. 9, no. 8, pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

75. Cho, K., Merrienboer, B., Gulcehre, C., et. al., Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179

76. Brittain, M.W., Yang, X., and Wei, P., Autonomous Separation Assurance with Deep Multi-agent Reinforcement Learning, *J. Aerospac. Inform. Syst.,* 2021, vol. 18, no. 12, pp. 890–905. https://doi.org/10.254/1.1010973

77. Hoekstra, J.M. and Ellerbroek, J., BlueSky ATC Simulator Project: an Open Data and Open Source Approach, *7th International Conference on Research in Air Transportation*, Philadelphia, USA, 2016, vol. 131, p. 132.

78. Schulman, J., Wolski, F., Dhariwal, P., et. al., Proximal Policy Optimization Algorithms, *ArXiv*, 2017. https://doi.org/10.48550/arXiv.1707.06347

79. Vaswani, A., Shazeer, N., Parmar, N., et. al., Attention is All You Need, *31st Conference on Neural Information Processing Systems (NIPS)*, Long Beach, USA, 2017. https://doi.org/10.48550/arXiv.1706.03762

80. Dalmau, R. and Allard, E., Air Traffic Control Using Message Passing Neural Networks and Multi-agent Reinforcement Learning, *Conference: SESAR Innovation Days (SID), Virtual Event*, 2020, pp. 7–10.

81. Groot, J., Ellerbroek, J., and Hoekstra, J., Using Relative State Transformer Models for Multi-Agent Reinforcement Learning in Air Traffic Control, *Conference: SESAR Innovation days (SID),* Seville, Spain, 2023.

82. Wollkind, S., Valasek, J., and Ioerger, T., Automated Conflict Resolution for Air Traffic Management Using Cooperative Multi-Agent Negotiation, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004. https://doi.org/10.2514/6.2004-4992

83. Pritchett, R. and Genton, A., Negotiated Decentralized Aircraft Conflict Resolution, *IEEE Transactions on Intelligent Transportation Systems*, 2017, vol. 19, no. 1, pp. 81–91. https://doi.org/10.1109/TITS.2017.2693820

84. Lai, J., Cai, K., Liu, Z., et. al., A Multi-Agent Reinforcement Learning Approach for Conflict Resolution in Dense Traffic Scenarios, *IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, San Antonio, USA, 2021. https://doi.org/10.1109/DASC52595.2021.9594437

85. Wu, Z., Pan, S., Chen, F., et. al., A Comprehensive Survey on Graph Neural Networks, 2019. https://doi.org/10.48550/arXiv.1901.00596

86. Mendonça, M., Ziviani, A., and Barreto, A., Graph-Based Skill Acquisition for Reinforcement Learning, *ACM Computing Surveys (CSUR)*, 2019, vol. 52, no. 1. https://doi.org/10.1145/3291045

87. Papadopoulos, G., Bastas, A., Vouros, G.A., et. al., Deep Reinforcement Learning in Service of Air Traffic Controllers to Resolve Tactical Conflicts, *Expert Systems with Applications*, 2024, vol. 236, no. 1. https://doi.org/10.1016/j.eswa.2023.121234

88. Isufaj, R., Aranega Sebastia, D., and Piera, M.A., Toward Conflict Resolution with Deep Multi-Agent Reinforcement Learning, *J. Air Transport*, 2022, vol. 30, no. 3, pp. 71–80. https://doi.org/10.2514/1.DO296

89. Vouros, G., Papadopoulos, G., Bastas, A., et. al., Automating the Resolution of Flight Conflicts: Deep Reinforcement Learning in Service of Air Traffic Controllers, *PAIS 2022*, pp. 72–85. https://doi.org/10.48550/arXiv.2206.07403

90. Kipf, T.N. and Welling, M., Semi-supervised Classification with Graph Convolutional Networks, 2017. https://doi.org/10.48550/arXiv.1609.02907

91. Velickovic, P., Cucurull, G., Casanova, A., et. al., Graph Attention Networks, 2018. https://doi.org/10.48.550/arXiv.170.0903

92. Zhang, Y., Xu, S., Zhang, L., et. al., Short-Term Multi-Step-Ahead Sector-Based Traffic Flow Prediction Based on the Attention-Enhanced Graph Convolutional LSTM Network (AGC-LSTM), *Neural Computing and Applications*, 2024. https://doi.org/10.1007/s00521-024-09827-3

93. Li, Y., Zhang, Y., Guo, T., et. al., Graph Reinforcement Learning for Multi-Aircraft Conflict Resolution, *IEEE Transactions on Intelligent Vehicles*, 2024. https://doi.org/10.1109/TIV.2024.3364652

94. Ribeiro, M., Tseremoglou, I., and Santos, B., Certification of Reinforcement Learning Applications for Air Transport Operations Based on Criticality and Autonomy, *AIAA Science and Technology Forum and Exposition*, Orlando, Florida, USA, 2024. https://doi.org/10.2514/6.2024-1463

This paper was recommended for publication by N.I. Selvesyuk, a member of the Editorial Board