Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 9283143, 16 pages
https://doi.org/10.1155/2022/9283143

Research Article

A Tactical Conflict Detection and Resolution Method for En Route Conflicts in Trajectory-Based Operations

Dong Sui and Kai Zhang

The College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Correspondence should be addressed to Kai Zhang; zhangkainone@nuaa.edu.cn

Received 11 April 2021; Revised 20 November 2021; Accepted 2 February 2022; Published 24 February 2022

Academic Editor: David Rey

In trajectory-based operation (TBO), the four-dimensional trajectories (4DTs) of aircraft are shared with all flight-related stakeholders, which makes flights visible and controllable and helps flights arrive at a location within a fixed time range. With the technical support from TBO, the operation of flights in a route sector will be more efficient and the air traffic volume will increase. However, more flights will also result in more flight conflicts. In this study, a deterministic conflict detection and resolution (CDR) module is established to assist air traffic controllers in detecting and resolving high-density conflicts rapidly and in advance. In the conflict detection (CD) submodule, a spatial data structure with low time complexity, the R tree algorithm, is used. R tree can effectively reduce the comparison number between the 4DTs of all aircraft. The experiment results show that the computing time of the R tree presents a logarithmic curve with the increase in the number of aircraft and the efficiency of the CD is more significantly improved. In the conflict resolution (CR) submodule, considering the aircraft performance and terrain constraints, the Monte Carlo tree search (MCTS) algorithm is proposed to solve the problem of huge search space and to quickly provide an effective resolution policy for pairwise conflict. The simulation results in dense airspace indicate that the MCTS-based CR algorithm has good performance in terms of safety and efficiency.

1. Introduction

In recent years, the demand for civil aviation has gradually increased. To accommodate the rise of air traffic volume, the federal aviation administration (FAA) [1], EUROCONTROL [2], and the international civil aviation organization (ICAO) [3] have performed a series of relevant studies on trajectory-based operations (TBOs). In TBO, the four-dimensional trajectories (4DTs) calculated by the flight management system (FMS) are shared with all flight-related stakeholders (such as control units and airlines) [4], which makes flights visible and controllable and helps flights arrive at a location within a fixed time range, which in turn improves the efficiency of air traffic operation in dense airspace. In the civil aviation air traffic management (ATM) modernisation strategy (CAAMS) issued by the civil aviation administration of China (CAAC), TBO is one of the important concepts, and it will provide powerful guidance for the future development of the ATC system of China [5].

However, the growth of air traffic flow will lead to more flight conflicts in airspace, which will produce tremendous workload stress for air traffic controllers. Therefore, it is necessary for controllers to have a decision support tool (DST) for conflict detection and resolution (CDR) that can autonomously detect conflicts and provide feasible solutions to help controllers avoid conflicts in time.

1.1. Related Work. The intervals between two aircraft include the horizontal interval and the vertical interval. A pairwise flight conflict will occur if neither of the intervals meets the requirements of ATM at the same time [6] (the criterion of the horizontal interval is 10 km, and that of the vertical interval is 300 m in this study). In TBO, conflict can be detected by judging whether there are spatiotemporal overlaps between the 4DTs of aircraft [7].

The study of conflict detection (CD) is focused on determining how to make predictions accurately and rapidly, and CD methods can be divided into three categories: deterministic (nominal), worst case, and probabilistic, according to reference [8]. In TBO, however, most uncertainty (except that caused by wind) for the waypoints of a flight path can be eliminated or controlled within an acceptable range using 4DTs downloaded from aircraft [9]. Therefore, determining how to utilise definite and accurate 4DTs to detect conflict efficiently is the focus of this research; that is, we study the deterministic CD method.

Pairwise comparison (the pairwise algorithm) is the simplest and the most direct algorithm for CD, and it is used to compare all aircraft in an airspace and check whether the safety interval standards between two aircraft are met one by one [10]. However, when there are more aircraft, the pairwise algorithm cannot withstand the huge amount of computation, resulting in the inability to meet the real-time requirements of CD. Therefore, nonpairwise algorithms, such as various search methods and planning methods, are proposed to reduce the computing time.

Previous deterministic CD studies have commonly used a mathematical geometric model to set up aircraft protection zones [10, 11]. Based on this, the grid-based methods have also been proposed to reduce the calculation amount by using the "space-for-time" method [12, 13]. In addition, an integrated system composed of multiple CD algorithms can reduce the number of missed detections and false alarms while improving accuracy [14]. These methods have achieved satisfactory results. However, the detection process is complex and costly, and the problem of computing time in high traffic density still exists, so scaling these methods for more crowded air traffic can be challenging. The spatial data structure (SDS) method has been proposed to detect time-based separation infringements between aircraft. It has been reported for the SDS-based CD algorithm in reference [7] that using a simplified wake vortex model with 4D tubes can avoid nonefficient pairwise trajectory comparisons. However, the 4DTs of all aircraft need to be sorted and stored in the database, which occupies a certain count of storage space, and the process is relatively complicated. Therefore, another SDS-based method, the R tree algorithm, is applied to CD, and this method is simple and easy to implement and has lower time complexity.

A conflict must be resolved after being detected. Like CD, conflict resolution (CR) is also a problem with finite discrete solution space, and its optimality can be verified. Conventional CR methods can be classified into three categories. The first category is swarm intelligence optimisation algorithms (such as genetic algorithms [15–18] or ant colony algorithms [19]). These kinds of algorithms usually divide the resolution process into a series of discrete segments with equal times or distance lengths and optimise them separately. The second category is the optimal control theory [20, 21]. These algorithms are primarily used to construct the optimal CR model of aircraft with optimal control theory and to design the avoidance path of aircraft on a single or multidimensional level. The third category is hybrid system models, which delegate the CR function of the controllers to each aircraft [22, 23]. The core concept is to propose decentralised management schemes that will switch the hybrid control system to different optimal control solutions according to the change in the information structure between agents. However, the results of the CR are easily affected due to the complicated process and other external factors, which limit these methods for CR in an environment of higher density.

Remarkably, some artificial intelligence methods have been applied to the CR problem in recent years, especially the deep reinforcement learning (DRL) method. In 2019, the K-control actor-critic (KCAC) algorithm chose the random position for aircraft to avoid conflict [24], and a suitable training and learning environment for aircraft conflict detection and resolution was developed by changing the 2D continuous speed actions and headings to obtain the optimal or suboptimal conflict-free flight trajectory. This method can output the resolution policy at the millisecond level and has obvious advantages in terms of calculation efficiency, but the cost of obtaining a relatively reliable model is somewhat high because of the training process and the testing of many hyperparameters. Moreover, the stability of the model is also affected by the sample data, which may make the CR performance fluctuate frequently.

1.2. Main Contributions.

There are two main contributions in this paper. First, a new, simple but efficient spatial data structure, the R tree, is used to detect en route conflicts in the CD submodule. The R tree is a nonpairwise algorithm with logarithmic time complexity, which can effectively reduce the computing time for identifying conflicts. The R tree is built rapidly, and the 4DTs for all aircraft updated in a certain interval can be flexibly processed. Second, to balance the advantages and disadvantages between traditional methods and DRL methods in terms of computational costs and success rates, the Monte Carlo tree search (MCTS) algorithm is selected for the CR process. The MCTS can relieve the problem of falling into local solutions in a huge search space and quickly provide an effective resolution policy. The experiment results show that compared with the pairwise algorithm, the CD efficiency of the R tree is both consistent with its complexity and significantly improved for the same detection accuracy, and the MCTS-based CR algorithm has good performance in terms of real-time function and safety.

1.3. Four-Dimensional Trajectory Prediction.

Four-dimensional trajectory prediction (4DTP) is a basic module in flight simulation. Combined with the current status of the aircraft (such as the weight, altitude, and type) and intention information (such as the flight plan), 4DTP simulates the FMS of the aircraft and calculates the future 4DTs for CD. In addition, 4DTP can help the CR module to check whether the resolution actions are effective, convert the actions into 4DT instructions that are based on the time window, and finally upload the instructions to the aircraft to avoid conflict. The accuracy of 4DTP directly affects the reliability and the overall performance of the CDR module. Therefore, an accurate aircraft motion model based on BADA (base of aircraft data) is used to develop the 4DTP of this study.

BADA is a database of aircraft performance parameters developed by Eurocontrol [25]. BADA analyses each stage of the aircraft operation in detail and provides thrust, drag, fuel consumption, and other parameters. The current flight state of the aircraft can be determined according to the real-time thrust, acceleration, rate of climb, and descent. The BADA-based 4DTP regards the aircraft flying in the route as a particle, simulates the force analysis of the particle under different conditions, and then calculates the future 4DTs of the aircraft according to the performance data in BADA, the motion formula, and the intention model.

The aircraft motion model is mainly determined by the total energy model (TEM) and the horizontal motion model. The TEM is the core of 4DTP and is used for the state transition of an aircraft in the vertical direction (such as an altitude change). A descending aircraft is taken as an example:

$$m\frac{\mathrm{d}v_{\mathrm{TAS}}}{\mathrm{d}t} = (T - D) + mg\sin\gamma,\qquad \sin\gamma \approx \frac{\mathrm{d}h}{\mathrm{d}t\cdot v_{\mathrm{TAS}}}, \tag{1}$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{(T-D)v_{\mathrm{TAS}}}{mg}\left(1 + \frac{v_{\mathrm{TAS}}}{g}\cdot\frac{\mathrm{d}v_{\mathrm{TAS}}}{\mathrm{d}h}\right)^{-1}, \tag{2}$$

where m is the aircraft mass, v_{TAS} is the true airspeed of the aircraft, T and D are the thrust and drag, respectively, g is the gravity acceleration, γ is the descent angle, and h is the altitude. In (2), $[1 + (v_{\mathrm{TAS}}/g)\cdot(\mathrm{d}v_{\mathrm{TAS}}/\mathrm{d}h)]^{-1}$ is the energy distribution coefficient, which can be converted into $f(M)$ with the Mach number M, indicating the ratio of the energy used for climbing or descending to the total available energy.

The horizontal motion model is composed of two sections: linear motion and turning motion. The linear motion of the aircraft is affected by the wind direction φ_{wind} and the wind speed v_{wind}. The flight distance Δd and the heading β_{MH} are obtained by calculating the drift angle ψ_{DA} and the ground speed v_{GS}. The calculation formula is as follows:

$$\begin{aligned}\theta_{\mathrm{WA}} &= \varphi_{\mathrm{wind}} - \delta_{\mathrm{MC}},\\ \psi_{\mathrm{DA}} &= \arcsin\left(\frac{v_{\mathrm{wind}}}{v_{\mathrm{TAS}}}\cdot\sin\theta_{\mathrm{WA}}\right),\\ v_{\mathrm{GS}} &= v_{\mathrm{TAS}}\cos\psi_{\mathrm{DA}} + v_{\mathrm{wind}}\cos\theta_{\mathrm{WA}},\\ \Delta d &= v_{\mathrm{GS}}\Delta t,\\ \beta_{\mathrm{MH}} &= \delta_{\mathrm{MC}} + \psi_{\mathrm{DA}},\end{aligned} \tag{3}$$

where θ_{WA} is the wind angle, δ_{MC} is the course angle, and Δt is the time step.

The turning motion of the aircraft uses the "flyby" method. As shown in Figure 1, r is the turn radius, o and o' represent the centre of the turn cycle, α is the angle between two segments, L is the distance between the starting point of turning and the endpoint of the current segment, h and h' are the difference between the aircraft altitude and 8,100 m at different times ($h < h'$, as the aircraft in the figure is climbing). The turn rate is calculated with $R = g\cdot\tan\phi/v_{\mathrm{TAS}}$, where ϕ is the rotation angle of the aircraft body. When the turning angle of the aircraft is equal to or greater than α, the turning of the aircraft is over, and the aircraft enters the next leg.

The intention model can be regarded as an abstract description of the pilot or FMS controlling the aircraft movement according to the operational requirements and constraints (such as the flight plan and environment), that is, a set of instructions including speed control, altitude control, horizontal control, and the change of aircraft configuration.

2. Conflict Detection Submodule

In TBO, an aircraft is required to download its future 4DTs and other information (including true airspeed, wind, and atmospheric pressure) to the ground automation system for the CD. The scope of this research is the tactical CDR in the en route stage. The CD module detects all conflicts five minutes in advance, and then, the CR module gives the actions executed within these five minutes. Because the dimensions of the data set composed of 4DTs and other information become higher with the increase of the number of aircraft, in order to reduce the calculation amount, the R tree that has a good efficiency in processing high-dimensional spatial data is applied to the CD.

2.1. R Tree. The R tree, proposed by Guttman in 1984 [26], is a high-performance data index structure for spatial query and processing technology that solves the problem of storing and searching in high-dimensional space, and its average time complexity is $O(\log N)$ (it is known that the time complexity of the pairwise algorithm is $O(N^2)$).

The R tree is a tree-like storage and index structure based on a minimum bounding rectangle (MBR). The "R" of the R tree is the abbreviation of "Rectangle." As shown in Figure 2, it is assumed that 12 different irregular spatial objects are surrounded by 12 MBRs (R8–R19), which are taken as the leaf nodes that have only a parent node and no child node. Then, the MBRs with a close distance are classified and surrounded by larger MBRs (R3–R7), which are the nonleaf nodes that both the parent and child nodes have. The process continues until only two MBRs (R1 and R2) form the root node that has only child nodes and no parent node.

After being built, the R tree can be used to quickly obtain the spatial objects in the range of point P (thick blue line, hereinafter referred to as the search box). The steps are as follows: (1) Traversal from the root nodes R1 and R2 occurs. If a node intersects with the search box, the location relationship between the child nodes and the search box will be checked. R4 is the child node where R1 intersects the search

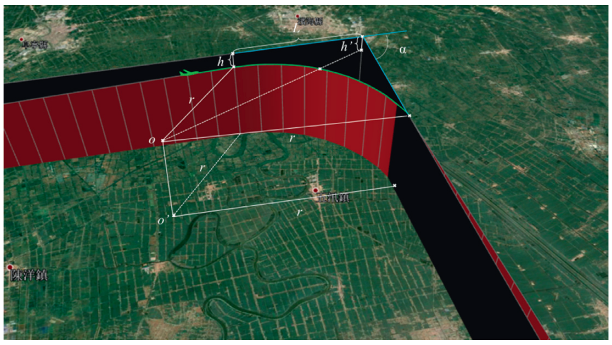

Figure 1: The diagram of the aircraft turning (3D): The blue line is the planned trajectory, and the green line is the trajectory predicted by the BADA.

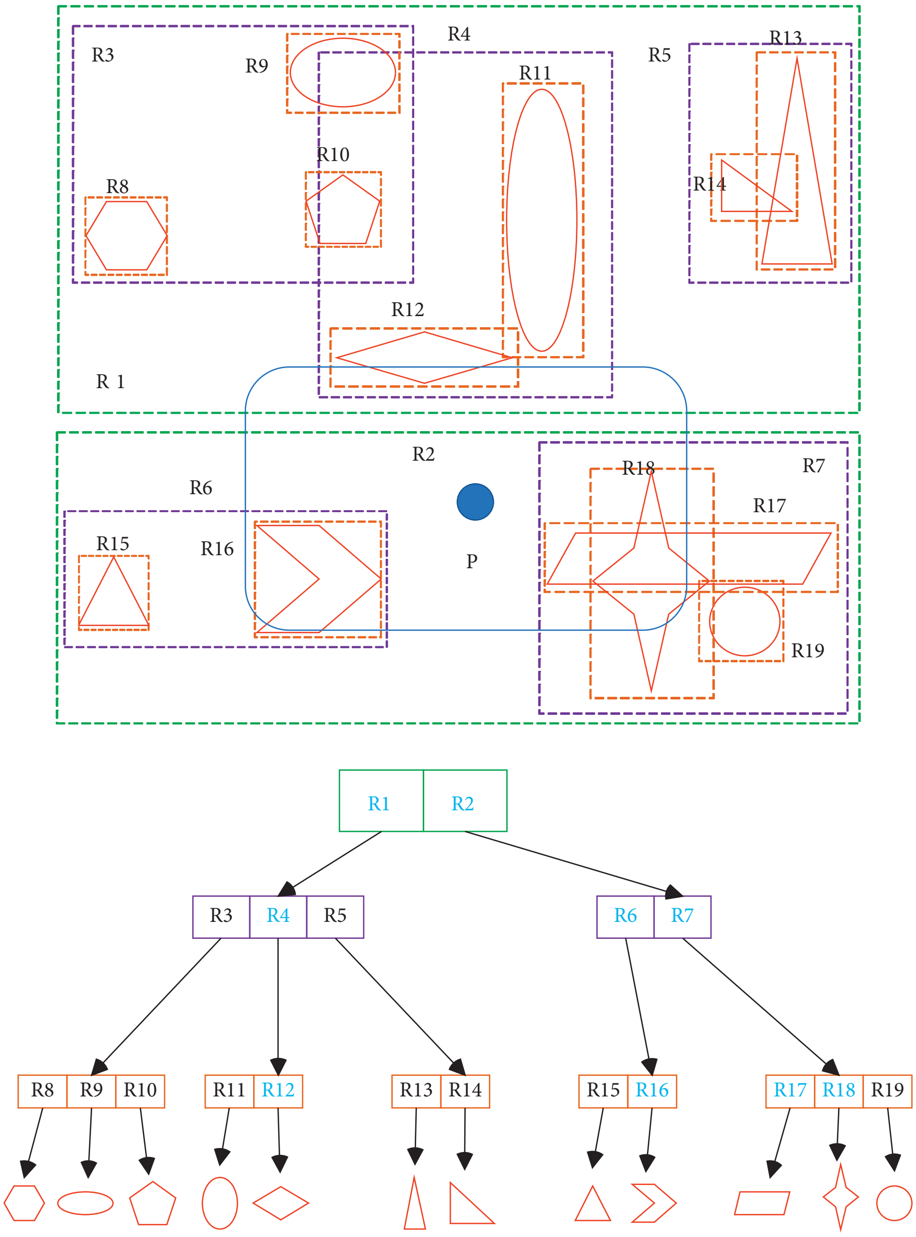

Figure 2: Schematic diagram of the R tree structure [27].

box, while R6 and R7 are the intersected child nodes of R2. (2) R4, R6, and R7 are traversed. With a similar method, the leaf nodes intersecting with the search box, including R12, R16, R17, and R18, are found, and the search process ends. (3) The returned result is that the irregular figures in R12, R16, R17, and R18 are within the range of point P.

2.2. Conflict Detection Algorithm. The CD based on the R tree regards all aircraft as spatial objects, as shown in Figure 2, and constructs the R tree by using the aircraft positions (Figure 3). The surrounding aircraft in the detection range of an aircraft is returned quickly (C and D are two aircraft in the detection range of aircraft A), and then, it is judged whether the distances of A-C and A-D meet the above standards, regardless of the aircraft beyond the detection range. It is worth noting that when the detection range is extended to three dimensions that are equal to the safety interval standards; that is, when the length, width, and height of the detection range are 20 km, 20 km, and 600 m, respectively, both C and D are regarded as the conflicting aircraft for A without further judgment. R tree effectively reduces the number of pairwise comparisons, the number of aircraft is greater, and the detection efficiency is significantly improved.

Figure 4 shows the flowchart of the CD algorithm based on the R tree, and it is assumed that the current time is T. The time will be $T + 300$ after 5 minutes, and the 4DTs set for all aircraft in $T \sim T + 300$ is Predictions.

$$\text{Predictions} = \left\{\text{Traj}_t^i | t = T+1, T+2, \ldots, T+300; i = 1, 2, \ldots, n\right\}, \quad (4)$$

$$\text{Traj} = (fpl\text{ID}, ac\text{ID}, ac\text{Type}, \text{time}, \text{lng}, \text{lat}, \text{alt}, h\ dg\ , hSp\ d, vSp\ d, \text{TR}), \quad (5)$$

where n is the number of aircraft, Traj_t^i is the state of aircraft i at t, including the call sign (fplID), registration (acID), aircraft type (acType), timestamp (time), longitude (lng), latitude (lat), altitude (alt), speed ($hSp\ d$), heading ($h\ dg$), climb and descent rate ($vSp\ d$), and turn rate (TR). The specific procedure for CD is as follows:

(1) *Initialisation.* The *INITIALISATION* function receives Predictions, establishes an iterative loop, and then extracts the 4DT set locations_t for all aircraft at t from Predictions.

$$\text{locations}_t = \left\{(\text{time}, \text{lng}, \text{lat}, \text{alt})_t^i | i = 1, 2, \ldots, n\right\}. \quad (6)$$

The loop continues from T to $T + 300$ (in fact, only the 4DT set at $T + 300$ needs to be extracted, because the conflicts at $T + 299$ have been detected at $T - 1$).

(2) *Build or Update an R Tree.* In the *BUILDER* function, if the R tree does not exist, a new R tree is built by locations_t according to the above method. Otherwise, all nodes for the existing R tree will be updated, the data locations_{t-1} in the existing R tree will be replaced by the new data locations_t, and the structure will be changed.

(3) *Conflict Detection.* The *DETECTION* function uses the built or updated R tree to search for the surrounding aircraft for the main aircraft and then traverses them to identify conflicts. The status of the conflicting aircraft pairs and the conflict information (the format of these data is mentioned in the simulation section) form a conflict set $\text{conflicts} = \left\{\text{conflict}_{i,j}^t\right\}$, where $\text{conflict}_{i,j}^t$ refers to the conflict between aircraft i and aircraft j detected at t. Finally, conflicts and Predictions are returned as the input of the CR module.

2.3. Conflict Resolution Submodule. Once flight conflicts are detected in advance, corresponding solutions should be automatically generated by the CR module. This section describes how a Markov decision process (MDP) is used for the theoretical modelling of the CR process, and the MCTS algorithm is selected as the solving method of the model. It is worth noting that only the resolution of pairwise conflict in the en route phase is studied in this research, but the MCTS-based CR algorithm can still be extended to the multiaircraft conflict environment. Previous literature has shown that a multiaircraft conflict is composed of multiple pairwise conflicts [28], and the multiaircraft conflict can be disintegrated by resolving all pairwise conflicts one by one. This aspect will be described in future work.

2.4. Modelling Based on an MDP. An MDP is defined as a tuple $\langle S, A, P\rangle$, where S is the set of states of the environment, A is the set of actions that the agent can use to interact with the environment, and P is the transition probability function that defines the probability from one state to another. An MDP addresses problems that can be discretised in time, and the state of the next time step S_{t+1} is related to only the state of the current time step S_t and the selected action A_t. The agent enters S_{t+1} after executing A_t based on S_t, and the transition process $S_t \longrightarrow S_{t+1}$ is up to P.

The process of CR in the real situation is that air traffic controllers formulate the resolution scheme, issue instructions to the aircraft, and perform continuous monitoring until the conflict is completely resolved, and in fact, this process can be discretised into state transitions at each moment. The aircraft in conflict, sector situations, available instructions, and the release time are involved and interact with each other in a complex way. Therefore, with Markov's basic concept, the CR process can be analysed as follows:

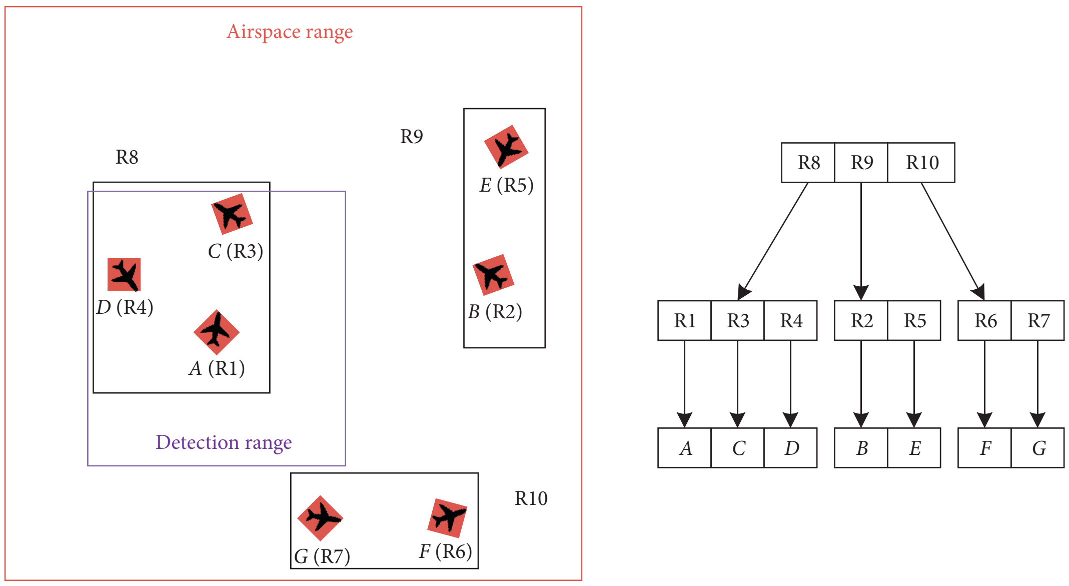

Figure 3: Schematic diagram of conflict detection based on R tree: R1-R7 are the MBRs of aircraft (A–G).

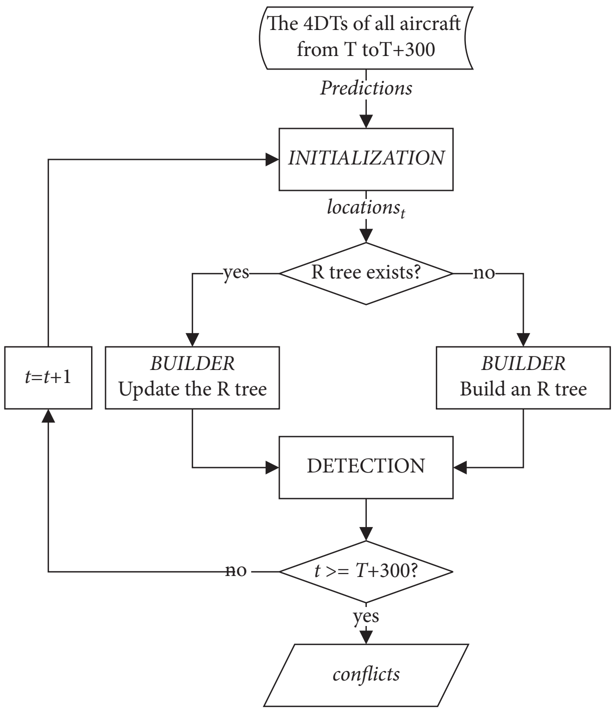

Figure 4: The flowchart of the CD algorithm based on the R tree.

(1) State space S: S refers to the states of all aircraft in the rectangular airspace for which the centre is the midpoint of two conflicting aircraft at the conflict time and for the length, width, and height of 100 km, 100 km, and 1,200 m, respectively:

$$S = \left\{ s_t^i \mid t = T, T+1, \ldots, T+300; i = 1, 2, \ldots, n \right\}, \quad (7)$$

where n is the number of aircraft, T is the current time, and $T+300$ is the conflict time. In this equation, s_t^i represents the state elements that are shown in (5).

(2) Action space A: there are three types of adjustment actions (speed, altitude, and heading) for conflicting aircraft flying in three-dimensional space; that is, $A = \left\{ A_{sp\ d}, A_{alt}, A_{h\ dg} \right\}$.

$A_{sp\ d} = \{-30, -20, -10, 0, +10, +20, +30\}$ represents the speed adjustment range, and the units are knots. The target speed in the action is equal to the current speed plus $A_{sp\ d}$.

$A_{\text{alt}} = \{-600, 0, +600\}$ is the altitude adjustment range, and the units are metres. The reason for this setting is that according to the reduced vertical separation minima (RVSM), the separation between adjacent flight levels (FL) below 12,500 m within the same direction range (true course) is 600 m. The target altitude in the action is equal to the current FL plus A_{alt}. It is worth noting that the upper FL for 8,400 m is 9,200 m, and the difference is 800 m. In this case, it is necessary to correct A_{alt} from 600 m to 800 m. Therefore, the altitude adjustment in this study can also be regarded as changing one FL up or down from the current FL of the conflicting aircraft.

$A_{h\,dg} = \{\text{offset}_{\text{left}}, 0, \text{offset}_{\text{right}}\}$ refers to the heading adjustment range. The heading adjustment in this study adopts a fixed offset mode (Dogleg manoeuvre). An offset manoeuvre is a common instruction in real CR. The term $\text{offset}_{\text{left}}$ represents the left offset (Figure 5), and $\text{offset}_{\text{left}}$ represents the right offset:

(1) *Action Limitations.* All of the adjustments require consideration of the aircraft's performance limitations and the terrain. A_{alt} needs to meet the requirement for the minimum safe altitude of the route. When A_{alt} is higher than 12,500 m or lower than 6,000 m, the target altitude takes the value of 12,500 m or 6,000 m. $A_{sp\,d}$ should also meet the requirements of the flight performance envelope of the aircraft. The speed limits for different configurations and the FL are different. For example, the descending aircraft has minimum, nominal, and maximum descending true airspeeds in BADA, and the target speed needs to be within this range.

(2) *Execution Time.* As described above, the collision detection advance time in this study is 5 minutes, that is, 300 seconds. Assuming that the current time is T, the conflict time is $T + 300$. Because the execution of the conflict relief action requires a specific duration, the action execution time in this study is set to $(T + 30, T + 150, T + 270)$; that is, three actions need to be executed from T to $T + 300$, with an interval of 120 seconds between adjacent actions (Figure 6).

(3) *Execution Object.* For this study, only one of the two conflicting aircraft executes one of the above actions at each execution time.

(4) *Resolution Effect.* The primary goal of CR is to resolve the pairwise conflict that will occur at $T + 300$ by executing actions within $T \sim T + 300$. However, with consideration of the Domino effect, it should be ensured that there will be no conflict between the two conflicting aircraft and other aircraft in the CR process. In addition, the effect of the action should continue to play a role after $T + 300$, and there should be no other conflict involving two conflicting aircraft at $T + 300 \sim T + 360$.

(3) Transition probability function P:

$$P_{s,s'} = P[s_{t+1} = s' | s_t = s]. \tag{8}$$

In this study, the transition function P is encapsulated in the BADA-based 4DTP module aforementioned. It is worth noting that the transition function P of the MDP is deterministic. If a certain state s is given, an action a imposed by the controller is the only one outcome. That is, a specific case of MDP is used in this paper, which is simpler than a general MDP. Thus, P would be a *one-hot* matrix representing a pair $(s, a \longrightarrow s')$. However, the randomness is still present by having several possible actions at each moment, and this is the reason that the algorithm of conflict resolution used is a tree search algorithm.

2.5. Conflict Resolution Algorithm.

In the CR process, the number of aircraft and the motion parameters can change at any time, and the states built by these parameters are also different, which makes the state space very large. Moreover, all the groups of instructions that the controller can use make the action space very large, too. Therefore, the MCTS algorithm is chosen as the solution algorithm for CR.

2.5.1. Monte Carlo Tree Search.

The MCTS is a search algorithm based on a tree structure, which is effective for decision-making in a large search space [29]. MCTS considers both exploration and utilisation by setting the configuration of the weights, which can achieve a more heuristic method to avoid falling into the local optimal solution and to solve the problem of the value of all subtrees not being able to be calculated due to the huge search space.

The MCTS algorithm has three important characteristics: (1) *Heuristic.* One of the most significant advantages of the MCTS is that it does not require domain knowledge, making it easy to apply to any field that can use tree modelling. (2) *Anytime.* The MCTS will immediately backpropagate the results of each game to ensure that all values are always up to date after each iteration of the Algorithm 3) *Asymmetric.* Tree selection allows the algorithm to favour more potential nodes (the selection probability of other nodes is not allowed to converge to zero), in order to form an asymmetric tree over time.

As shown in Figure 7, each iteration of the MCTS algorithm has four steps: *selection*, *expansion*, *simulation*, and *backpropagation*. The first step is to select the most worthwhile child node of the root node in the tree. The default policy is to choose the unexplored child nodes preferentially. If all the child nodes have been explored, the child node with the largest reward value is selected. The second step is expansion. A new child node is created for the previously selected child node, and an action is performed at random. The third step is simulation; that is, the newly expanded child node starts to simulate the game until the end or until a specified number of steps and then obtains the score after the simulation (usually 1 for success and 0 for

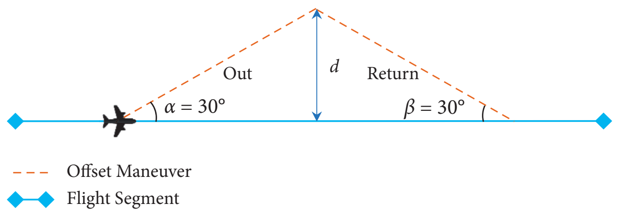

Figure 5: The diagram of the left offset: there are two "Out" and "Return" stages.

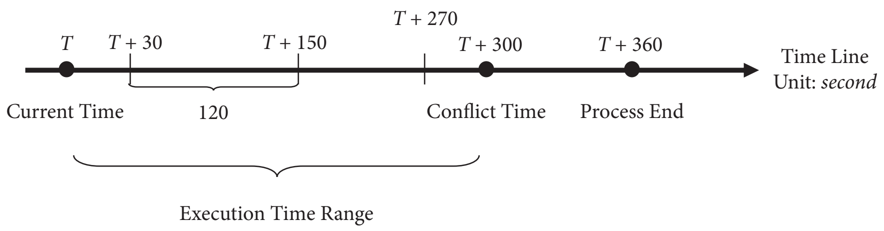

Figure 6: The diagram of the action execution time: the timeline is not proportional.

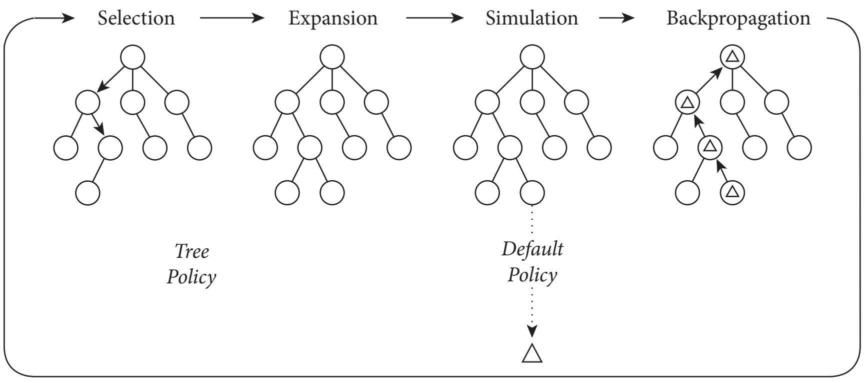

Figure 7: The search procedure diagram for the MCTS algorithm [29].

failure). The fourth step is backpropagation. The scores of the newly extended child nodes are fed back to all the previous parent nodes, and the number of successes and visits for these nodes are updated for the calculation of the reward value. In the end, the actions corresponding to the child node with the largest reward value in each layer are selected as the resulting strategy.

In the *selection* step, the selection of child nodes is based on the reward value. The main difficulty of this step is to maintain a certain balance between the utilisation of effective update with multiple iterations and the exploration of a few simulated movements after the average success rate is high. For this reason, the classic upper confidence bounds (UCB) formula is used to expand the minimax search and to obtain a game tree search algorithm: UCB applied to Trees (UCT):

$$\mathrm{UCT} = \frac{w_i}{n_i} + C_p\sqrt{\frac{\ln n}{n_i}}, \tag{9}$$

where w_i is the number of times the simulation result succeed after the i_{th} iteration; n_i is the number of simulation times after the i_{th} iteration; C_p represents the exploration parameter, which is a constant greater than or equal to zero, and its specific value is usually selected by experience; and n is the total times of iteration, which is equal to the sum of all n_i. In (9), the part before the plus sign is used for exploitation, the latter part is used for exploration, and C_p is used to balance the two terms.

2.5.2. Hypotheses. As mentioned above, the actual CR process is very complex, and the controller needs to consider various factors that have different influences on the decision-making process. Hence, to establish an abstract model that can clearly describe the CR process, several hypotheses are proposed, as follows:

(1) It is assumed that upper-air wind, navigation errors, and other meteorological factors are not considered
(2) It is assumed that the ground-air communication and aircraft operation response times can be ignored
(3) Emergencies, airspace restrictions, and transfer of control responsibilities are not considered
(4) Regardless of the priority, all conflicting aircraft may be required to execute resolution actions

(5) It is assumed that all aircraft, except those aircraft executing the action to resolve conflict, fly point-by-point according to the flight plan

It is believed that DST is a system/project biased towards engineering applications. Air traffic attaches great importance to safety. If the above assumptions, especially the wind uncertainty, are not considered, the reliability and practicability of DST will become low. However, for example, the uncertainty is not only the focus but also the difficulty, which needs to be studied in combination with appropriate methods (such as learning uncertainty by using machine learning methods). Therefore, only after a thorough analysis of the essence of DST or CDR, these assumptions and wind vector will be taken into account in the next work.

2.5.3. Solution Algorithm. CR is the decision-making process of the air traffic controllers. The conflicting aircraft changes its state to avoid collision by executing one or more instructions sent by the controllers. As mentioned above, the resolution policy in this study is an action sequence in which there are three actions, and T_1, T_2, and T_3 are their execution times (Figure 6). At each execution time, one of the n actions in the action space is executed by one of two conflicting aircraft until the CR process is completed. We represent this series of decisions in the form of a tree (Figure 8). Each node of the tree represents the state set of all aircraft at that time. Because the depth h of the tree and the number of nodes in each layer is n^h, resulting in a search space that is too large, the MCTS algorithm is selected to solve for the CR process. The search steps of the MCTS-based CR algorithm are as follows:

(1) *Initialisation.* It is assumed that the current time is T and the conflicting time is $T + 300$. At $T_1 = T + 30$, the state S_{T_1} of the aircraft is the root node v_0, where S_{T_1} represents the state of all aircraft (including conflicting aircraft and environmental aircraft). The termination condition of the search process is that the iteration number reaches the specified number. Each iteration consists of four steps: selection, expansion, simulation, and backpropagation. At the end of this function, the optimal action sequence a including the action with the largest UCT value in each layer is returned (in Figure 8, the action sequence is $a = \{a_2, a_2, a_n\}$).

(2) *Selection and Expansion.* If all child nodes of v_0 have been explored or visited, the node is considered a fully extended node. Then, the child node that has the largest UCT value of v_0 is selected as the new root node, and then, it is judged whether the new root node is fully extended. If the answer is yes, the new root node is selected, and the loop continues. Otherwise, the function will randomly select one from the child nodes of v_0 that is never visited as v, and the simulation steps for v are performed.

(3) *Simulation.* All the child nodes of each layer below v will randomly execute an action to adjust the state of the aircraft. If the last action leads to new conflicts, the simulation is terminated and Failed is returned with a reward of 0. If no new conflicts appear and the termination conditions are met, Solved with a reward of 1 is the result, and the reward is returned and assigned to Δ in the backpropagation step.

(4) *Backpropagation.* All the parent nodes of the node v update the total visited times by adding 1, they update the success times by adding Δ, and then, the UCT formula is used to recalculate and update the UCT values. If the end conditions are not met, the algorithm returns to step 2.

2.6. Simulation

2.6.1. Scenarios

(1) Data Sources. The basic database of the simulation environment includes the navigation database, and flight plans from the civil aviation airspace centre of China, and the date for these data is June 1, 2018. The version of BADA is 3.12.

(2) Scenario Location and Size. The shape of the scenario airspace is a cuboid for which the centre is the location of a waypoint that is randomly chosen from the waypoint set of China's airspace, and both the length and the width are 200 km. Because the scope of this research is the en route phase, the maximum allowed altitude of the scenario is 12,000 m and the minimum value is 6,000 m.

(3) Hardware. The hardware consists of an HP Z840 workstation, an Intel Xeon(R) CPU E5-2630 V3 @ 2.4 GHz, 32 GB RAM, and an operating system of Windows 7.

(4) Software. The software consists of the unopened air traffic operation simulation system (ATOSS) based on BADA and developed in the laboratory of the authors, the Python programming language, and its integrated development environment (IDE) PyCharm. The R tree API is from https://pypi.python.org/pypi/Rtree/, and the link for the MCTS API is https://github.com/pbsinclair42/MCTS.

(5) Construction Process. (1) A scenario airspace range (Figure 9) is randomly selected. (2) Several flight plans from all flight plans that fly through the selected airspace scenario range are selected. (3) To obtain more scenario samples, random changes to the start time, aircraft type, start altitude, and target altitude in the flight plan are added. (4) The simulation is run, and conflicts are detected. The aircraft appears at the airspace boundary at the start time and the start altitude and finally maintains the target altitude. The terminating condition of the scenario is that all aircraft fly out of the airspace range. (5) The information for the detected pairwise conflicts in the scenario is screened and recorded. The pairwise conflicts for which the conflicting aircraft has flown out of the scenario airspace before the CR process is over are not recorded in order to facilitate the differentiation of the flight time

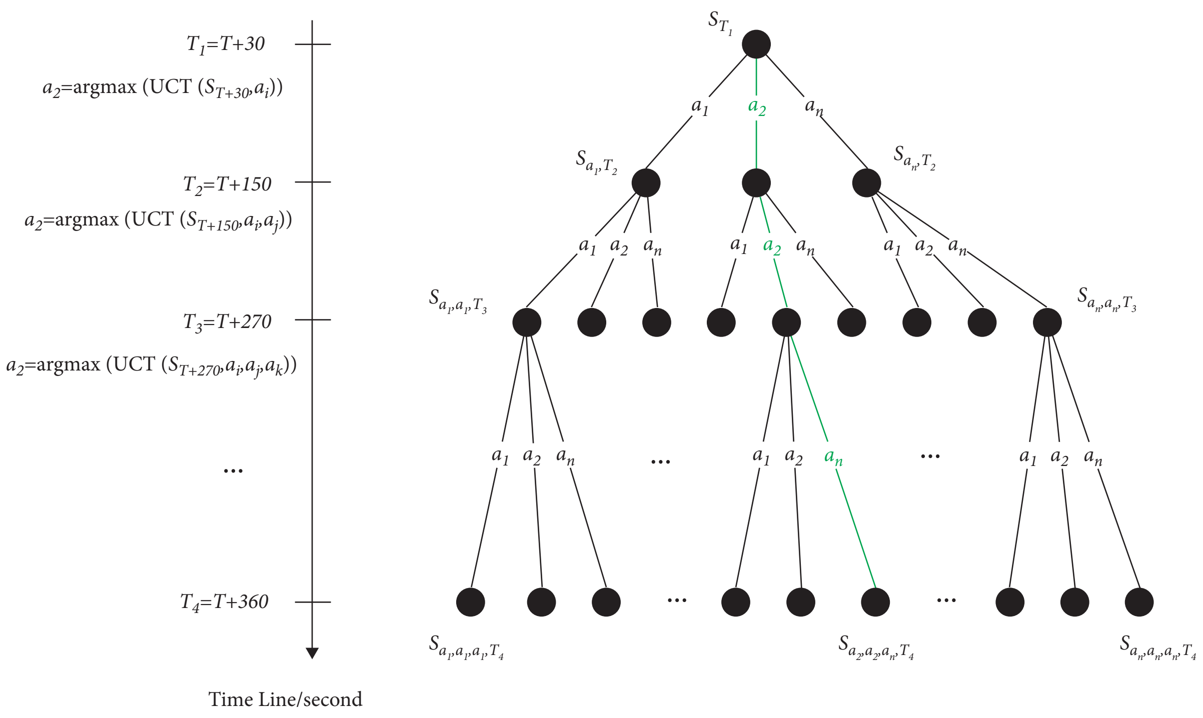

Figure 8: The tree search structure of the conflict resolution based on the MCTS.

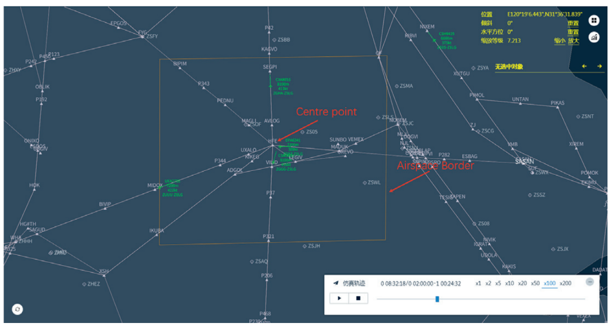

Figure 9: The scenario range shown at the interface of ATOSS.

within the scenario before and after CR. (6) All of the scenario information is stored in the database (the format of the scenario sample is shown in Table 1 and Table 2). It is worth emphasising that the focus of this research is the exploration of CR algorithms, and the overall resolutions of all conflicts in the sector are not studied. Therefore, there is at least one pairwise conflict in each scenario, but only the first pairwise conflict is used.

2.7. Conflict Detection Simulation. Three experiment factors for the CD simulation are the [7] *efficiency, flexibility,* and *completeness*, which are the requirements for the CD algorithms proposed in a published paper by Isaacson and Erzberger [30]. In addition, the aforementioned pairwise algorithm is taken as the control group, and the R tree algorithm is the experimental group.

The *efficiency* E_t is the indicator that shows the ability to identify the conflicts from a huge number of trajectories in real time. It is assumed that the mean computing times of the experimental group and the control group are $\overline{T}_e$ and $\overline{T}_c$, respectively. The relationships between E_t, $\overline{T}_e$, and $\overline{T}_c$ are shown as follows:

$$\overline{T}_e = \frac{1}{n}\sum_{i=0}^{n} T_e^i,$$
$$\overline{T}_c = \frac{1}{n}\sum_{i=0}^{n} T_c^i, \tag{10}$$
$$E_t = \frac{T_c - T_e}{T_c} \times 100\%.$$

The *flexibility* is the ability to process the whole set of trajectories when the states of the aircraft are updated. It is acceptable that the processing time is within 10 seconds (the update cycle of the radar is 12 seconds).

Table 1: The scenario information for the pairwise conflicts (partial).

Scenario ID	Conflicting aircraft	Conflict time (UCT)	Horizontal distance (m)	Vertical distance (m)
1	JYH1107-CSN6590	00 : 26 : 22	9935.6	0.0
2	CES9905-CCA4506	01 : 46 : 48	9324.2	0.0
3	CSZ9870-CHB6219	01 : 46 : 56	9998.8	0.0
4	CSN6263-HBH3322	01 : 46 : 00	1791.7	290.7
5	LKE9861-CES2067	01 : 56 : 08	9990.6	0.0
6	CCA4344-CCA4540	00 : 35 : 34	1406.2	298.2
7	CSN6786-CSS6920	00 : 57 : 30	3100.8	299.9
. . .	. . .	. . .		
N	CSN3198-CCA8258	01 : 10 : 12	9819.6	141.3

Table 2: The data form of the flight plan in a scenario (partial).

Callsign	Registration	Aircraft type	Start time (UCT)	Route	Initial level (m)	Target level (m)
JYH1107	B2182	A330	00 : 11 : 59	ZBSJ-ZGGG	6900	6900
CCA4141	B5426	CRJ7	00 : 26 : 57	ZUGY-ZSYN	7200	7800
CHH7191	B7618	CRJ7	00 : 17 : 11	ZHNY-ZGGG	8900	8900
UEA6675	B3386	B744	00 : 29 : 32	ZUXC-ZSWX	10400	8400
CQH8843	B1895	B737	00 : 08 : 46	ZBYN-ZSCN	8100	8100
. . .	. . .	. . .	. . .	. . .	. . .	
CSN6590	B6790	A340	00 : 11 : 59	ZSCG-ZGHA	6900	6900

The *completeness* is the accuracy. In theory, the conflicts detected by the pairwise algorithm are the most comprehensive for the same conditions, and the accuracy is set to a relative 100% in this research. It is necessary to have few false and missing alarms as possible in the conflicts detected by the nonpairwise algorithms. Therefore, completeness can also be regarded as the consistency of the conflicts detected with the two methods.

The simulation for CD includes eight experiments (Table 3). First, 10,000 different scenarios (the centre point of the scenario, the flight plans, and the information about the conflict are different) are selected according to the number of aircraft in the scenario for each experiment. Then, the R tree algorithm and the pairwise algorithm are used to detect conflicts and record the time consumptions and the information for the conflict. Finally, the relationships between the mean computing time/efficiency E_t and the number of aircraft in the scenario are shown in an integrated illustration, and the data for flexibility and completeness are collected.

The experiment result is shown in Figure 10. When the number of aircraft in the scenario increases, the mean computing time of the control group becomes much longer than that of the experimental group, and the *efficiency* is more obviously improved. In addition, the mean computing time of detecting conflicts in the 80-aircraft scenarios is 1.78 seconds (including the processing time for updating trajectories, that is, the *flexibility*). Through the comparison of the conflict information of the two groups, it is reported that the conflicts of the experimental group and the control group are the same, and the *completeness* is 100%.

2.8. Conflict Resolution Simulation. The CR module in the DSTs shares part of the CR work of the controllers and reduces their workload by automatically providing effective resolution suggestions. The experiments described in this section mainly verify the CR algorithm based on two aspects: *success rate* and *mean computing time* [31]. From the perspective of actual operation, the resolution actions need to not only be able to successfully free the conflict but also avoid excessively affecting the normal flight of the aircraft. Therefore, the statistics and the analysis of the flight time difference (*delay*) in the airspace before and after the actions are executed are determined.

A genetic algorithm (GA) is a random global search and optimisation method evolving from the mechanism of biological evolution. A GA abstractly represents a certain number of initial solutions as the chromosomes of the current population. To make the population evolve towards a better solution, the iteration begins with three steps: (1) fitness calculation of individual, (2) natural selection, and (3) gene recombination and mutation. The reason why a GA is used as the control group is that GAs are the earliest and most widely used type of algorithm for the CR problem. Together with the MCTS algorithm of the experimental group, they belong to the category of heuristic search algorithms.

2.8.1. Success Rate and Mean Computing Time. The *success rate* (SR) refers to the percentage of conflict scenarios successfully resolved:

$$\mathrm{SR} = \frac{\mathrm{scenario}_{\mathrm{solved}}}{\mathrm{scenario}_{\mathrm{total}}} \times 100\%, \tag{11}$$

where $\mathrm{scenario}_{\mathrm{solved}}$ is the number of scenarios successfully resolved, and $\mathrm{scenario}_{\mathrm{total}}$ is the number of total scenarios. The *mean computing time* (MCT) refers to the average value of the computing time (CT) required to search for effective resolution actions for n scenarios:

Table 3: The design of the conflict detection experiment.

Experiment ID	R tree	Pairwise algorithm	The number of aircraft per scenario	The number of scenarios
1	√	√	10	10,000
2	√	√	20	10,000
3	√	√	30	10,000
4	√	√	40	10,000
5	√	√	50	10,000
6	√	√	60	10,000
7	√	√	70	10,000
8	√	√	80	10,000

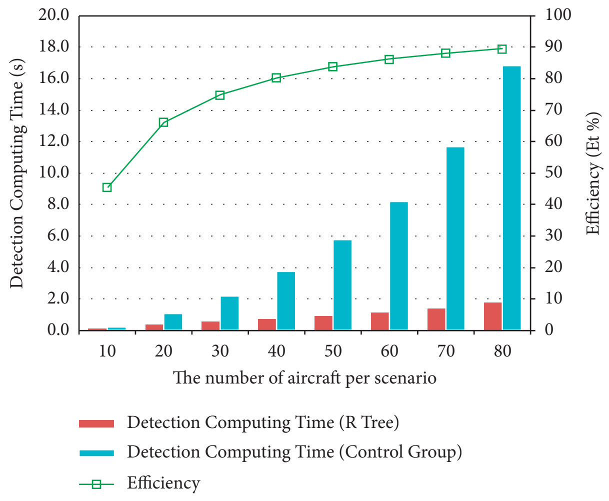

Figure 10: The relationship of mean computing time/efficiency and the number of aircraft per scenario.

$$\mathrm{MCT} = \frac{1}{n}\sum_{i=0}^{n} \mathrm{CT}_i, \qquad n = \mathrm{scenario}_{\mathrm{solved}}. \tag{12}$$

As shown in Table 4, there are nine experiments. Experiments 1–8 are designed to test the relationship between the SR/MCT and the iteration times, which are parameters of the MCTS. The other parameters are the same, such as 10,000 conflict scenarios with 80 aircraft (10,000 scenarios are different). Experiment 9 is designed to compare the SR and the MCT of the GA and the MCTS for the same batch of conflict scenarios that are randomly selected from the scenarios of Experiment 4. The parameter settings of the GA include the settings that the individual number of populations is 20, the number of generations for evolution is 100, the probability of gene recombination is 0.5, the mutation probability is 0.01, and the fitness function is the sum of squares of the distance between the conflicting aircraft and other aircraft. The number of iterations for the MCTS is 40,000, and an iteration occurs after the four steps shown in Figure 7 are completed.

There are two curves in green and blue in Figure 11. The figure shows the changes of the SR and the MCT with the increase in the number of iterations. Both the MCT and the SR of the MCTS rise gradually with the increase of the number of iterations. The MCT is limited to 11 seconds, and SR reaches about 95% when the number of iterations is 45,000.

The CT for the two groups is collected and represented in the form of a histogram (Figure 12). For the same scenarios, the CT of the MCTS is mostly concentrated within 20 seconds, while that of the GA is mainly distributed between 20 seconds and 90 seconds. The MCT of the MCTS is 12.44 seconds, and its SR is 94.93%. The MCT of the GA is 54.37 seconds, and its SR is 86.65%. For the experimental conditions in this research, the MCTS gives the resolution policy faster than the GA does, and the SR of the former is also higher. Therefore, the experiment group can also better meet the requirements of real-time function and safety.

2.8.2. Delay. The above experiments verify the safety performance of the MCTS algorithm. However, the quality of the actions should be also considered, including the flight time in the airspace and the fuel consumption, which are the indicators of operational efficiency. In this research, the CR is mainly studied from the perspective of air traffic control units, so the flight time is selected, and the fuel consumption will be examined in future work.

The *delay* is defined as the difference between the planned flight time T_p and the actual flight time T_a required to arrive at the final waypoint in the scenario (Figure 13). The mean delay $\overline{\mathrm{delay}}$ is the average value of the delays in these

Table 4: The design of the success rate and mean computing time for CR.

Experiment ID	MCTS	GA	The number of iterations of the MCTS/ 1e4	The number of aircraft per scenario	The number of scenarios
1	√		1	80	10,000
2	√		2		
3	√		3		
4	√		4		
5	√		5		
6	√		6		
7	√		7		
8	√		8		
9	√	√	8	80	2,000

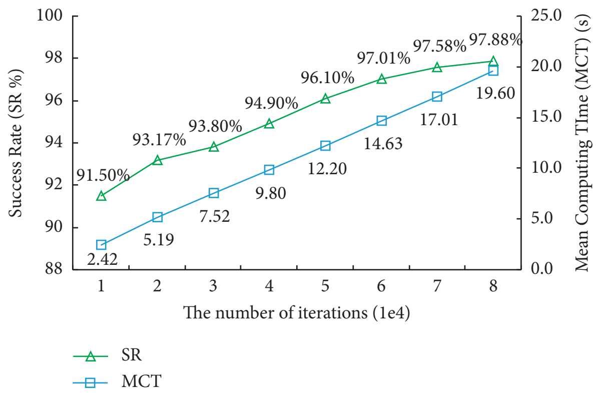

Figure 11: The change of the success rate and the mean computing time with the increase of the number of iterations.

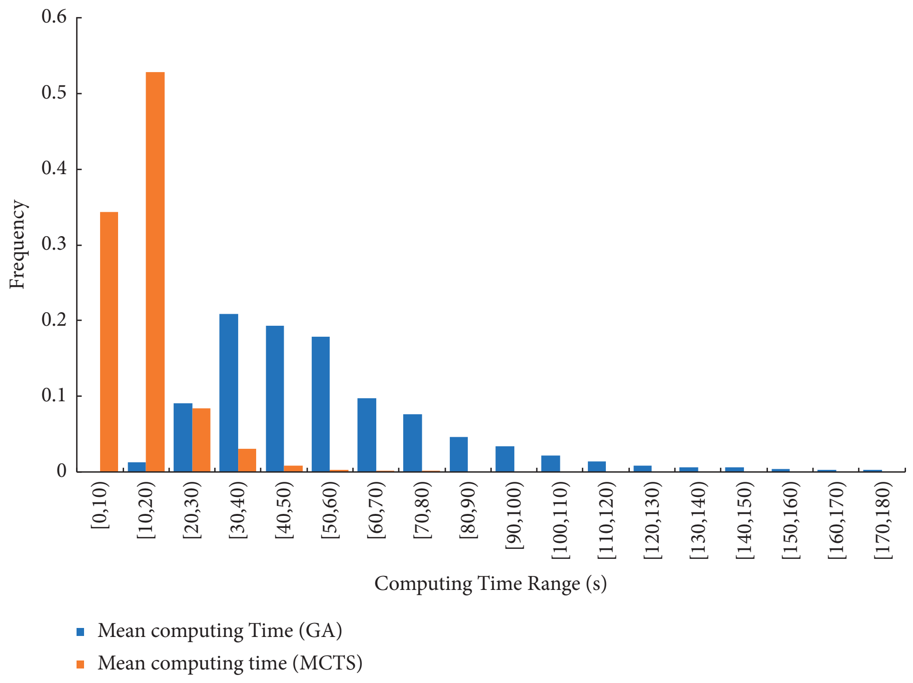

Figure 12: The histogram of computing time. The blue bar is the computing time of the GA, and the yellow bar is the computing time of the MCTS.

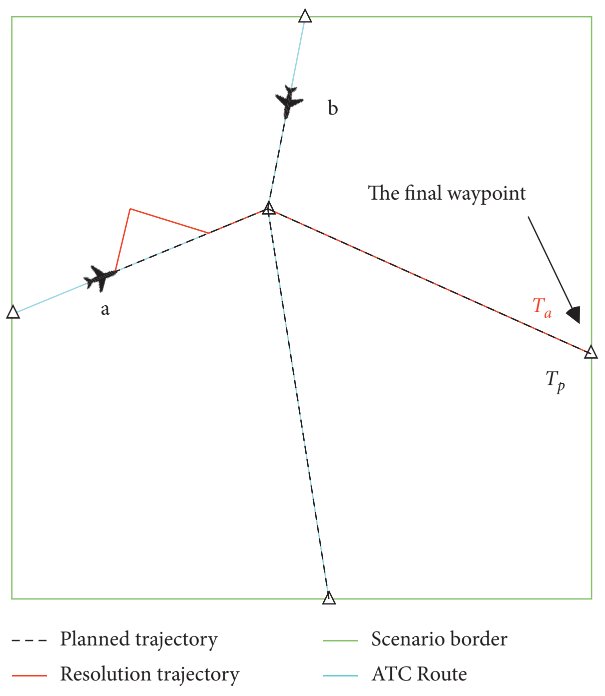

Figure 13: Schematic diagram of delay.

Table 5: The design of delay for CR.

Experiment ID	The number of aircraft per scenario	The number of scenarios
1	40	
2	60	10,000
3	80	

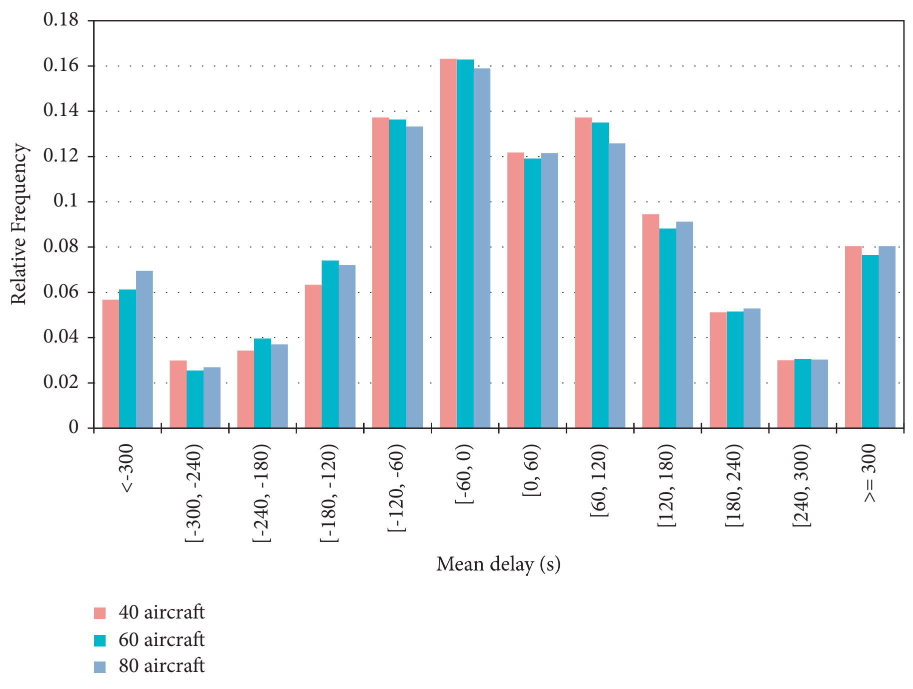

Figure 14: The distribution of delays: "40 aircraft" means that there are 40 aircraft in a scenario.

successfully solved scenarios. The formulas for $de\ lay$ and $\overline{\text{delay}}$ are shown as follows:

$$\begin{aligned}\text{delay} &= T_a - T_p,\\ \overline{\text{delay}} &= \frac{1}{n}\sum_{i=0}^{n}\text{delay}_i,\\ n &= \text{scenario}_{\text{solved}}.\end{aligned} \tag{13}$$

As shown in Table 5, there are three experiments. The purpose of this simulation is to obtain the distribution of the delay after testing a certain number of scenarios and the change of the delay with the increase of the number of aircraft in the scenario. It is worth noting that the method for choosing conflict scenarios in this simulation is different. Experiment 1 selects 10,000 scenarios with 40 aircraft (10,000 scenarios are different), and Experiments 2 and 3 use the scenarios of Experiment 1, but 20 and 40 environmental aircraft are randomly placed in each scenario without changing the original conflicts.

Figure 14 shows the histogram of the delays. For the 40-aircraft scenarios, delays of ± 1 minutes account for nearly 29% of the total delays, and more than 85% of the delays are within ± 5 minutes. Therefore, it is shown that the MCTS-based CR algorithm has a small influence on the normal operation of the flights. In addition, the traffic density in the scenario increases as the number of aircraft becomes greater. However, the delays distributions of the scenarios in which the number of aircraft is different are similar, which proves that the MCTS can still search for a high-quality solution even if the traffic is more crowded.

3. Conclusions

With the aim of reducing the workload of air traffic controllers and improving the capability of solving the conflicts in en route TBO, this paper proposes a CDR module based on shared 4DTs to help develop a DST.

Real-time use and accuracy are the key goals of conflict detection (CD). The CD module needs to be able to process the 4DTs of all aircraft as quickly as possible and to identify the conflicts without missing or false alarms. Pairwise comparison is the most accurate but least efficient CD method, and determining how to shorten the detection time with the condition of ensuring accuracy is the focus of this study. Therefore, the R tree, which is simple and easy to implement, and which has a lower-time-complexity spatial data structure, is proposed. The simulation results prove that the computing time of the R tree shows a logarithmic curve with the number of aircraft, and the conflicts detected by the pairwise algorithm are the same. In the conflict resolution (CR) module, in order to solve the problem of a huge search space, the Monte Carlo tree search (MCTS) algorithm is selected to search for a feasible resolution policy. The simulation results show that the MCTS-based conflict resolution method has high efficiency and a high success rate, and it requires less computing time in heuristic search algorithms. R tree and MCTS can be combined to provide technical support for a future auxiliary DST.

However, there are still many deficiencies in this study. For example, the uncertainty (especially that due to wind) is not considered in the CD process, so the practicability of the R tree-based CD algorithm needs to be tested and improved. Second, the constraints of the controller's acceptable level, fuel consumption, and uncertainty (such as high-altitude wind and pilot operation), which make the CR algorithms more practical, also need to be included. In addition, the execution time of actions in this study is fixed, which is incompatible with the CR process of real sector airspace operation. Finally, the CR in this study is based on a single scenario and does not consider more complex situations such as multiaircraft and continuous conflicts. The above-mentioned issue is worth studying further in future work.

Data Availability

The scenarios are randomly generated in Python and stored in the MongoDB database, and the relevant codes for this method can be found at the following link: https://github.com/Lydia-Yahuhe/cdr_mcts.git. However, the detailed data (such as flight plans, ATS route, and waypoints) used to support the findings of this study are available from the corresponding author upon request.

Disclosure

This work has been jointly completed by the project team of the Intelligent Air Traffic Control Laboratory of the Nanjing University of Aeronautics and Astronautics (NUAA), and the content cannot be separated from the efforts of every member.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work has been supported by the Safety Ability Project of the Civil Aviation Administration of China (No. TM 2018-5-1/2) and the Open Foundation Project of the Graduate Student Innovation Base (Laboratory) of the NUAA, China (No. kfjj20190720). The authors also thank LetPub (http://www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

References

[1] Faa, *NextGen Implementation Plan*, Federal Aviation Administration, Washington, D.C., USA, 2016.

[2] Sesar, *European ATM Master Plan*, Eurocontrol, Brussels, Belgium, 2019.

[3] ICAO, *The Aviation System Block Upgrades*, International Civil Aviation Organization, Montreal, Canada, 2016.

[4] M. Konyak, D. Warburton, J. Lopez-Leones, and P. Parks, "A demonstration of an aircraft intent interchange specification for facilitating trajectory-based operations in the national airspace system," in *Proceedings of the AIAA Guidance,*

Navigation, and Control Conference and Exhibit, p. 7145, HI, USA, August 2008.

[5] ATMB, *The concept of Trajectory-Based Operation (TBO) for air traffic management of civil aviation of China*, Civil Aviation Administration of China, Beijing, China, 2019.

[6] ICAO, *Procedures for Air Navigation Services - Air Traffic Management*, International Civil Aviation Organization, Montreal, Canada, 16th edition, 2016.

[7] S. Ruiz and M. A. Piera, "A medium term conflict detection and resolution system for terminal maneuvering area based on spatial data structures and 4D trajectories," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 396–417, 2013.

[8] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[9] X. Guan, X. Zhang, and R. Lv, "A large-scale multi-objective flights conflict avoidance approach supporting 4D trajectory operation," *Science China Information Sciences*, vol. 60, no. 11, Article ID 112202, 2017.

[10] Y. Jen, J. T. Klosowski, and J. S. B. Mitchell, "Geometric algorithms for conflict detection/resolution in air traffic management," in *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 1835–1840, CA, USA, December 1997.

[11] J. Goss, R. Rajvanshi, and K. Subbarao, "Aircraft conflict detection and resolution using mixed geometric and collision cone approaches," in *Proceedings of the AIAA guidance, navigation, and control conference and exhibit*, p. 4879, Rhode Island, USA, August 2004.

[12] M. Jardin, "Grid-based strategic air traffic conflict detection," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, CA, USA, August 2005.

[13] N. L. Fulton, "Airspace design: towards a rigorous specification of conflict complexity based on computational geometry," *Aeronautical Journal*, vol. 103, no. 1020, pp. 75–84, 1999.

[14] S. Alam, K. Shafi, H. A. Abbass, and M. Barlow, "An ensemble approach for conflict detection in Free Flight by data mining," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 3, pp. 298–317, 2009.

[15] N. Durand, J. M. Alliot, and J. Noailles, "Automatic aircraft conflict resolution using genetic algorithms," in *Proceedings of the 1996 ACM symposium on Applied Computing*, pp. 289–298, PA, USA, March, 1996.

[16] X. Guan, X. Zhang, D. Han, Y. Zhu, J. Lv, and J. Su, "A strategic flight conflict avoidance approach based on a memetic algorithm," *Chinese Journal of Aeronautics*, vol. 27, no. 1, pp. 93–101, 2014.

[17] R. K. Cecen and C. Cetek, "A two-step approach for airborne delay minimization using pretactical conflict resolution in free-route airspace," *Journal of Advanced Transportation*, vol. 2019, Article ID 4805613, 17 pages, 2019.

[18] P. Han, G. Liu, H. Niu, X. Yang, and C. Ma, "Research on civil aircraft conflict resolution based on TBO,"vol. 1, pp. 52–56, in *Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 1, IEEE, Chengdu, China, December 2019.

[19] N. Durand and J. M. Alliot, "Ant colony Optimization for Air Traffic Conflict Resolution," in *Proceedings of the Atm seminar*, CA, USA, July 2009.

[20] P. K. Menon, G. D. Sweriduk, and B. Sridhar, "Optimal strategies for free-flight air traffic conflict resolution," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 202–211, 1999.

[21] Z. Liu, K. Cai, X. Zhu, and Y. Tang, "Large scale aircraft conflict resolution based on location network," in *Proceedings of the 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, pp. 1–8, IEEE, St. Petersburg, FL, USA, September 2017.

[22] G. J. Pappas, C. Tomlin, and S. S. Sastry, "Conflict resolution for multi-agent hybrid systems," in *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 1184–1189, Kobe, Japan, December 1996.

[23] C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, S. Sastry, and G. Meyer, "Hybrid control in air traffic management systems 1," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 5512–5517, 1996.

[24] Z. Wang, H. Li, J. Wang, and F. Shen, "Deep reinforcement learning based conflict detection and resolution in air traffic control," *IET Intelligent Transport Systems*, vol. 13, no. 6, pp. 1041–1047, 2019.

[25] A. Nuic, *User Manual for the Base of Aircraft Data () Revision 3. 12*, Eurocontrol, Brussels, Belgium, 2014.

[26] A. Guttman, "R-trees," in *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pp. 47–57, MA, USA, June 1984.

[27] A. Guttman, "R-tree," 1984, https://en.wikipedia.org/w/index.php?title=R-tree&oldid=1039468457.

[28] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Review of conflict resolution methods for manned and unmanned aviation," *Aerospace*, vol. 7, no. 6, p. 79, 2020.

[29] C. B. Browne, E. Powley, D. Whitehouse et al., "A survey of Monte Carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[30] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997.

[31] T. Farley, M. Kupfer, and H. Erzberger, "Automated conflict resolution: a simulation evaluation under high demand including merging arrivals," in *Proceedings of the 7th AIAA ATIO Conf, 2nd CEIAT Int'l Conf on Innov and Integr in Aero Sciences, 17th LTA Systems Tech Conf; followed by 2nd TEOS Forum*, Belfast, Northern Ireland, September 2007.