Hindawi
Journal of Advanced Transportation
Volume 2018, Article ID 5042961, 18 pages
https://doi.org/10.1155/2018/5042961

Research Article

A Causal Model for Safety Assessment Purposes in Opening the Low-Altitude Urban Airspace of Chinese Pilot Cities

Jun Tang [1,2] **and Wenyuan Yang** [1]

[1] *College of Systems Engineering, National University of Defense Technology, Changsha, China*
[2] *Technical Innovation Cluster on Aeronautical Management, Universitat Autonoma de Barcelona, Sabadell, Spain*

Correspondence should be addressed to Jun Tang; jun.tang@e-campus.uab.cat

Received 20 March 2018; Revised 6 September 2018; Accepted 19 September 2018; Published 10 October 2018

Academic Editor: Shamsunnahar Yasmin

China has been gradually relaxing its ban on the use of low-altitude airspace across the country. To guarantee the high reliability of air traffic management (ATM), conflict detection and conflict resolution (CDR) approaches are indispensable to maintain safe separation between neighbouring small fixed-wing aircraft. In this study, we analyse a temporal and spatial integrated strategy for safety assessment purposes in opening the low-altitude urban airspace of Chinese pilot cities. First, we present a detailed mathematical description of the proposed algorithms based on a spatial grid partitioning system (SGPS). For our system, a conflict detection (CD) algorithm is designed to determine if two trajectories pass through the same grid space within overlapping time windows. A conflict resolution (CR) algorithm integrates a proposed time scheduling-based technique (TST) and vertical change-based technique (VCT), which operate under predetermined basic principles. Then, based on our novel CDR algorithms, a causal model is constructed in graphical modelling and analysis software (GMAS) to generate a state space that can provide a global perspective on scenario dynamics and better understanding of induced conflict occurrences. Finally, simulation results demonstrate that the proposed approach is practical and efficient.

1. Introduction

The Chinese State Council and military authorities jointly decided to open the country's low-altitude airspace to general aviation (GA) in November 2010 [1]. China began to allow private planes to enter low-altitude airspace below 1,000 m without military approval in 2015 [2]. The country will continue its reform of airspace management to boost the aviation industry. China has launched pilot projects in its north-eastern and south-central regions, as well as 10 pilot cities: Hainan, Changchun, Guangzhou, Tangshan, Xi'an, Qingdao, Hangzhou, Ningbo, Kunming, and Chongqing. It is expected that China's total number of general aircraft will surpass 5,000 by 2020 with an annual growth rate of approximately 19% and that potential aviation market demand will reach 15.5 billion USD, which would represent a new economic engine according to the Civil Aviation Administration of China (CAAC) [3].

GA in China has been developing at a fast pace with new players joining the market and local governments becoming increasingly involved. It has a large market potential driven by state and local economic development plans, public demand for business jets, and the need for public services and individual recreation [4]. Recent GA progress has been accelerated by loosening regulations on low altitude airspace that enables the safe and efficient operation of small fixed-wing aircraft, as well as simplified permit procedures for general and business aviation operations. Currently, the balance between airspace capacity and customer demand is maintained by air traffic management (ATM) using a series of decision support tools that are applied to ATM system and ensured by advanced communications and navigation technology [5–7]. However, as a result of the current policy of gradually opening low-altitude airspace in China, potential uncertainties and perturbations may occur in low-altitude airspace traffic management because of increasing airspace density. Each small fixed-wing aircraft must follow its planned four-dimensional (4D) trajectory, which can be defined by a series of sequential waypoints recorded as three-dimensional (3D) spatial information with corresponding timestamps

[8–12]. Time and space deviations in one trajectory may affect other 4D trajectories because of the crowded spatiotemporal connectivity between trajectories [13–15]. This domino effect makes management very challenging.

Air traffic control (ATC) is a service provided by ground-based air traffic controllers who direct aircraft on the ground and through controlled airspace, including the low-altitude airspace discussed in this research, and can provide advisory services to aircraft in noncontrolled airspace. The conflict detection and conflict resolution (CDR) methods proposed in this paper belong to the ATC service, the primary purpose of which worldwide is to prevent collisions, organize and expedite the flow of air traffic, and provide information and other support for pilots. The opening low-altitude airspace in China, which permits private planes to enter, is below 1,000 m. To avoid collisions in low-altitude airspace, ATC enforces traffic separation rules, which ensure each aircraft maintains a minimum amount of empty space around it at all times.

We propose a temporal and spatial integrated strategy for safety assessment purposes in opening the low-altitude urban airspace of Chinese pilot cities. To handle multithreat encounters between planned trajectories reported to relevant ATM organizations, our novel strategy uses CDR algorithms. A conflict is considered to be a loss of safe separation between two or more aircraft. A spatial grid partitioning system (SGPS) is introduced to transform continuous airspace into limited discrete meshes or grids, which can express trajectories through several grids within various time windows based on a relationship judgment of arranged waypoints. To detect conflicts between trajectories, our conflict detection (CD) algorithm is designed to determine if two trajectories pass through the same grid space within overlapping time windows.

This strategy is applied when a conflict emerges that is a loss of safe separation between two or more aircraft. There are two sets of rules for flying any aircraft: VFR and IFR. VFR stands for Visual Flight Rules and IFR means Instrument Flight Rules. The risk situations discussed in this research are about IFR flights and supporting air traffic controllers. In general, traditional aircraft resolves the detected conflicts through simple altitude adjustments, which are determined by a traffic alert and collision avoidance system (TCAS) [16, 17]. Because of their higher flexibility and manoeuvrability, small fixed-wing aircraft in low-altitude airspace should consider both altitude and velocity adjustment. Thus, our conflict resolution (CR) algorithm integrates a time scheduling-based technique (TST) and vertical change-based technique (VCT), which both operate under predetermined basic principles. These two techniques are applied to revise predefined trajectories by altering temporal and spatial information separately.

To summarize the remainder of this paper, Section 2 summarizes common existing CDR algorithms; Section 3 describes the CDR problem as it relates to low-altitude urban airspace; Section 4 provides a detailed mathematical description of the proposed algorithms; Section 5 depicts the causal model which is constructed based on the temporal and spatial integrated strategy; Section 6 presents the results of multithreat scenario simulations and an analysis of computational performance; our conclusions and future work are detailed in Section 7.

2. Literature Review

Various CDR theories and techniques have become essential to managing collision avoidance problems and improving the safety of trajectory planning for small fixed-wing aircraft in low-altitude airspace. CD, CR, and the optimality of various solutions have been widely discussed by many researchers and practitioners. These systems can be classified into two main types, i.e., tactical and strategic. The first type is geometric algorithms that analyse the relationships between small fixed-wing aircraft and intruders within a geometric space to implement passive collision avoidance through conflict detection and resolution. The second type is trajectory planning algorithms with minimum security constraints that plan collision-free safety routes between the current location and a target location by using trajectory planning algorithms based on the status information of intruders. Dimensionality, manoeuvrability, communication, number of participants, and resolutions are the five key elements that should be considered for small fixed-wing aircraft. A detailed summary of publications addressing CDR problems is provided in Table 1.

Geometric algorithms are regarded as the most intuitive method. Park, Oh, and Tahk [18] proposed a single resolution maneuvering logic called 'Vector Sharing Resolution.' In the event of a conflict, using the miss distance vector to the closest point of approach (CPA), this model evaluates the worst-case conflict scenario between small fixed-wing aircraft and provides directions for small fixed-wing aircraft to share the conflict region. These resolution manoeuvres are generated cooperatively. Considering alteration of horizontal movement alone, Chakravarthy and Ghose [19] developed a novel collision cone approach (CCA) to aid in collision detection and avoidance between irregularly shaped moving objects with unknown trajectories. Goss, Rajvanshi, and Subbarao [20] and Carbone et al. [21] extended the CCA to 3D cases and applied it to aircraft collision avoidance in 3D dynamic environments. The basic principle of the CCA is to construct a sphere protection zone with an intruder at the centre where the target aircraft must avoid the sphere protection zone. The collision cone is constructed using the tangent from the target aircraft to the sphere. The optimal avoidance manoeuvre is calculated by changing the relative velocity of the aircraft to alter the collision cone. Luongo et al. [22] presented an innovative 3D analytical solution for the resolution of pair-wise and noncooperative collision avoidance problems. Their approach facilitates different minimum separations in the vertical and horizontal planes of the nearby aircraft with respect to a nominal trajectory. Smith and Harmon [23] studied a polymerized collision cone method that allows small fixed-wing aircraft to detect and avoid multiple intruders simultaneously.

Trajectory planning algorithms include field collision avoidance, linear programming, and other methods. Khatib

Table 1: Summary of publications addressing the CDR problem.

Publication	Dimension	Manoeuvres	Communication	Multiple		Resolution				
						Geometry algorithm		Trajectory planning algorithm		
	3D	T/V/S/C	Cooperative	Pairwise	Global	PCA	CCA	Force field	Linear programming	Intelligent
Park, Oh, and Tahk [18]	X	C(T,S)	X	X		X				
Chakravarthy and Ghose [19]		T		X			X			
Goss, Rajvanshi, and Subbarao [20]	X	C(T,S)		X			X			
Carbone et al. [21]	X	C(T,S)		X			X			
Luongo et al. [22]	X	C(T,S)		X			X			
Smith and Harmon [23]	X	C(T,V,S)			X		X			
Khatib [24]		T						X		
Liu, Guo, and Liu [25]	X	C(T,V,S)		X				X		
Temizer et al. [26]	X	C(T,V,S)			X				X	
Holt, Biaz, and Aji [27]	X	C(T,V,S)			X			X	X	
Sharma and Ghose [28]	X	C(T,V,S)	X		X					X

Note. T=turns, V=vertical manoeuvres, S=speed changes, and C()=combined manoeuvres.

[24] presented a force field method in which each aircraft is treated as a charged particle and modified electrostatic equations are used to generate conflict resolution manoeuvres. Liu, Guo, and Liu [25] presented a novel method that combines the Lyapunov theorem with force field methods and proved that the balance point is a saddle point, meaning that the balance point is stable only when a small fixed-wing aircraft reaches the target. Temizer et al. [26] formulated the problem of collision avoidance as a partially-observable Markov decision process (POMDP). A genetic POMDP solver can be used to derive avoidance strategies that optimize a cost function that balances flight-plan deviation and collision avoidance. Holt, Biaz, and Aji [27] compared three algorithms based on a mixed linear programming (MILP) strategy, the A∗ algorithm, and artificial potential fields. The results demonstrated that MILP excelled with a small number of aircraft but had computation issues as the number of aircraft increased. The A∗ algorithm struggled with small field sizes but performed well on larger airspace. For collision avoidance problems with multiaircraft coordination and fleet formation, Sharma and Ghose [28] studied the use of a swarm intelligence algorithm to derive a series of behaviour rules for small fixed-wing aircraft in clusters, including agglomeration rules, follow rules, guidance rules, and scattered rules. By setting different weights for the rule parameters, it is possible to achieve the collision avoidance for small fixed-wing aircraft fleets.

Typically, the advantages of geometric algorithms are mainly reflected in obtaining of an optimal solution, whereas trajectory planning algorithms typically need to sacrifice solution optimality to ensure computational efficiency. In future applications, CDR algorithms should improve collision avoidance performance for continuous conflicts in multivehicle, noncooperative, and 3D environments. Additionally, both geometric algorithms and trajectory planning algorithms do not consider the performance parameters of various aircraft. In future research on opening low-altitude airspace, CDR algorithms should place an emphasis on the verification of numerical, semiphysical simulations and flight test methods to develop autonomous collision avoidance systems for small fixed-wing aircraft that meet the requirements of given safety standards to accelerate the airspace opening process.

3. Problem Statement

As stated in Mearns et al. [29], theoretically aircraft collisions should occur with an extremely low probability because airliners are managed by local agencies through standard procedures and various automatic electronic systems for route planning and flight execution. However, various sources of uncertainty (especially the influence of wind) affect the aircraft's ability to maintain their trajectories in a precise and straight way. Thus, the risk of conflict would be existed in the actual flight. Typically, the term 'conflict' is defined as an event in which the time interval, distance, or other parameters between two or more aircraft violate the rules of safe separation. The protected airspace volume of each aircraft should not be violated by any other aircraft. To summarize, although CDR approaches and their corresponding alert and strategy thresholds may be different, their underlying function processes are all homothetic.

The general CDR process can be summarized as follows. First, the regional air traffic situation is derived from programmed trajectories or monitored using equipped sensors and communication devices. For state estimation, the state information of nearby aircraft in the same situation must be broadcasted and collected as known data (i.e., vehicle position and velocity at a given corresponding time). Because of the performance limitations of sensors and the influence of the external environment, uncertainties typically exist in the obtained state values. These uncertainties may affect the CDR results and should be taken into consideration. Next, a dynamic model is used to predict aircraft states in the next period of time. These models serve as future state providers to determine if an encounter will occur. The predicted states are calculated based on procedural information regarding planned routes or extrapolated from current positions and velocity vectors. Sequentially, the current and predicted states of each involved aircraft are used in an aggregative comparison with definitional metrics, such as required time intervals, and distance separations. If the definitional conflict metrics are satisfied, a conflict is detected and the relevant information is given to human operators to determine if corrective action is required to mitigate collision risk. Note that not all detected conflicts will deteriorate into collisions, meaning that resolution strategies are not necessary in all situations. The conflict resolution stage should be initiated when a corresponding action is considered to be necessary. A specific resolution model takes effect based on the set of current states to provide decision advice to human operators. In view of the different work patterns of various CDR approaches, either or both phases of conflict detection and conflict resolution may be automatically or manually executed. Consequently, conflict detection can be regarded as the stage of risk exploration/warning and conflict resolution acts as the stage of action selection/execution.

The protected airspace volume of each low-altitude airspace aircraft is based on the minimum safe distance described by a horizontal distance, vertical distance, and temporal interval. Each aircraft is enveloped by a cylinder isolating it from invading aircraft to ensure safety. Let V_m and H_m be the minimum vertical and horizontal safe distances, respectively. dV and dH are the actual vertical and horizontal distances, respectively, between the target aircraft and invading aircraft. If the following condition is met, a conflict will occur:

$$(dV < V_m) \wedge (dH < H_m) \tag{1}$$

An array consisting of six elements is used to record the conflict:

$$CF_i = (idc, idt, t_s, t_e, P_s, P_e) \tag{2}$$

where idc is the conflict ID number, idt is the ID number of the aircraft trajectory, t_s and t_e are the start time and end time of the conflict, and P_s and P_e are the start position and

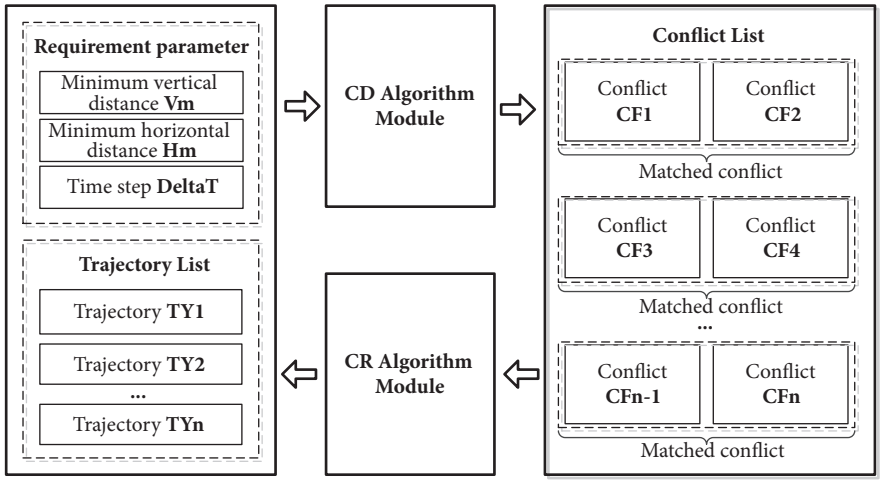


Figure 1: Fundamental structure of the proposed CDR system.

end position of the conflict in WGS84 coordinates. The 3D coordinates $P = (lat, lon, alt)$ include latitude, longitude, and altitude values. For the invading aircraft, the matching conflict CF' satisfies the following conditions with the CF:

$$\left(idc = idc'\right) \wedge \left(t_s = t_s'\right) \tag{3}$$

4. Conflict Detection and Resolution

The main function of the CD module is to detect any potential conflicts between discrete trajectories and transfer the conflict data to the CR module. The input and output data streams, as well as the overall system architecture, are presented in Figure 1. The requirement parameters and trajectory list comprise the input data. The requirement parameters include the minimum vertical distance V_m and minimum horizontal distance H_m that are used to estimate conflicts and ensure safe distances. $DeltaT$ is a discretized time step that is set to meet both precision and efficiency requirements. A smaller $DeltaT$ results in higher precision but lowers efficiency for large scale computations and has greater storage requirements for recorded data. The trajectory list contains all relevant trajectories with unique identification numbers. A trajectory can be represented by

$$TY_i = \left(idt, t_s, t_e, PList\right) \tag{4}$$

where TY_i is the i-th trajectory, idt is the ID of the trajectory, t_s is the start time of the trajectory, t_e is the end time of the trajectory, and $PList$ is the list of planned route points, which are 4D pieces of information (3D position and a time stamp). A planned route point RP_i can be described as

$$RP_i = \left(idp, t, P\right) \tag{5}$$

where idp is the ID of route point RP_i, t is the corresponding time stamp for RP_i, and P is the WGS84 coordinate of RP_i.

The output dataset of the CD algorithm module is a conflict list containing all detected conflicts. Within the conflict list, neighbouring conflicts are matched conflicts that meet the conditions given in (3). Matched conflicts contain valuable information that the CR algorithm uses to determine appropriate corrective manoeuvres. After resolving any detected conflicts, the newly generated trajectories are checked again to determine if any conflicts exist that require resolution.

4.1. Spatial Grid Partitioning System and CD Algorithm. Comparing the discrete waypoints of pair-wise trajectories to detect potential conflicts would lead to very high computation complexity. Thus, the SGPS is introduced. It transforms a continuous space into limited discrete meshes or grids, which can express trajectories using several grid spaces with time windows based on a relationship judgment of planned waypoints [30]. To partition an airspace, the boundary of the considered airspace (i.e., the boundary feature vector (BFV)) is defined as

$$BFV = \left(Lat_{\min}, Lat_{\max}, Lon_{\min}, Lon_{\max}, \mathrm{Alt}_{\min}, \mathrm{Alt}_{\max}\right) \tag{6}$$

where the minimum and maximum values in the WGS84 coordinate space define the airspace. Once the BFV is validated, the partitioned equidistant grid can be represented as

$$PG_i = \left(P_{ma}, P_{mo}, P_{mh}\right) \tag{7}$$

where P_{ma}, P_{mo}, and P_{mh} are the points with the maximum latitude, maximum longitude, and maximum altitude, respectively, which are selected from six points of a cube and considered to be the feature points of the partitioned grid.

To transform a trajectory expression from RP into a grid with a time window, the relationship criterion (8) is used to

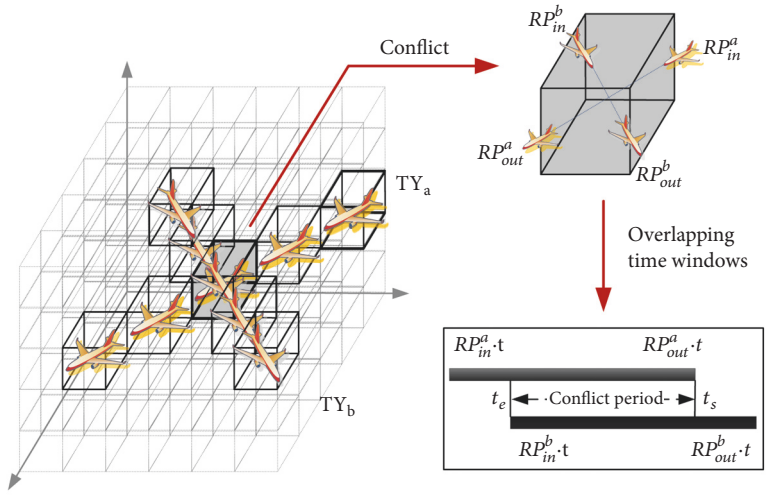


Figure 2: Detailed conflict illustration between two intersecting trajectories.

judge whether or not the discrete route point RP_i is enveloped by the partitioned grid PG_j:

$$\begin{aligned} PG_j.P_{ma}.lat > RP_i.P.lat > PG_j.P_{mo}.lat \\ PG_j.P_{mo}.lon > RP_i.P.lon > PG_j.P_{mh}.lon \\ PG_j.P_{mh}.alt > RP_i.P.alt > PG_j.P_{ma}.alt \end{aligned} \tag{8}$$

Additionally, the first and last route point in the partitioned grid are chosen as the entry waypoint RP_{in} and departure waypoint RP_{out}. If the period $[RP_{in}.t, RP_{out}.t]$ is used as a time window, the list of grid spaces within the time window TWG, expressed by the following, can be constructed to represent the trajectory:

$$TWG_i^j = (lpg, RP_{in}, RP_{out}) \tag{9}$$

where TWG_i^j is the j-th grid space within the time window of the i-th trajectory and lpg is the label of the partitioned grid.

By applying SGPS to CDR, trajectories can be converted from discrete waypoints into a list of grids with time windows. Generally, the format of a hash table is utilized to store trajectory information. However, it is not convenient for managing a grid space that multiple trajectories pass through. A Boolean table should also be generated using the storage format while constructing the hash table. The *Position* value of each unit at position (i, j) is the position in the grid passed through by TWG_j. The *Pass* value of each unit at position (i, j) depends on the condition in

$$BT(i, j).Pass = \begin{cases} 1 & i\text{th trajectory passes through } j\text{th partitioned grid space} \\ 0 & otherwise \end{cases} \tag{10}$$

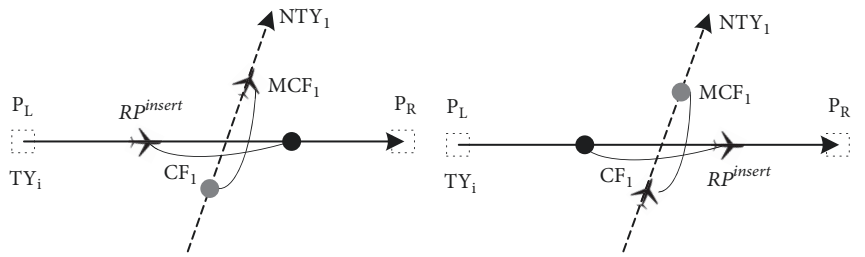
To detect conflicts between trajectories, the system should determine if two trajectories pass through the same grid space within overlapping time windows. For instance, Figure 2 illustrates two trajectories TY_a and TY_b between which a conflict may exist. Besides, we can find the corresponding TWG_a^p and TWG_b^q trajectories using the same table. There is a conflict between TY_a and TY_b if the following condition (11) is satisfied:

$$\begin{aligned} &\left(RP^b_{in}.\mathrm{t} \le RP^a_{in}.t \le RP^b_{out}.t\right) \\ &\wedge \left(RP^a_{in}.\mathrm{t} \le RP^b_{in}.t \le RP^a_{out}.t\right) \end{aligned} \tag{11}$$

For the overlapping time windows, the conflict time window $[t_e, t_s]$ can be calculated using

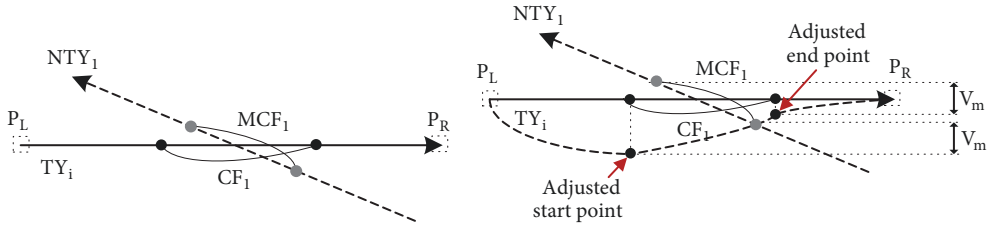
$$\begin{aligned} t_s &= \max\left(RP^a_{in}.t, RP^b_{in}.t\right) \\ t_e &= \min\left(RP^a_{out}.t, RP^b_{out}.t\right) \end{aligned} \tag{12}$$

4.2. Temporal and Spatial Integrated CR Algorithm. The CR algorithm proposed in this paper utilizes concepts based on a set of manoeuvres, such as altitude and speed adjustments, to solve the conflict resolution problem. These manoeuvres are determined based on the matched conflict information detected by the CD algorithm and are performed in a medium-term time window. In this study, it is assumed that aircraft fly with uniform velocities and travel in straight lines between any two adjacent waypoints. Additionally, the time required to execute state changes (i.e., time to implement any manoeuvre) is ignored.



(a) Decelerate situation (b) Accelerate situation

Figure 3: Decelerate and accelerate situations for CR using TST schematic diagram.



(a) Conflict situation (b) Vertical change for CR

Figure 4: Altitude adjustment of start and end points for CR by VCT schematic diagram.

There are several measures that can be taken to resolve conflicts, including modifications of speed, direction, or altitude. Changes in direction or altitude will lead to trajectory variations, whereas changes in speed will only result indifferent times for passing through waypoints. To generate an optimal strategy, a time scheduling-based technique (TST) and vertical change-based technique (VCT) are integrated. The TST strategy is given top priority in this paper because it adds fewer new waypoints to the original trajectories.

A candidate trajectory TY_i must be checked against the conflict-free trajectory list $NTYList$ to generate a series of matched conflicts marked with timestamps. Matched conflict grouping aims to divide detected conflicts into enumerable groups. Any conflict CF_j of TY_i must appear between two waypoints, which are either adjacent or separated by several waypoints. Let P_L denote the closest waypoint to all waypoints before the start time of CF_j and P_R denote the closest waypoint to all waypoints after the end time of CF_j.

$$P_L = \left\{RP_i \mid RP_i \in TY_i.PList, RP_i.t < CF_j.t_s, \max i\right\} \quad (13)$$

$$P_R = \left\{RP_i \mid RP_i \in TY_i.PList, RP_i.t > CF_j.t_e, \min i\right\} \quad (14)$$

All conflicts of TY_i sharing the same P_L and P_R will be put into the same group. Then, the first conflict in each group is chosen for resolution.

The trajectory to be amended is referred to as a candidate trajectory. It may initiate a new conflict with any trajectories in the conflict-free trajectory list, which will not be modified further. As shown in Figure 3, a conflict CF_1 and its matched conflict MCF_1 occur between the candidate trajectory TY_i and a conflict-free trajectory NTY_1. The TST performs a decelerating or accelerating operation on the aircraft to resolve the conflict. In Figure 3(a), the aircraft associated with TY_i should decelerate to reach the conflict start point until another aircraft leaves the conflict end point. In contrast, in Figure 3(b), the aircraft associated with TY_i should accelerate to leave the conflict end point before another aircraft reaches the conflict start point. The deceleration process will be attempted first by the aircraft associated with the candidate trajectory. If deceleration is infeasible, then the acceleration process will be attempted. Deceleration and acceleration processes may appear in alternating fashion in the CR course if multiple conflicts are detected in one group.

Modification of the candidate trajectory by the TST in both situations essentially inserts a new waypoint into the original trajectory. It can be seen in Figure 3 that the aircraft icons on TY_i are the waypoints inserted to avoid collision. The newly inserted waypoint RP^{insert} can be determined using (15). A time constant δ is introduced, without which the safety distance envelope may be violated (i.e., $\delta = 0$):

$$RP^{insert} = \begin{cases} (idp, CF.t_e + \delta, CF.P_s) & \text{deceleration} \\ (idp, CF.t_s - \delta, CF.P_e) & \text{acceleration} \end{cases} \quad (15)$$

The VCT changes the altitude of waypoints in the candidate trajectory to maintain a safe vertical distance during a conflict time window. As shown in Figure 4, a 2D schematic diagram can be used to express this concept. Two pairwise conflicts CF_1 and MCF_1 detected between candidate trajectories TY_i and NTY_1 in the conflict-free trajectory list are illustrated in Figure 4(a). The VCT attempts to adjust the altitude of the start and end points of the conflict to maintain a vertical safe distance. Two corresponding waypoints are constructed and inserted into the candidate trajectory after adjustment. Figure 4(b) depicts the main operations used to

adjust the start point and end point of the conflict. Besides, V_m denotes the vertical safe distance.

Whether the aircraft will ascend or descend should be determined before it arrives at the conflict start point. Therefore, relative altitude change velocity (RACV) is defined to represent the altitude change velocity of the candidate aircraft relative to the matched aircraft in the conflict:

$$RACV = dA_{candidate} - dA_{matched} = \frac{CF.P_e.alt - CF.P_s.alt}{CF.t_e - CF.t_s} - \frac{MCF.P_e.alt - MCF.P_s.alt}{MCF.t_e - MCF.t_s} \tag{16}$$

If $RACV > 0$, the candidate aircraft should stay above the matched aircraft. Otherwise, it should stay below the matched one. Additionally, the two constructed waypoints RP_s^{insert} and RP_e^{insert} can be calculated using the following, respectively:

$$RP_s^{insert} = \begin{cases} (idp, CF.t_s, (CF.P_s.lat, CF.P_s.lon, MCF.P_s.alt + V_m)) & RACV > 0 \\ (idp, CF.t_s, (CF.P_s.lat, CF.P_s.lon, MCF.P_s.alt - V_m)) & otherwise \end{cases} \tag{17}$$

$$RP_e^{insert} = \begin{cases} (idp, CF.t_e, (CF.P_e.lat, CF.P_e.lon, MCF.P_e.alt + V_m)) & RACV > 0 \\ (idp, CF.t_e, (CF.P_e.lat, CF.P_e.lon, MCF.P_e.alt - V_m)) & otherwise \end{cases} \tag{18}$$

5. Model Formulation

This research presents the mechanism of temporal and spatial integrated strategy used to resolve detected conflicts. The two synergic techniques, TST and VCT, are applied to revise the predefined trajectories from the different views of altering temporal and spatial information separately. Actually, both of them finally just modify the trajectories in terms of adding new waypoints to the original one. The CDR algorithms based model is constructed in the graphical modelling and analysis software (GMAS) [31] , generating the state space to provide a global perspective on the scenario dynamics and a better understanding of the potential conflict occurrence for risk assessment. It would offer auxiliary supports in the analysis of hectic traffic scenarios, e.g., terminal maneuvering area (TMA) and hot spots [11]. And the airspace capacity would be increased in opening low-altitude urban airspace, while a higher amount of flights would be safely and efficiently managed.

5.1. Graphical Modelling and Analysis Software. For the experiments, we previously used the state space analysis tool called TIMSPAT, developed at the Logistics and Aeronautics Unit of the Autonomous University of Barcelona. The tool has been shown to be effective for the performance analysis of very demanding and flexible industrial systems [32]. Yet it is failing to provide the graphical modelling interface faced industrial systems; therefore the errors are difficult to detect in the developing process and TIMSPAT model constituted by a set of text files is not easy to understand the model architecture. Besides, as one of the most commonly used tools for modelling and simulating discrete-event systems, though CPN Tools [33–35] stand out as an industrial strength software that provides both a graphical editing interface and an interactive simulator for constructing and analysing models, it has some limitations. Its earlier version supports extraordinarily simple calculation only And even with this extension, the up-to-date version is still difficult to integrate complex operation. In addition, it has a state space analysis plug-in, but the absence of efficient search algorithms has limited its applicability and it cannot scale up to industrial-sized problems [36]. To overcome the above-mentioned shortcomings of TIMSPAT and CPN Tools, the graphical modelling and analysis software (GMAS) has been developed. GMAS is a powerful graphical and mathematical modelling tool, which has been extensively used to model, simulate, and analyse complex systems characterized by concurrency, parallelism, causal dependency, resource sharing, and synchronization.

The graphical components of GMAS model include start component, data component, function component, nested function component, link component and end component.

Definition 1. The GMAS model can be defined as the following nine-tuple [37]:

$$GM = \left(S, D, H, H', F, E, F_W, M, M_0\right) \tag{19}$$

wherein

$S = \{s_1\}$ represents the set of start component and the element is unique.

$D = \{d_1, d_2, \ldots, d_a\}$ represents the set of data components and a is the amount.

$H = \{h_1, h_2, \ldots, h_m\}$ represents the set of function components and m is the amount.

$H' = \{h'_1, h'_2, \ldots, h'_n\}$ represents the set of nested function components and n is the amount.

$F = \{f_1, f_2, \ldots, f_u\}$ represents the set of link components and u is the amount.

$E = \{e_1\}$ represents the set of end component and the element is unique.

$F_W : F \longrightarrow \{f_{1,w}, f_{2,w}, \ldots, f_{u,w}\}$ is the set of functions on each link component.

$M : S \cup E \cup D \longrightarrow \{s_1, e_1, d_1, d_2, \ldots, d_a\}$ is the set of state identifications, which are the state data of start component,

end component, and data components during the model operation.

$M_0 : S \cup D \longrightarrow \{s_{1,0}, d_{1,0}, d_{2,0}, \ldots, d_{a,0}\}$ is the set of initial identifications, which are the initial state data of start component and data components before the model operation.

$D \cap (H \cup H') = \emptyset$ (set D does not intersect with the union of set H and H') and $D \cup (H \cup H') \neq \emptyset$ (set D and the union of set H and H' are not empty at the same time).

$F \subseteq [(S \cup D \cup E) \times (H \cup H')] \cup [(H \cup H') \times (S \cup D \cup E)]$ indicates that link component connects start component, data component, or end component with function component or nested function component and it is the set of directed arcs.

Among them, $G = (S, D, H, H', F, E, F_W)$ forms the physical structure of the GMAS model, (G, M) is called identified GMAS model, and its feature is the introduction of state identification M which is a vector set of the data in start component, end component and data components at the corresponding time. (G, M, M_0) expresses the complete GMAS model in which the initial states M_0 have been provided with the input data in start component and the initial data in data components at the beginning [31].

5.2. Model Representation. The causal model should be informed with the trajectories of original aircraft involved in the same scenario. Each aircraft trajectory is discrete to be several segments with corresponding end points as the pivotal waypoints that the corresponding aircraft will follow. It attempts to detect and resolve the threat based on the proposed CDR algorithms and determines whether a negative domino threat occurs in the resolution process of the previous threats. Based on the mathematical description presented in Section 4, the purpose of the model that results in the GMAS formalism is not only to resolve conflicts based on step-by-step logic but also focuses on exploring the dynamic relations between the resolution trajectories and the neighbouring trajectories. The developed causal model illustrated in Figure 5 is based on the aircraft tracking waypoints, consisting of three kinds of agents (Agent Trajectory List Update, Agent Conflict Detection, and Agent Conflict Resolution) that model for the successive CDR operations. It includes seven function components $(h_1, h_2, h_3, h_4, h_5, h_6, h_7)$ and two nested function components (h_1': TST and h_2': VCT), depicted in Table 2.

The description of start, data, and end components are shown in Table 3. The link components are used to connect theses components with correlative function components and transmit the input and output data streams.

5.2.1. Agent Trajectory List Update. The discrete trajectories seem to take a 'snapshot' of the involved aircraft in the checked scenario. The new trajectories should be added to the trajectory list to update it, while several conflicts may be initiated with any existing trajectories in conflict-free trajectory list which will remain unchanged. Agent Trajectory List Update contains three function components h_1, h_2, and h_3, wherein h_1 aims to set the initial parameters, i.e., conflict-free trajectory list *NTYList*=NULL, conflict-existing trajectory list *CTYList=NULL*, and count *lable i=0*; h_2 is used to update trajectory list *TYList* when the new trajectories are added; h_3 mainly sorts the trajectory list *TYList* based on the defined trajectory priority, i.e., arrival time in this paper.

5.2.2. Agent Conflict Detection. The candidate trajectory TY_i should be checked with the conflict-free trajectory list *NTYList*. The main function of Agent Conflict Detection is to detect the potential conflict and transfer the generated matched conflicts marked by time to the Agent Conflict Resolution. Agent Conflict Detection contains two function components h_4 and h_5, wherein h_5 detects the potential conflict between *TYi* and *NTYList*, to check whether two trajectories pass the same grid with the crossed time-window; h_6 mainly divides the detected matched conflicts into enumerable *Nc* groups, to make it convenient for the identification and resolution of the successive conflicts.

5.2.3. Agent Conflict Resolution. In view of the information of detected conflicts from the Agent Conflict Detection, the proposed CR algorithm implemented by means of the Agent Conflict Resolution tries to resolve these conflicts based on the altitude or/and speed adjustment. Agent Conflict Resolution contains two function components h_6 and h_7 and two nested function components h_1' and h_2', wherein h_6 aims to execute the provided temporal and spatial integrated principle, used as a conflict resolution controller; h_1' changes the aircraft velocity that causes the different time to pass waypoints, while h_2' amends the aircraft altitude and direction that leads to the trajectory variation; h_7 collects the revised *TYi* which has no conflicts with other trajectories and adds it to the conflict-free trajectory list *NTYList*. The description of two nested function components h_1' and h_2' is shown below:

(i) The nested function component h_1' TST employs the decelerating or accelerating operation on the aircraft to resolve the conflict, and it includes four function components $h_{1,1}'$, $h_{1,2}'$, $h_{1,3}'$, and $h_{1,4}'$, wherein $h_{1,1}'$ aims to set the related parameters; $h_{1,2}'$ chooses the *ith* element of TY_iwaypoints as the inserted one to avoid collision; $h_{1,3}'$ indicates that aircraft of TY_i should accelerate to depart from the conflict end point before another aircraft reach the conflict start point, while $h_{1,4}'$ indicates that aircraft of TY_i should decelerate to reach the conflict begin point until another aircraft departs from the conflict end point.

(ii) The nested function component h_2' VCT amends the altitude and direction of the aircraft to ensure a safety vertical distance during the conflict time, and it includes four function components $h_{2,1}'$, $h_{2,2}'$, $h_{2,3}'$ and $h_{2,4}'$, wherein $h_{2,1}'$ aims to set the related parameters; $h_{2,2}'$ chooses the *i-th* element of TY_iwaypoints as the one which begins to change; $h_{2,3}'$ splits the altitude variation equitably based on the relative altitude change velocity; $h_{2,4}'$ calculates the two newly constructed waypoints RP_s^{insert} and RP_e^{insert}.

Table 2: Information of function components.

Class	Num.	Description
Main Model	h_1	Set initial parameters
	h_2	Update trajectory list *TYList*
	h_3	Sort *TYList* by set priority
	h_4	Check the conflict between *TYi* and *NTYList*
	h_5	Divide matched conflicts into *Nc* groups
	h_6	Conflict Resolution Controller
	h'_1	Time Scheduling Based Technique
	h'_2	Vertical Change Based Technique
	h_7	Add revised *TYi* to *NTYList*
h'_1	$h'_{1,1}$	Set related parameters
	$h'_{1,2}$	Choose the *i-th* element of TY_i waypoints as *RP*
	$h'_{1,3}$	Accelerate
	$h'_{1,4}$	Decelerate
h'_2	$h'_{2,1}$	Set related parameters
	$h'_{2,2}$	Choose the *i-th* element of TY_i waypoints as *RP*
	$h'_{2,3}$	Split the altitude variation equitably
	$h'_{2,4}$	Calculate RP_s^{insert} and RP_e^{insert}

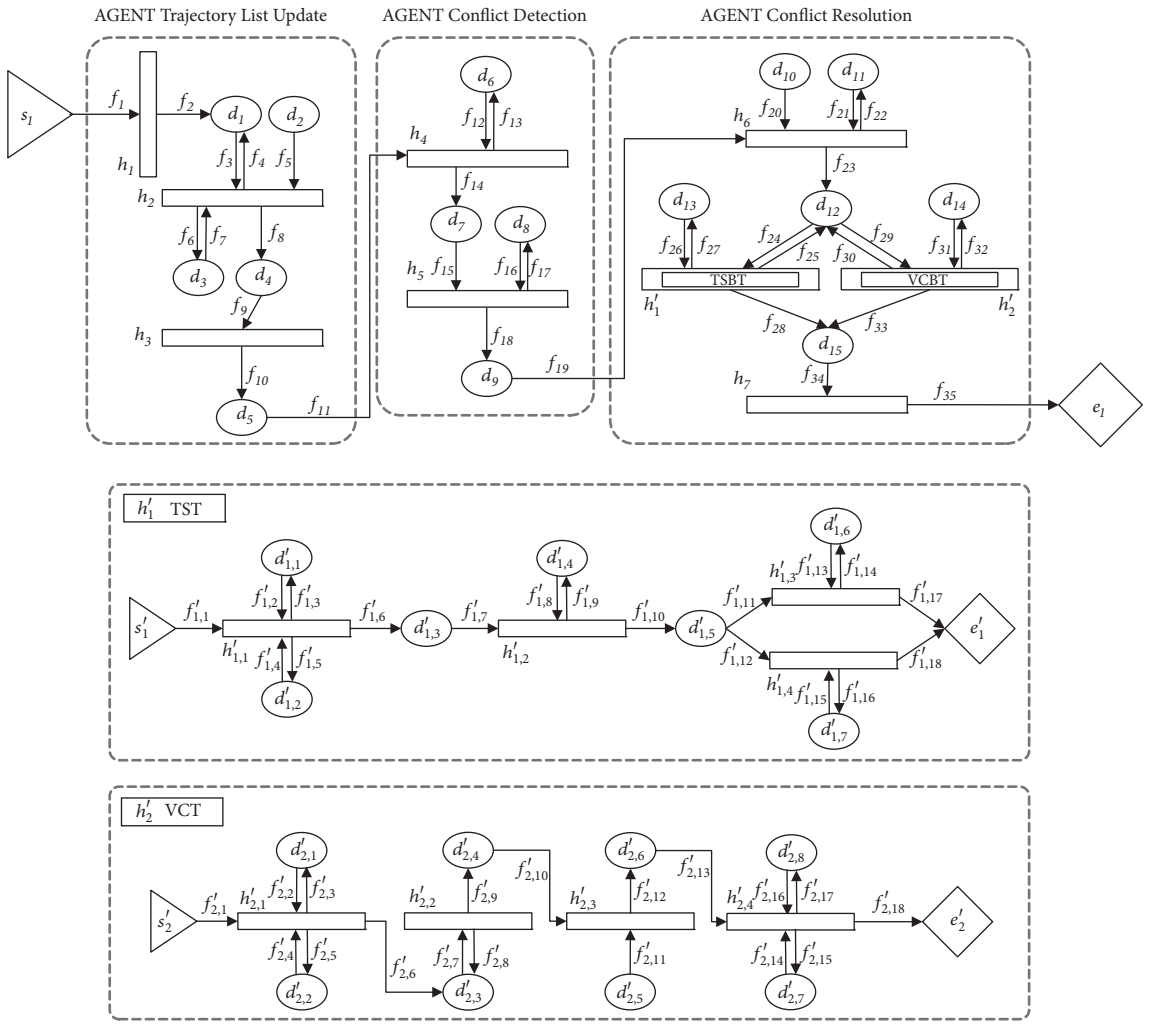


Figure 5: GMAS-based causal model of CDR algorithms.

Table 3: Information of start, data, link, and end components.

Class	Num.	Input	Output	Description
Main Model	s_1	/	f_1	Basic trajectory data
	d_1	f_2,f_4	f_3	Initial parameters
	d_2	/	f_5	New trajectories
	d_3	f_6	f_7	Counting component
	d_4	f_8	f_9	Updated trajectory list
	d_5	f_{10}	f_{11}	Sorted trajectory list
	d_6	f_{13}	f_{12}	Counting component
	d_7	f_{14}	f_{15}	Potential conflict
	d_8	f_{17}	f_{16}	Counting component
	d_9	f_{18}	f_{19}	Divided detected matched conflicts
	d_{10}	/	f_{20}	Temporal and spatial integrated principle
	d_{11}	f_{22}	f_{21}	Counting component
	d_{12}	f_{23},f_{25},f_{30}	f_{24},f_{29}	Enumerable groups with integrated principle
	d_{13}	f_{27}	f_{26}	Counting component
	d_{14}	f_{32}	f_{31}	Counting component
	d_{15}	f_{28},f_{33}	f_{34}	Revised conflict trajectory list
	e_1	f_{35}	/	Improved conflict-free trajectory list
h_1'	s_1'	/	$f_{1,1}'$	TST trajectory groups
	$d_{1,1}'$	$f_{1,3}'$	$f_{1,2}'$	Counting component
	$d_{1,2}'$	$f_{1,5}'$	$f_{1,4}'$	Related parameters
	$d_{1,3}'$	$f_{1,6}'$	$f_{1,7}'$	TST trajectory groups with set parameters
	$d_{1,4}'$	$f_{1,9}'$	$f_{1,8}'$	Counting component
	$d_{1,5}'$	$f_{1,10}'$	$f_{1,11}'$, $f_{1,12}'$	Selected *i-th* element of TY_i waypoints
	$d_{1,6}'$	$f_{1,14}'$	$f_{1,13}'$	Acceleration criterion
	$d_{1,7}'$	$f_{1,16}'$	$f_{1,15}'$	Deceleration criterion
	e_1'	$f_{1,17}'$, $f_{1,18}'$	/	Revised conflict trajectory list
h_2'	s_2'	/	$f_{2,1}'$	VCT trajectory groups
	$d_{2,1}'$	$f_{2,3}'$	$f_{2,2}'$	Counting component
	$d_{2,2}'$	$f_{2,5}'$	$f_{2,4}'$	Related parameters
	$d_{2,3}'$	$f_{2,6}'$, $f_{2,8}'$	$f_{2,7}'$	VCT trajectory groups with set parameters
	$d_{2,4}'$	$f_{2,9}'$	$f_{2,10}'$	Selected *i-th* element of TY_i waypoints
	$d_{2,5}'$	/	$f_{2,11}'$	Altitude variation criterion
	$d_{2,6}'$	$f_{2,12}'$	$f_{2,13}'$	Splited altitude variation
	$d_{2,7}'$	$f_{2,15}'$	$f_{2,14}'$	Counting component
	$d_{2,8}'$	$f_{2,17}'$	$f_{2,16}'$	Newly constructed waypoints
	e_2'	$f_{2,18}'$	/	Revised conflict trajectory list

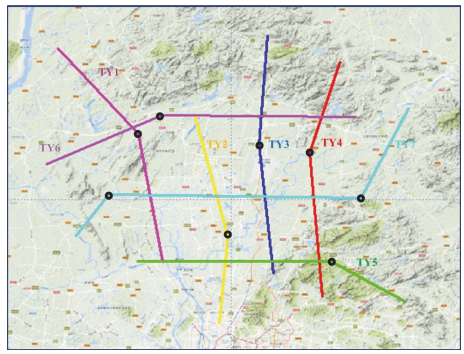


Figure 6: Scenario with seven trajectories.

6. Numerical Experiments

In order to verify and explain the temporal and spatial integrated strategy, a special scenario is introduced. Additionally, scenarios with different numbers of trajectories are used to test the algorithm's efficiency. A ThinkPad laptop with an Intel Core i7 2.4 GHz processor and 8GB of RAM was used for the computations in the numerical experiments.

6.1. Specific Case. Figure 6 presents the top view of a scenario with seven trajectories labelled from LAA-01 to LAA-07, which are distributed such that they cross with four conflicts. This is a realistic case using the low altitude airspace above the city of Guangzhou in China as a background. Relevant data were provided by the Air Traffic Control Centre of

Table 4: Information on trajectories in the simulated scenario.

Trajectory	*idt*	*Plist*
TY_1	LAA-01	(0s, 23.254°, 113.158°, 800m), (470s, 23.443°, 113.122°, 800m), (694s, 23.568°, 113.004°, 800m)
TY_2	LAA-02	(180s,23.471°, 113.205°, 800m), (480s, 23.292°, 113.253°, 800m), (600s, 23.164°, 113.241°, 800m)
TY_3	LAA-03	(60s,23.588°, 113.313°, 800m), (330s, 23.427°, 113.300°, 800m), (683s, 23.238°, 113.322°, 800m)
TY_4	LAA-04	(300s,23.549°, 113.419°, 800m), (480s, 23.416°, 113.375°, 800m), (780s, 23.203°, 113.393°, 800m)
TY_5	LAA-05	(240s,23.196°, 113.514°, 800m), (407s, 23.256°, 113.406°, 800m), (786s, 23.256°, 113.120°, 800m)
TY_6	LAA-06	(394s,23.397°, 112.987°, 800m), (642s, 23.471°, 113.156°, 800m), (1021s, 23.467°, 113.442°, 800m)
TY_7	LAA-07	(0s,23.292°, 113.030°, 800m), (166s, 23.353°, 113.078°, 800m), (669s, 23.348°, 113.450°, 800m), (847s, 23.480°, 113.520°, 800m)

Table 5: CD process for the seven trajectories in the Guangzhou scenario.

CD step	Matched conflicts
1	{LAA-01, 241s,280s, (23.351°, 113.139°, 800.0m), (23.367°, 113.136°, 800.0m)}, {LAA-07, 241s,280s, (23.352°, 113.134°, 800.0m), (23.352°, 113.163°, 800.0m)}; {LAA-02, 355s, 410s, (23.366°, 113.233°, 800.0m), (23.334°, 113.242°, 800.0m)}, {LAA-07, 355s, 410s, (23.351°, 113.218°, 800.0m), (23.350°, 113.258°, 800.0m)}; {LAA-03, 466s, 505s, (23.354°, 113.309°, 800.0m), (23.334°, 113.311°, 800.0m)}, {LAA-07, 466s, 505s, (23.350°, 113.300°, 800.0m), (23.349°, 113.329°, 800.0m)}; {LAA-04, 556s, 596s, (23.362°, 113.379°, 800.0m), (23.334°, 113.382°, 800.0m)}, {LAA-07, 556s, 596s, (23.349°, 113.367°, 800.0m), (23.349°, 113.396°, 800.0m)}.
2	{LAA-02, 404s, 410s, (23.337°, 113.241°, 800.0m), (23.334°, 113.242°, 800.0m)}, {LAA-07, 404s, 410s, (23.351°, 113.234°, 800.0m), (23.351°, 113.239°, 800.0m)}; {LAA-03, 486s, 505s, (23.344°, 113.310°, 800.0m), (23.334°, 113.311°, 800.0m)}, {LAA-07, 466s, 505s, (23.350°, 113.301°, 800.0m), (23.350°, 113.316°, 800.0m)}; {LAA-04, 567s, 596s, (23.354°, 113.380°, 800.0m), (23.334°, 113.382°, 800.0m)}, {LAA-07, 567s, 596s, (23.349°, 113.367°, 800.0m), (23.349°, 113.391°, 800.0m)}.
3	{LAA-03, 491s, 505s, (23.341°, 113.310°, 800.0m), (23.334°, 113.311°, 800.0m)}, {LAA-07, 491s, 505s, (23.350°, 113.300°, 800.0m), (23.350°, 113.312°, 800.0m)}. {LAA-04, 570s, 596s, (23.352°, 113.380°, 800.0m), (23.334°, 113.382°, 800.0m)}, {LAA-07, 567s, 596s, (23.349°, 113.367°, 800.0m), (23.349°, 113.389°, 800.0m)}.
4	{LAA-04, 579s, 596s, (23.346°, 113.381°, 800.0m), (23.334°, 113.382°, 800.0m)}, {LAA-07, 567s, 596s, (23.349°, 113.367°, 800.0m), (23.349°, 113.383°, 800.0m)}.

the Guangzhou government, and the detailed information on trajectories is shown in Table 4. The identified seven trajectories are depicted with different coloured lines and the points marked with black dots are the turning waypoints of their corresponding trajectories. Each trajectory was preplanned for a particular mission, but potential conflicts must be resolved, which will require the adjustment of these trajectories. The parameters for safe distances are set to $H_m = 500m$ and $V_m = 100m$. Additionally, a discretization step of $TimeStep = 1s$ is adopted.

There are no conflicts between the trajectories of aircraft LAA-01 through LAA-06. However, four pairs of matched conflicts appear when the trajectory of LAA-07 is added to the trajectory list. Figure 7 displays the potential conflicts in both 2D and 3D views. Additionally, Table 5 contains detailed information regarding the initial detected conflicts.

Table 6: CR process for the seven trajectories in the Guangzhou scenario.

CR step	CR technique	Plist of revised Trajectory
0	/	(0s,23.292°, 113.030°, 800m), (166s, 23.353°, 113.078°, 800m), (669s, 23.348°, 113.450°, 800m), (847s, 23.480°, 113.520°, 800m)
1	**TST**	(0s,23.292°, 113.030°, 800m), (166s, 23.353°, 113.078°, 800m), (282s, 23.352°, 113.134°, 800m), (669s, 23.348°, 113.450°, 800m), (847s, 23.480°, 113.520°, 800m)
2	**TST**	(0s,23.292°, 113.030°, 800m), (166s, 23.353°, 113.078°, 800m), (282s, 23.352°, 113.134°, 800m), (412s, 23.351°, 113.234°, 800m), (669s, 23.348°, 113.450°, 700m), (847s, 23.480°, 113.520°, 800m)
3	**TST**	(0s,23.292°, 113.030°, 800m), (166s, 23.353°, 113.078°, 800m), (282s, 23.352°, 113.134°, 800m), (412s, 23.351°, 113.214°, 800m), (507s, 23.350°, 113.300°, 800m), (669s, 23.348°, 113.450°, 800m), (847s, 23.480°, 113.520°, 800m)
4	**VCT**	(0s,23.292°, 113.030°, 800m), (166s, 23.353°, 113.078°, 800m), (282s, 23.352°, 113.134°, 800m), (412s, 23.351°, 113.214°, 800m), (507s, 23.350°, 113.300°, 800m), (579s, 23.349°, 113.367°, 700m), (596s, 23.349°, 113.383°, 700m), (669s, 23.348°, 113.450°, 800m), (847s, 23.480°, 113.520°, 700m)

Data records for the entire CDR process are contained in Tables 5 and 6. Note that four steps are used to detect and resolve the conflicts. Steps 1, 2, and 3 use the TST to resolve conflicts. However, the conflict between LAA-04 and LAA-07 cannot be resolved by the TSTB because of speed constraints. Thus, the VCT is applied to resolve this conflict. The modification of the trajectory of LAA-07 in each step is detailed in Table 6.

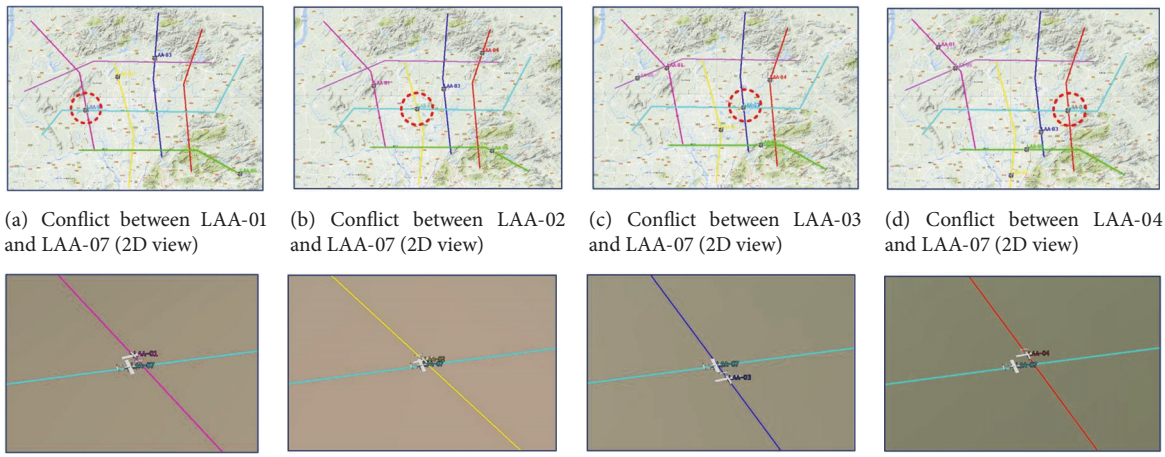
Using the CDR algorithms, the conflicts between the relevant trajectories can be successfully resolved. The critical safe distance is maintained and Figure 8 displays the resolution process in 2D and 3D views. From the scenario presented in Figure 6, three conflicts are resolved by the TST to form Figure 8. However, the conflict depicted in Figure 7(d) cannot be resolved through time adjustments. Therefore, a vertical altitude adjustment of the trajectory is necessary, as shown in Figure 8(d).

6.2. Further Investigation

6.2.1. Average Execution Time. A large number of scenarios must be generated for properly testing the efficiency of the proposed strategy. A scenario generation algorithm with an intersecting chessboard pattern is utilized. The main concept of the scenario generation algorithm is to create two different directions to form an interlaced trajectory net from north to south (FNTS) and from west to east (FWTE). A trajectory is generated in a limited airspace that can be described by six parameters, as shown in (6). In this investigation, the parameters are $Lat_{\min} = 22^{\circ}$, $Lat_{\max} = 23^{\circ}$, $Lon_{\min} = 114^{\circ}$, $Lon_{\max} = 115^{\circ}$, $Alt_{\min} = 500m$, and $Alt_{\max} = 800m$. If a trajectory is ready for generation, its waypoints can be determined using

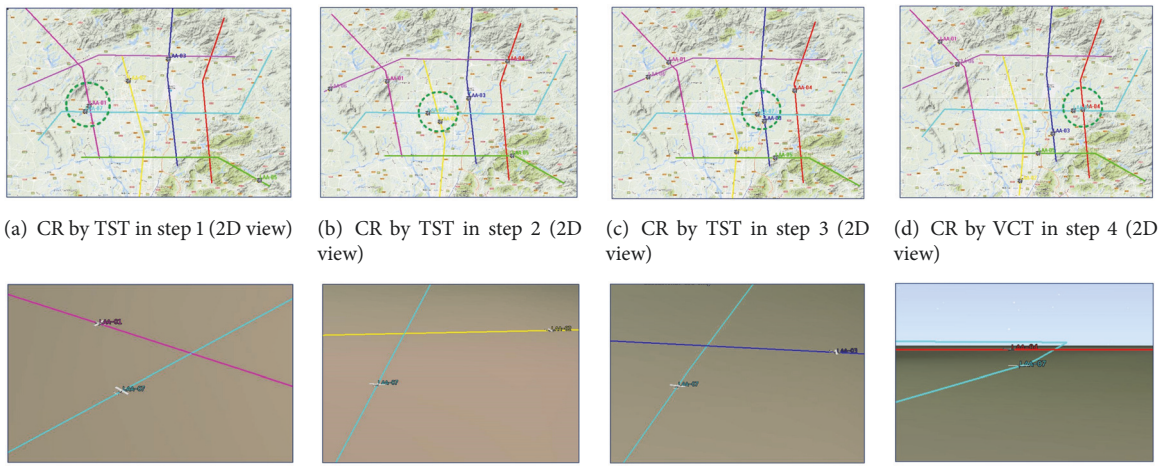
$$RP_0 = \begin{cases} \left(\left(\text{rand}\left(Lat_{\min}, Lat_{\max}\right),\ \text{rand}\left(0, \dfrac{\left(Lon_{\max} - Lon_{\min}\right)}{\left(N_p + 1\right)}\right),\ \text{rand}\left(Alt_{\min}, Alt_{\max}\right) \right), \right. & \text{if FWTE} \\ \left. idt, randint\,(0, T) \right) & \\ \left(\left(\text{rand}\left(0, \dfrac{\left(Lat_{\max} - Lat_{\min}\right)}{\left(N_p + 1\right)}\right),\ \text{rand}\left(Lon_{\min}, Lon_{\max}\right),\ \text{rand}\left(Alt_{\min}, Alt_{\max}\right) \right), \right. & \text{if FNTS} \\ \left. idt, randint\,(0, T) \right) & \end{cases} \tag{20}$$

$$RP_{i+1} = \begin{cases} \left(\left(\text{rand}\left(Lat_{\min}, Lat_{\max}\right),\ RP_i.P.lon + \dfrac{\left(Lon_{\max} - Lon_{\min}\right)}{N_p},\ \text{rand}\left(Alt_{\min}, Alt_{\max}\right) \right), \right. & \text{if FWTE} \\ \left. idt, RP_i.t + V_{\max} + \dfrac{2d_{i,i+1}}{V_{\min}} \right) & \\ \left(\left(RP_i.P.lat + \dfrac{\left(Lon_{\max} - Lon_{\min}\right)}{N_p},\ \text{rand}\left(Lon_{\min}, Lon_{\max}\right),\ \text{rand}\left(Alt_{\min}, Alt_{\max}\right) \right), \right. & \text{if FNTS} \\ \left. idt, RP_i.t + V_{\max} + \dfrac{2d_{i,i+1}}{V_{\min}} \right) & \end{cases} \tag{21}$$



(a) Conflict between LAA-01 and LAA-07 (2D view)
(b) Conflict between LAA-02 and LAA-07 (2D view)
(c) Conflict between LAA-03 and LAA-07 (2D view)
(d) Conflict between LAA-04 and LAA-07 (2D view)
(e) Conflict between LAA-01 and LAA-07 (3D view)
(f) Conflict between LAA-02 and LAA-07 (3D view)
(g) Conflict between LAA-03 and LAA-07 (3D view)
(h) Conflict between LAA-04 and LAA-07 (3D view)

Figure 7: Detected conflicts presented in 2D and 3D views.



(a) CR by TST in step 1 (2D view)
(b) CR by TST in step 2 (2D view)
(c) CR by TST in step 3 (2D view)
(d) CR by VCT in step 4 (2D view)
(e) CR by TST in step 1 (3D view)
(f) CR by TST in step 2 (3D view)
(g) CR by TST in step 3 (3D view)
(h) CR by VCT in step 4 (3D view)

Figure 8: CR in simulated scenario presented in 2D and 3D views.

where N_p is the number of waypoints in the generated trajectory, T is the start time of the trajectory, rand(a, b) is a function for generating a random number between a and b, and randint(a, b) is a function for generating a random integer between a and b.

Six groups of scenario were randomly generated for our numerical experiments. Each group contains 10 different scenarios with the same number of trajectories. Table 7 lists the execution time for each scenario, which was recorded at the end of the corresponding experiment.

To observe the relationships between these execution times within each group, we use the box plots presented in Figure 9. The first three groups (n=80/120/160) are asymmetric, which indicates that execution time is more sensitive to random factors compared to the last three groups (n=200/240/280). For the small scale cases with less than 160 trajectories, execution time likely varied because of the stochastic perturbation caused by random factors. However, as the number of generated trajectories grows, the probabilities of conflicts occurring in the generated scenarios tend to become more uniform. A line graph of the average execution times for each group of scenarios is plotted in Figure 10. It is clear that average execution time increases with the number of considered trajectories. The time complexity of the proposed algorithm is nonlinear, but it is better than exponential. Overall, the time required for our spatial integrated strategy is acceptable for the air traffic management of local low-altitude airspace.

6.2.2. Function Component Analysis. In the simulation model, the set of data components is $D = \{d_1, d_2, \ldots, d_j, \ldots, d_a\}$, $j = 1, 2, \ldots, a$, and the set of function

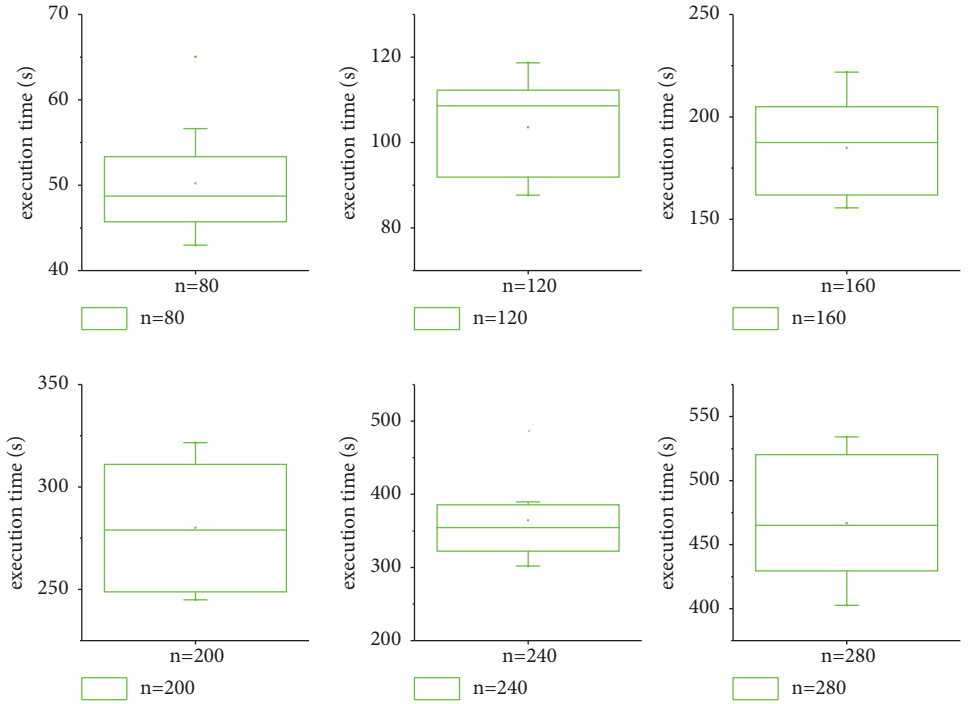


Figure 9: Box plot of execution times grouped by different numbers of trajectories.

Table 7: Execution times for scenarios grouped with different numbers of trajectories.

Trajectory Num.	Execution times of 10 randomly generated scenarios (s)
n=80	50.054, 42.982, 65.052, 53.345, 56.638, 43.786, 46.406, 50.830, 47.401, 45.722
n=120	118.691, 87.674, 112.246, 114.095, 93.902, 108.759, 91.884, 108.447, 90.581, 109.087
n=160	195.884, 166.827, 199.309, 156.510, 161.883, 221.824, 179.134, 204.962, 206.671, 155.644
n=200	255.742, 250.050, 244.957, 248.816, 321.642, 311.000, 246.151, 306.879, 313.914, 302.267
n=240	350.172, 357.369, 302.070, 380.144, 351.800, 322.322, 385.692, 389.604, 314.197, 490.014
n=280	474.792, 520.326, 534.109, 461.321, 468.744, 429.521, 440.444, 404.926, 531.296, 402.598

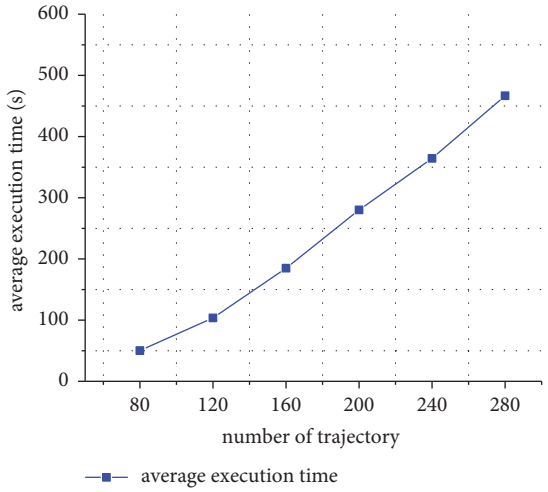


Figure 10: Average execution time with various numbers of trajectories.

components is $H = \{h_1, h_2, \ldots, h_v, \ldots, h_m\}, v = 1, 2, \ldots, m$. The set of reachable states $K(G, M_0) = \{m_0, m_1, \ldots, m_{k-1}\}$ is defined as the collection of all state identifiers that can be reached from the initial state identification M_0 according to the activation rules. The corresponding function components, activating probability and activating execution time of the generated timing states $\{m_1, \ldots, m_{k-1}\}$ based on m_0, are, respectively, set as $h_1, \ldots, h_{k-1}$, $r_1, \ldots, r_{k-1}$, and $t_1, \ldots, t_{k-1}$, and, therein, there may be the same function components of h_s $(s = 1, 2, \ldots, k-1)$. The average execution time of a specific function component is defined as the quotient of the sum of its activation probabilities multiplied by execution time over the number of possible activations. A greater value indicates a longer average time for each activation during simulation (i.e., the component represents a more time-consuming process within a complex system). $t(h_v)$ denotes the average execution time of a function component h_v and can be computed as follows [37]:

$$t(h_v) = \frac{\sum_{h_s=h_v} r_s t_s}{\sum_{h_s=h_v} 1}, \quad v = 1, \ldots, m; \; s = 1, \ldots, k-1 \quad (22)$$

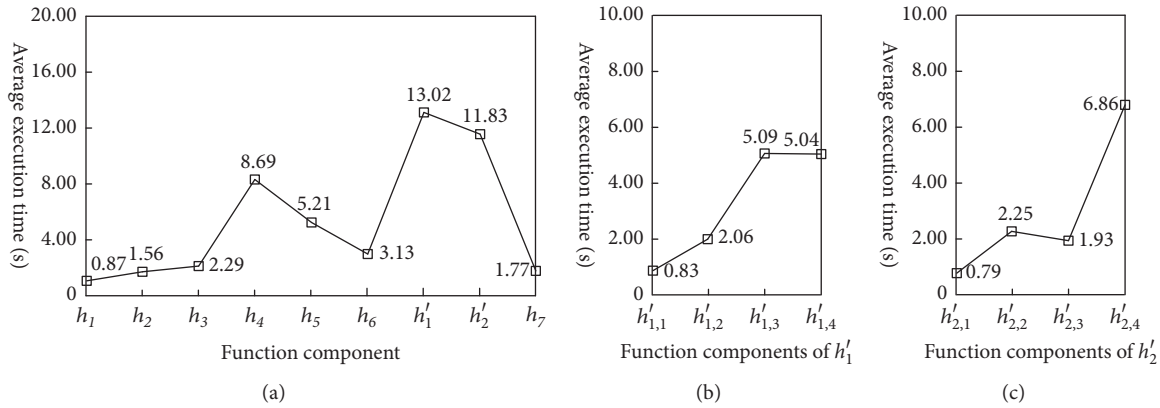


Figure 11: Average execution times of function components in the encounter model (a) and submodels h'_1 (b) and h'_2 (c).

The calculation of this factor is useful for system developers that wish to analyse and optimize the processes with the great influence on a system's efficiency. Figure 11 plots the average execution times of function components in the GMAS-based encounter model and submodels h'_1, h'_2 for test cases with different initial states (averaged over 150 runs). The total average execution time was 48.37s for the n = 80 trajectory scenarios. Figure 11(a) illustrates that the highest execution times are associated with the nested function components h'_1 and h'_2, which are used as the resolution approach. The relatively higher execution time is associated with h_4, which aims to detect conflicts between *TYi* and *NTYList*. Its average execution time can be adjusted by changing the number of added trajectories. The operations of other function components are uncomplicated, meaning that their average execution times are comparatively low, particularly h_1. The average execution times of the four function components $(h'_{1,1}, h'_{1,2}, h'_{1,3}, h'_{1,4})$ of h'_1 are presented in Figure 11(b). Note that $h'_{1,3}$ and $h'_{1,4}$ possess the highest execution times. These components correspond to the complex underlying logic (acceleration and deceleration) at the core of TST. Figure 11(c) presents the average execution times of the four function components $(h'_{2,1}, h'_{2,2}, h'_{2,3}, h'_{2,4})$ of h'_2. Note that $h'_{2,4}$ has the highest execution time because it represents the main procedure of VCT that calculates RP_s^{insert} and RP_e^{insert}. Analogously, $h'_{1,1}$ and $h'_{2,1}$ have the lowest execution times because of the concise and intuitive process of setting related parameters.

7. Conclusion and Future Work

One of the most important strategic challenges related to opening low-altitude airspace is the need to find efficient methods to manage available airspace capacity and ensure a sustainable air transportation system, especially in terminal maneuvering areas where the number of flights is much higher. Improving decision support tools that reduce the task-load on air traffic controllers and improve safety is one of the main methods to take better advantage of airspace capacity. In this paper, an efficient temporal and spatial integrated strategy based on 4D trajectories for detecting and resolving conflicts in the opening low-altitude urban airspace of Chinese pilot cities was presented. The main contributions of this paper are listed below:

(i) Specification of a mathematical model for CDR algorithms based on a novel SGPS. A continuous space can be transformed into a discrete grid that characterizes trajectories using several grid spaces with time windows that correspond to planned pivotal waypoints. This method is utilized to compress critical trajectory data, which leads to significant improvements in computational efficiency.

(ii) Proposal of an innovative temporal and spatial integrated strategy for safety assessment. Our system is designed to aid controllers that manage air traffic flows at a tactical level by providing real-time information regarding possible future conflicts within a look-ahead time window of 30-60 min. It also provides recommendations of feasible methods to resolve conflicts.

(iii) Construction of a causal model to represent system evolution using GMAS. A generated state space can be used to provide a global perspective on scenario dynamics and better understanding of induced conflicts for safety assessment. It also offers auxiliary support in the analysis of hectic traffic scenarios and increases airspace capacity while safely and efficiently managing a higher number of flights.

(iv) Validation of the feasibility and effectiveness of the proposed CDR algorithms. Our algorithms were tested on practical manoeuvre scenarios and an analysis of computational performance was performed through various experiments. Thus, the goal of enhancing airspace capacity to alleviate local airspace network perturbations can be achieved.

As the next research step, we will consider adding no-fly zones to the problem by improving the proposed CDR algorithms. Furthermore, we will package our algorithms as an interactive system for the evaluation of potential induced

conflicts and simulation of incidents that provide a close approximation of real scenarios to support real-time online low-altitude airspace management.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work has been financially supported by the National Natural Science Foundation of China 71601181, the Young Talents Lifting Project 17JCJQQT048, and the National Defense Science and Technology Project Fund 3101175.

References

[1] L. Gu, "China to open low-altitude airspace," 2015: report, 2014, http://www.ecns.cn/cns-wire/2014/11-24/143995.shtml/.

[2] J. Han and Y. Hu, "Free flying and China dream," *Asian Social Science*, vol. 11, no. 26, pp. 212–216, 2015.

[3] F. Chao, "Analysis of China's General Aviation Development and Industry Chain," *China Business and Market*, vol. 28, no. 5, pp. 117–121, 2014.

[4] Q. Cui and Y. Li, "The change trend and influencing factors of civil aviation safety efficiency: The case of Chinese airline companies," *Safety Science*, vol. 75, pp. 56–63, 2015.

[5] T. G. Reynolds, "Air traffic management performance assessment using flight inefficiency metrics," *Transport Policy*, vol. 34, pp. 63–74, 2014.

[6] Y. Hu, H. Liao, S. Zhang, and Y. Song, "Multiple objective solution approaches for aircraft rerouting under the disruption of multi-aircraft," *Expert Systems with Applications*, vol. 83, pp. 283–299, 2017.

[7] O. S. Oubbati, A. Lakas, F. Zhou, M. Güneş, and M. B. Yagoubi, "A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs)," *Vehicular Communications*, vol. 10, pp. 29–56, 2017.

[8] J. Tang, M. A. Piera, and T. Guasch, "Coloured Petri net-based traffic collision avoidance system encounter model for the analysis of potential induced collisions," *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 357–377, 2016.

[9] T. Lehouillier, M. Ilies Nasri, F. Soumis, G. Desaulniers, and J. Omer, "Solving the air conflict resolution problem under uncertainty using an iterative biobjective mixed integer programming approach," *Transportation Science*, vol. 51, no. 4, pp. 1242–1258, 2017.

[10] C.-K. Chan, H. K. H. Chow, S. K. P. So, and H. C. B. Chan, "Agent-based flight planning system for enhancing the competitiveness of the air cargo industry," *Expert Systems with Applications*, vol. 39, no. 13, pp. 11325–11334, 2012.

[11] J. Nosedal, M. A. Piera, A. O. Solis, and C. Ferrer, "An optimization model to fit airspace demand considering a spatio-temporal analysis of airspace capacity," *Transportation Research Part C: Emerging Technologies*, vol. 61, pp. 11–28, 2015.

[12] J. Nosedal and M. A. Piera, "Causal analysis of aircraft turnaround time for process reliability evaluation and disruptions' identification," *Transportmetrica B: Transport Dynamics*, pp. 1–14, 2017.

[13] C. A. Kerrache, A. Lakas, N. Lagraa, and E. Barka, "UAV-assisted technique for the detection of malicious and selfish nodes in VANETs," *Vehicular Communications*, vol. 11, pp. 1–11, 2018.

[14] D. Chen, M. Hu, H. Zhang, J. Yin, and K. Han, "A network based dynamic air traffic flow model for en route airspace system traffic flow optimization," *Transportation Research Part E: Logistics and Transportation Review*, vol. 106, pp. 1–19, 2017.

[15] G. Bruno, E. Esposito, and A. Genovese, "A model for aircraft evaluation to support strategic decisions," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5580–5590, 2015.

[16] M. J. Kochenderfer, M. W. M. Edwards, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, "Airspace encounter models for estimating collision risk," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 487–499, 2010.

[17] J. Tang, "Review: Analysis and Improvement of Traffic Alert and Collision Avoidance System," *IEEE Access*, vol. 5, pp. 21419–21429, 2017.

[18] J.-W. Park, H.-D. Oh, and M.-J. Tahk, "UAV collision avoidance based on geometric approach," in *Proceedings of the SICE Annual Conference 2008 - International Conference on Instrumentation, Control and Information Technology*, pp. 2122–2126, Japan, August 2008.

[19] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: a collision cone approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 28, no. 5, pp. 562–574, 1998.

[20] J. Goss, R. Rajvanshi, and K. Subbarao, "Aircraft conflict detection and resolution using mixed geometric and collision cone approaches," in *Proceedings of the Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, pp. 670–689, USA, August 2004.

[21] C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, "A novel 3D geometric algorithm for aircraft autonomous collision avoidance," in *Proceedings of the 45th IEEE Conference on Decision and Control 2006, CDC*, pp. 1580–1585, USA, December 2006.

[22] S. Luongo, F. Corraro, U. Ciniglio, V. Di Vito, and A. Moccia, "A novel 3D analytical algorithm for autonomous collision avoidance considering cylindrical safety bubble," in *Proceedings of the 2010 IEEE Aerospace Conference*, USA, March 2010.

[23] A. L. Smith and F. G. Harmon, "UAS collision avoidance algorithm based on an aggregate collision cone approach," *Journal of Aerospace Engineering*, vol. 24, no. 4, pp. 463–477, 2011.

[24] O. Khatib, "Real-time obstacle avoida nce for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[25] J. Y. Liu, Z. Q. Guo, and S. Y. Liu, "The simulation of the uav collision avoidance based on the artificial potential field method," in *In Advanced Materials Research*, pp. 1400–1404, Trans Tech Publications, 2012.

[26] S. Temizer, M. J. Kochenderfer, L. P. Kaelbling, T. Lozano-Pérez, and J. K. Kuchar, "Collision avoidance for unmanned aircraft

using Markov Decision Processes," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Canada, August 2010.

[27] J. Holt, S. Biaz, and C. A. Aji, "Engineering notes:Comparison of unmanned aerial system collision avoidance algorithms in a simulated environment," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 881–883, 2013.

[28] R. K. Sharma and D. Ghose, "Collision avoidance between UAV clusters using swarm intelligence techniques," *International Journal of Systems Science*, vol. 40, no. 5, pp. 521–538, 2009.

[29] K. Mearns, B. Kirwan, T. W. Reader, J. Jackson, R. Kennedy, and R. Gordon, "Development of a methodology for understanding and enhancing safety culture in Air Traffic Management," *Safety Science*, vol. 53, pp. 123–133, 2013.

[30] S. Ruiz, M. A. Piera, J. Nosedal, and A. Ranieri, "Strategic deconfliction in the presence of a large number of 4D trajectories using a causal modeling approach," *Transportation Research Part C: Emerging Technologies*, vol. 39, pp. 129–147, 2014.

[31] J. Tang, F. Zhu, and L. Fan, "Simulation modelling of traffic collision avoidance system with wind disturbance," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 4, pp. 36–45, 2018.

[32] O. T. Baruwa and M. A. Piera, "Identifying FMS repetitive patterns for efficient search-based scheduling algorithm: A colored Petri net approach," *Journal of Manufacturing Systems*, vol. 35, pp. 120–135, 2015.

[33] R. Davidrajuh and B. Lin, "Exploring airport traffic capability using Petri net based model," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10923–10931, 2011.

[34] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3-4, pp. 213–254, 2007.

[35] J. Skorupski and A. Florowski, "Method for evaluating the landing aircraft sequence under disturbed conditions with the use of Petri nets," *The Aeronautical Journal*, vol. 120, no. 1227, pp. 819–844, 2016.

[36] O. T. Baruwa, M. A. Piera, and A. Guasch, "TIMSPAT – Reachability graph search-based optimization tool for colored Petri net-based scheduling," *Computers & Industrial Engineering*, vol. 101, pp. 372–390, 2016.

[37] J. Tang and F. Zhu, "Graphical Modelling and Analysis Software for State Space-based Optimization of Discrete Event Systems," *IEEE Access*, vol. 6, pp. 38385–38398, 2018.