Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)

A Heuristic Algorithm for Aircraft 4D Trajectory Optimization Based on Bezier Curve

Weibin Dai, Jun Zhang
National Key Lab of CNS/ATM
Beihang University
Beijing, China

Daniel Delahaye
OPTIM Team
French Civil Aviation University
Toulouse, France

Xiaoqian Sun*
National Key Lab of CNS/ATM
Beihang University
Beijing, China

***Abstract*—In this study, we propose an aircraft 4D trajectory optimization model based on Bezier curve. Many real-world factors (such as winds, obstacles, uncertainties) and actions (the modification of departure time, the trajectory shape, aircraft speed and altitude) are taken into account. To solve the model, an improved simulated annealing algorithm with two phases was proposed: the first phase for reducing the number of conflicts and the second phase for decreasing the total flight time. A national-size dataset for France which is provided by a fast time simulator (ΠRATS) is used as a case study. The experimental results show that the algorithm provides conflict-free trajectories within a very short time for all instances. For the objective to deconflict aircraft, the algorithm is almost linearly scalable for large-scale instances. For a given limited run time (such as 6 hours), the algorithm provides good solutions with small values of objective function (total flight time, changes of aircraft speed and obstacles encounters).**

***Keywords*—Aircraft trajectory optimization; Obstacle avoidance; Winds; Uncertainty; Simulated annealing algorithm**

I. Introduction

In recent years, the rapid growth of air transportation demands result in a high level of congestion at airports and airspace, which leads to considerable economic losses and safety problems. Let N_h and N_v be the minimum distance for separation in the horizontal plane and the vertical direction. Aircraft are considered to be in conflict if both of these minimum separation distances are violated. In order to cope with the increasing demand in air traffic, conflict-free trajectory operations are applied to Air Traffic Management (ATM) system. Aircraft will follow its trajectory which is defined in four dimensions (3D space and time). The goal for this concept is to design a conflict-free trajectory for each aircraft and to optimize other properties (such as flight time and the speed changes) in addition [1].

In this study, we propose a model to address this trajectory optimization problem. In our model, Bezier curve is used to generate the shape of a trajectory on the horizontal profile by using a set of virtual points near the initial en-route segment of each trajectory. The influences of winds and obstacles are also taken into account. The real speed of aircraft at each time is affected by the winds. In addition, the states of aircraft are influenced by unexpected events, which causes the deviation between the real positions of aircraft and the planned trajectories. Therefore, the uncertainty of aircraft position is also taken into account in the model. To reduce the conflicts between aircraft and improve trajectories, we allow a modification of the departure time, the trajectory shape, the aircraft speed and the altitude for each trajectory. To solve the model, we propose an improved simulated annealing algorithm. The algorithm is divided into two phases: the first phase for reducing the number of conflicts and the second phase for decreasing the total flight time. A national-size dataset which is provided by a fast time simulator (ΠRATS)[1] is used as a case study. The major contributions of this paper are:

1) We propose an aircraft trajectory optimization model based on Bezier curve. Many real-world factors (such as winds, obstacles, uncertainties) and actions (the modification of departure time, trajectory shape, aircraft speed and altitude) are taken into account.
2) We designed and implemented an improved simulated annealing algorithm, whose performance is evaluated on a realistic large-scale data set covering 8,310 trajectories with 19,816 initial conflicts.
3) The experiments show that our algorithm is almost linearly scalable for large-scale instances with the objective of trajectory deconfliction. In addition, when further optimizing trajectories while preserving the conflict-free property, our algorithm provides good solutions within reasonable run time.

The remainder of this paper is organized as follows: We provide a literature review on aircraft trajectory optimization in Section II. The mathematical model is presented in Section III. The methodology which is used to implement and solve the model is proposed in Section IV. Experiments on a national-size dataset are reported in Section V. The paper concludes with Section VI.

II. Literature review

In recent years, a number of models and algorithms for aircraft trajectory optimization problems have been proposed.

Email address: daiweibin@buaa.edu.cn, buaazhangjun@vip.sina.com, daniel@recherche.enac.fr, sunxq@buaa.edu.cn

[1]Fast time simulator developed at OPTIM Team.

TABLE I. A summary table for most mentioned references

	Allowed actions for modification				Considered factors		
Ref.	Dep. time	Shape	Speed	Alti-tude	Wind effect	Obstacle avoidance	Uncer-tainties
[2]	✓						
[3]	✓	✓		✓	✓		
[4]	✓		✓				
[5]		✓					
[6]	✓	✓					✓
[7]			✓				
[8]			✓				
[9]			✓	✓			
[10]	✓	✓		✓	✓		
[11]		✓				✓	
[12]		✓			✓		
[13]		✓			✓	✓	
Ours	✓	✓	✓	✓	✓	✓	✓

Reference [14] proposed a comparison among different optimization approaches for traffic flow management. The performance of binary integer programming approach, genetic algorithm approach and simulated annealing approach were analyzed. Reference [15] provided a comprehensive review on meta-heuristic approaches for airside operation research. Many methods, such as genetic algorithm (GA [16]), variable neighborhood search (VNS, [17]), light propagation algorithm (LPA, [18]) and simulated annealing (SA, [6]) were compared together. It showed that new meta-heuristics should be developed to solve complicated real-world models.

In the framework of aircraft trajectory optimization, in order to avoid conflicts between aircraft and improve aircraft trajectories, several different actions can be used. The first commonly used action is to modify the departure time for aircraft [2], [19]. It is very effective to avoid conflicts by modifying the aircraft departure time locally because it does not change the route of the flight. However, in the case of high traffic demand, the delayed aircraft still needs to wait for many other aircraft to meet the capacity constraints of all airspace sectors [3].

The second action is to modify the shapes [4]–[6]. In [4], they built a model for the air traffic flow management problem with restricted capacities in airspace and airports. They proved that the complexity of the problem is NP hard and solved the problem on large-scale instances with several thousand flights. [5] proposed a local continuous method for solving air traffic conflict problems by using B-splines to modify trajectory shapes. In their method, only one optimization variable is needed to determine the trajectory for each aircraft. The experimental results showed that their method outperforms genetic algorithm. In addition, their method was shown to improve the global optimization method because it requires fewer iterations and fewer function evaluations. [6] proposed a strategic trajectory planning model to minimize the interaction between aircraft. The aircraft are separated by modifying the shapes of trajectories and the aircraft departure times. A set of virtual waypoints were used to determine the shapes of en-route trajectories. A heuristic algorithm based on simulated annealing and hill-climbing were proposed to solve the problem.

The third action is the speed regulation [7]–[9]. In [7] a mixed-integer optimization model for aircraft deconfliction based on speed regulation was proposed. A exact solver and a heuristic method were used to solve the problem. The former provides global optimality but needs long computation time, while the latter solves the problem locally but terminates fast. Based on this work, another model based on mixed-integer programming for avoiding aircraft conflicts were proposed [8]. The model was based on heading angle deviations and combined with another model based on aircraft speed regulation. Two models were solved sequentially by a two-step method. The experimental results showed the advantages of combining two maneuvers. [9] proposed an algorithm based on simulated annealing to detect and solve conflicts between aircraft. Three maneuvers were taken into account (speed, turn and altitude changes). Multiple objectives (the number and magnitude of maneuvers, time delays and deviations at waypoints) were analyzed.

In models of aircraft trajectory optimization, several factors are taken into account, such as the effect of winds [10], [20], [21], the avoidance of obstacles [11], [22] and the uncertainties of parameters [6], [12]. [13] focused on the aircraft trajectory planning problem where the obstacle avoidance and winds were taken into account. They proposed an algorithm based on Ordered Upwind Method based on Hamilton-Jacobi equation [23]. The optimal path was provided by their method. In order to be applied to long-path cases, the algorithm was extended to spherical coordinate systems. Reference [24] proposed a formulation for a 4D strategic air traffic management problem with convective weather uncertainty. A solution approach was provided to solve the problem. Experimental results on the real regional air traffic system in China showed the effectiveness of the approach.

In most previous research, either only few actions for avoiding conflicts and optimizing aircraft trajectories are used, or few real-world factors that affect trajectories were taken into account (see Table I). Therefore, a model that considers more actions and more factors (such as obstacles and winds) is proposed in this paper. In order to solve this model, an improved simulated annealing algorithm is also proposed.

III. Model for trajectory optimization

In this section, the mathematical model of trajectory optimization is presented. Bezier curve which is used to generate the shapes of trajectories is introduced in Section III-A. In Section III-B, the revised speed of aircraft with the influence of winds and obstacles is computed. The uncertainty of aircraft positions is discussed in Section III-C. In Section III-D, the definition and formulation of conflicts between aircraft are presented. Finally, the model is formulated in Section III-E.

A. Aircraft route design based on Bezier curve

In most previous research, flight trajectories are designed with a series of end-to-end routes [10]. However, the non-smooth points that connect these routes result in longer flight time and larger probability of conflicts near these points. Therefore, Bezier curve is used to design trajectories in this paper.

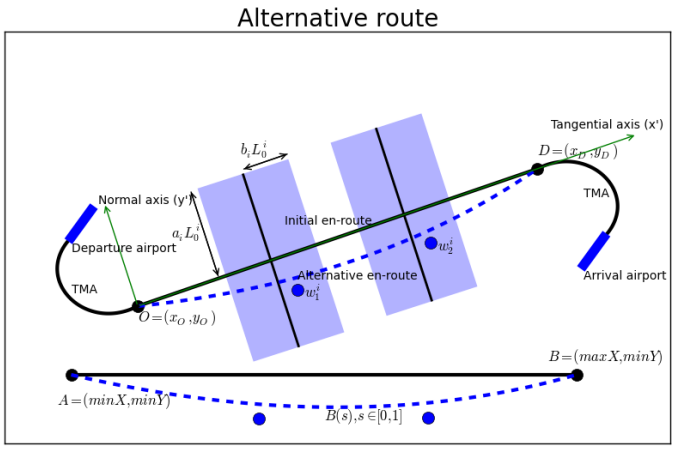


Figure 1: Allowed position for virtual points.

Bezier curve: Let $B(t)$ $(t \in [0,1])$ denote the n-degree Bezier curve that is determined by given *n+1* points $\{P_0, P_1, ..., P_n\}$. Then, we have:

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i \tag{1}$$

Route design for the en-route segment: In this subsection, we design the alternative routes for en-route segments on the horizontal profile for 4D trajectories. First, a set of virtual points are placed near the initial en-route segment. Then, the Bezier curves that determine the trajectories are constructed with these virtual points.

As shown in Figure 1, based on the initial en-route segment, the tangential axis (x') and the normal axis (y') are constructed. The coordinates of virtual points are defined using this relative coordinate system. Let $w_j^i = (w_{jx}^i, w_{jy}^i)$ be the relative coordinate of the j-th virtual point for the i-th trajectory. For each trajectory i, a set of virtual points $\{w_j^i, w_j^i = (w_{jx}^i, w_{jy}^i)\}_{j=1}^n$ is used to generate the Bezier curve, where n is the number of virtual points that is given by the user. Given a value of cruise altitude, the shape of the en-route segment in 3D space for each flight is determined based on the above Bezier curve.

B. The influences of winds and obstacles

The ground speed of aircraft is influenced by its true air speed and winds. In this paper, we use a 4D distribution function to represent the speed of the wind at each 3D coordinate (x, y, z) at time t as follows:

$$\overrightarrow{v_w}(x,y,z,t) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{2}$$

where a_{ij} $(i = 1,2,3, j = 1,2,3,4)$ and b_j $(j = 1,2,3)$ are given constants.

As shown in Figure 2(a), the ground speed ($\overrightarrow{v_f}$) is the linear combination of the true air speed of aircraft ($\overrightarrow{v_o}$) and the speed of the wind ($\overrightarrow{v_w}$). Thus, we have

$$\overrightarrow{v_f} = \overrightarrow{v_w} + \overrightarrow{v_o} \tag{3}$$

The ground speed of aircraft is along the tangent direction of the Bezier curve. Thus, we use α to denote the angle between $\overrightarrow{v_f}$ and $\overrightarrow{v_w}$. Regarding cosine law, we have

$$v_f^2 + v_w^2 - v_o^2 = 2 v_f v_w cos\alpha \tag{4}$$

where $v_f = \|\overrightarrow{v_f}\|$. Therefore, the value of the ground speed with the influence of wind is computed as follows:

$$v_f = v_w cos\alpha + \sqrt{v_o^2 - v_w^2 sin^2\alpha} \tag{5}$$

In the real-world air transportation, obstacles appear in some areas (bad weather areas, thunderstorms, etc.). When aircraft flies through these areas, it is in danger and the trajectory is penalized. For simplification, we use circles to denote obstacles. In this paper, we use homotopy to construct the obstacles that move with time. As shown in Figure 2(b), we design an initial state (at time t_0^i) and a final state (at time t_1^i) for each obstacle i. The initial state of the obstacle i is determined by the horizontal coordinate of its center (Obs_{0x}^i, Obs_{0y}^i), the radius of the (red) inner circle r_0^i and the radius of the (blue) outer circle R_0^i. Similarly, the final state is also determined by (Obs_{1x}^i, Obs_{1y}^i), r_1^i and R_1^i. Thus, the state of the obstacle at time t is computed, thanks to a linear homotopy, as follows:

$$\begin{pmatrix} Obs_{tx}^i \\ Obs_{ty}^i \\ r_t^i \\ R_t^i \end{pmatrix} = \frac{t_1 - t}{t_1 - t_0} \begin{pmatrix} Obs_{0x}^i \\ Obs_{0y}^i \\ r_0^i \\ R_0^i \end{pmatrix} + \frac{t - t_0}{t_1 - t_0} \begin{pmatrix} Obs_{1x}^i \\ Obs_{1y}^i \\ r_1^i \\ R_1^i \end{pmatrix} \tag{6}$$

The area in the inner circle is too dangerous for aircraft to fly though. If an aircraft j passes through the area between the inner circle and the outer circle at time t, there is a penalization $\xi_i^j(x,y,t)$ of the aircraft trajectory which depends on the distance between the aircraft and the center of the circle.

$$\xi_i^j(x,y,t) = \frac{(r_t^i)^2}{(x_j - Obs_{tx}^i)^2 + (y_j - Obs_{ty}^i)^2}, \\ (x_j - Obs_{tx}^i)^2 + (y_j - Obs_{ty}^i)^2 \leq (R_t^i)^2 \tag{7}$$

Therefore, the global distribution function $\xi^j(x,y,t)$ for the penalization of aircraft trajectory j is denoted by the maximum value of $\xi_i^j(x,y,t)$ among all obstacles encountered by the aircraft as follows:

$$\xi^j(x,y,t) = \max\{\xi_i^j(x,y,t), i \in \{1,2,...,N_{obs}\}\} \tag{8}$$

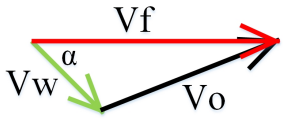
where N_{obs} is the number of obstacles. The value of $\xi^j(x,y,t)$ is larger than 1 if the aircraft j is inside the inner circle of any obstacle. The aircraft trajectory will be penalized with a very large value in this case. If $\xi^j(x,y,t) < 1$, the penalization is the summation of $\xi^j(x,y,t)$ with time.

$$X_j = \int_t \xi^j(x,y,t) dt \tag{9}$$

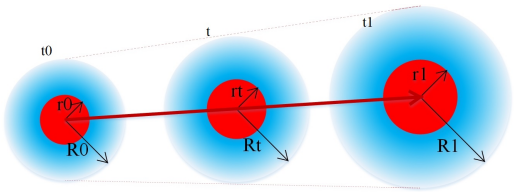
For simplification, we divide the time into many time steps with the same size Δt and discretize Equation (9) as follows:

$$X_j = \sum_k \xi^j(x_k, y_k, t_k) \Delta t \tag{10}$$

An example for the influence of winds and obstacles in the horizontal plane is shown in Figure 3 for which 8,310 flights are assigned in the area. The green arrows are used to represent the wind at each position and the length of each arrow is proportional to the speed of the wind. The flight trajectories prefer to avoid the (red) center areas of obstacles and follow the directions of winds.



(a) The computation of the ground speed ($\overrightarrow{v_f}$) of aircraft.



(b) One obstacle that moves with time.

Figure 2: The computation of aircraft speed with winds and the construction of moving obstacles. (a) The ground speed (v_f) is the linear combination of the true air speed of aircraft (v_o) and the speed of the wind (v_w). (b) The obstacle at any time t is constructed by the given initial state (t_0) and the final state (t_1).

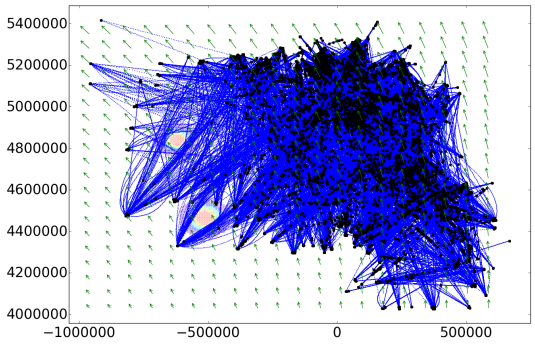


Figure 3: The influence of obstacles and the winds in the horizontal plane. The trajectories, virtual points and OD pairs are represented by blue curves, red points and black dots. Several circles are used to denote obstacles. The winds are represented by green arrows. All trajectories avoid the (red) center areas of obstacles with penalization $\xi = 1$.

C. The uncertainties of aircraft positions

In the real world, the states of aircraft are influenced by some unexpected events, which causes the deviation between the real positions of aircraft and the planned trajectories. In this paper, we use an uncertainty radius R_{un} to denote the uncertainties of aircraft positions. Assume that the planned position of an aircraft at time t is (x, y, z). The possible position of the aircraft in reality could be any point in a circle with center (x, y, z) and radius R_{un} in the horizontal:

$$\{(x_{un}, y_{un}, z_{un}) : (x_{un} - x)^2 + (y_{un} - y)^2 \leq R_{un}, z_{un} = z\} \tag{11}$$

where (x_{un}, y_{un}, z_{un}) is the possible position with uncertainties. Let N_h and N_v be the minimum distance for separation in the horizontal plane and the vertical direction. The modified parameters with uncertainties are computed as follows:

$$N_h^{un} = N_h + R_{un} \tag{12}$$

$$N_v^{un} = N_v \tag{13}$$

In our experiment, only cruising aircraft have been considered and no uncertainty has been used in the vertical dimension.

D. The conflicts between aircraft

Aircraft are in conflict if the minimum separation requirement (5NM horizontally and 1000 feet vertically) is not satisfied, i.e., more than one aircraft compete for the same space at the same period of time [6]. For the k-th point P_{ik} of trajectory i, the number of conflicts at this point is denoted by Φ_{ik}, which is the total number of times that the

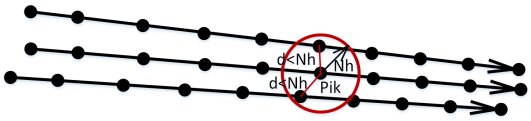


Figure 4: Conflicts $\Phi_{ik} = 2$ at point P_{ik} of trajectory i.

minimum separation requirement around P_{ik} is violated. For the example shown in Figure 4, there are $\Phi_{ij} = 2$ conflicts in the horizontal plane at point P_{ik}. The number of conflicts for a trajectory i is computed as follows:

$$\Phi_i = \sum_{k=1}^{K_i} \Phi_{ik} \tag{14}$$

Each conflict is computed once by both trajectories. Thus, the total number of conflicts between all trajectories is computed as follows:

$$\Phi_{total} = \sum_{i=1}^{N} \Phi_i/2 = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \Phi_{ik}/2 \tag{15}$$

E. Model formulation

In this subsection, we formulate the trajectory optimization problem. The parameters and variables are summarized in Table II.

Objective: We need to minimize four components in the objective function: the total number of conflicts, the total flight time for all aircraft, the total speed changes for all aircraft and the total penalization of trajectories by obstacles.

Given data: As input, we have N initial discretized 4D trajectories, the number of allowed virtual points for generating Bezier curves (M), the allowed range for determining the positions of virtual points ($0 \leq a_i \leq 1, 0 \leq b_i \leq 1$), the maximum allowed advanced and delayed shifts of the departure time for flight i ($\Delta t_a^i < 0, \Delta t_d^i > 0$), the maximum increasing and decreasing ratios of speed for aircraft ($\Delta v_a^i \in (-1, 0), \Delta v_d^i > 0$), as well as the highest and lowest altitude for aircraft (alt_{max}^i, alt_{min}^i).

Decision Variables: To optimize aircraft trajectories, we use the following variables: the shift of departure time for each flight i (δ_i), the relative coordinates of a set of virtual points for each flight i ($w^i = \{w_j^i | w_j^i = (w_{jx}^i, w_{jy}^i)\}_{j=1}^n$), the cruise altitude of each flight i (alt_i) and the cruise speed of each flight i (v_i). For simplification, we use vectors $\overrightarrow{w} = (w^i)_{i=1}^N$, $\overrightarrow{\delta} = (\delta_i)_{i=1}^N$, $\overrightarrow{alt} = (alt_i)_{i=1}^N$ and $\overrightarrow{v} = (v_i)_{i=1}^N$ to

TABLE II. The parameters and variables for the formulation.

Parameter	Description
N, n, N_{obs}	The number of the flights, the allowed virtual points for each trajectory and the obstacles.
N_h, N_v	The minimum distance for separation in the horizontal plane and in the vertical direction.
$\Delta t_a^i < 0, \Delta t_d^i > 0$	The maximum allowed advanced and delayed shifts of the departure time for each flight i.
$\Delta v_a^i \in (-1, 0), \Delta v_d^i > 0$	The maximum increasing and decreasing ratios of speed for aircraft.
alt_{max}^i, alt_{min}^i	The highest and lowest cruise altitude for aircraft.
L_0^i	The length of the initial en-route segment of each flight i.
v_0^i	The initially cruise speed of flight i.
λ_t	The coefficient that standardizes the total flight time in the objective function.
λ_v	The coefficient that standardizes the total changes of aircraft speed in the objective function.

Variable	Description
δ_i	The shift of departure time for each flight i.
$w^i = \{w_j^i \mid w_j^i = (w_{jx}^i, w_{jy}^i)\}_{j=1}^n$	The relative coordinates of a set of virtual points for each flight i.
alt_i	The cruise altitude of each flight i.
v_i	The cruise speed of each flight i.

represent these variables. Therefore, the decision variables of the problem is denoted by $(\overrightarrow{w}, \overrightarrow{\delta}, \overrightarrow{alt}, \overrightarrow{v})$.

Constraints: The following constraints should be satisfied in the model.

1) Allowed positions for virtual points: To reduce the search space and avoid undesirable shapes of trajectories, the location of each virtual point for Bezier curves are bounded. Assume that L_0^i is the length of the initial en-route segment of trajectory i. As shown in Figure 1, for each virtual point j of trajectory i, the relative coordinate (w_{jx}^i, w_{jy}^i) is restricted as follows:

$$w_{jx}^i \in \left[(\frac{j}{1+n} - b_i)L_0^i, (\frac{j}{1+n} + b_i)L_0^i\right] \tag{16}$$

$$w_{jy}^i \in \left[-a_i L_0^i, a_i L_0^i\right] \tag{17}$$

where $0 \leq a_i \leq 1$ and $0 \leq b_i \leq 1$. In addition, two areas must not overlap with each other, i.e.,

$$(\frac{j}{1+n} + b_i)L_0^i < (\frac{j+1}{1+n} - b_i)L_0^i \tag{18}$$

Thus, we have $b_i < \frac{1}{2(n+1)}$. For simplification, we divide each blue area into $(n_{px}+1) \times (n_{py}+1)$ grids in Figure 1 and the vertices of these grids are alternative coordinates for virtual points. The Equations (16–17) are discretized as follows:

$$W_{jx}^i := \left\{\frac{n_{px}-l}{n_{px}}(\frac{j}{1+n} - b_i)L_0^i + \frac{l}{n_{px}}(\frac{j}{1+n} + b_i)L_0^i\right\}_{l=0}^{n_{px}} \tag{19}$$

$$W_{jy}^i := \left\{(\frac{2l}{n_{py}} - 1)a_i L_0^i\right\}_{l=0}^{n_{py}} \tag{20}$$

where $(n_{px}+1)$ and $(n_{py}+1)$ are the numbers of possible coordinates for each virtual point in the tangential axis and normal axis.

2) Allowed shifts of departure time: It is unreasonable to shift the departure time for a large range. The value of variable δ_i is restricted to a range $[\Delta t_a^i, \Delta t_d^i]$. This range is discretized as follows:

$$\Delta_t^i = \Delta t_a^i + \frac{l}{n_t}(\Delta t_d^i - \Delta t_a^i), l \in \{0, 1, ..., n_t\} \tag{21}$$

where n_t is the number of discretization.

3) Allowed shifts of aircraft speed: The shift ratio of aircraft speed is restricted to a range $[\Delta v_a^i, \Delta v_d^i]$ and is discretized with a given number n_v:

$$\Delta_v^i = \Delta v_a^i + \frac{l}{n_v}(\Delta v_d^i - \Delta v_a^i), l \in \{0, 1, ..., n_v\} \tag{22}$$

4) Allowed shifts of cruise altitude: The shift of cruise altitude is restricted to a range $[alt_{min}^i, alt_{max}^i]$ and is discretized with a given number n_{alt}:

$$\Delta_a^i = alt_{min}^i + \frac{l}{n_{alt}}(alt_{max}^i - alt_{min}^i), l \in \{0, 1, ..., n_{alt}\} \tag{23}$$

Finally, we formulate the aircraft trajectory optimization problem as follows:

$$\min_{(\overrightarrow{w}, \overrightarrow{\delta}, \overrightarrow{alt}, \overrightarrow{v})} \Phi(\overrightarrow{w}, \overrightarrow{\delta}, \overrightarrow{alt}, \overrightarrow{v}) + \lambda_t \sum_{i=1}^{N} T_i(\overrightarrow{w}, \overrightarrow{\delta}, \overrightarrow{alt}, \overrightarrow{v})$$
$$+\lambda_v \sum_{i=1}^{N} |v_i/v_0^i - 1| + \lambda_o \sum_{i=1}^{N} X_i \tag{24}$$

$$\text{subject to } w_{jx}^i \in W_{jx}^i,\ j = 1, ..., n, i = 1, ..., N \tag{25}$$

$$w_{jy}^i \in W_{jy}^i,\ j = 1, ..., n, i = 1, ..., N \tag{26}$$

$$\delta_i \in \Delta_t^i,\ i = 1, ..., N \tag{27}$$

$$\frac{v_i - v_0^i}{v_0^i} \in \Delta_v^i,\ i = 1, ..., N \tag{28}$$

$$alt_i \in \Delta_a^i,\ i = 1, ..., N \tag{29}$$

Here, parameters λ_t, λ_v and λ_o are the weighted coefficients that transform the total flight time, the total changes of aircraft speed and the total obstacle penalization into the number of conflicts in the objective function, respectively. The flight time of each trajectory i is represented by T_i. Equations (25–26) are the constraints on the positions of virtual points on the x-axis and y-axis. Equations (27–29) are the constraints on the shifts of departure time, shifts of aircraft speed and shifts of cruise altitude, respectively.

IV. Methodology

In this section, we propose a methodology to compute the total number of conflicts between aircraft and the total flight time for all trajectories, as well as the simulated annealing algorithm for improving the aircraft trajectories.

A. Method for recording and transforming the shapes of en-route segments using standard Bezier curves

Regarding Figure 1, the shape of the alternative route is determined by the relative coordinates of the virtual points. The number of possible relative coordinates for each virtual point in the tangential and normal axis is restricted by Equations (19–20), i.e., $(n_{px}+1)(n_{py}+1)$ cases. Therefore, if the number of virtual points for the Bezier curve is small enough, a set of standard Bezier curves can be pre-generated before starting the program and they can be transformed to design the shape for a particular flight.

Assume that the minimum/maximum values of the whole area in x-axis and y-axis are represented by $minX/maxX$ and $minY/maxY$. We use the coordinates $A = (minX, minY)$ and $B = (maxX, minY)$ as the standard origin-destination (OD) pair (see Figure 1). The standard Bezier curves with all possible relative coordinates are pre-generated between them.

For a given flight with OD pair $O = (x_O, y_O), D = (x_D, y_D)$, assume that the corresponding standard Bezier curve for the current trajectory is $B(s), (s \in [0,1])$. The rotation angle between the standard curve and the required curve is computed as follows:

$$\beta = \arccos \frac{\overrightarrow{OD} \cdot \overrightarrow{AB}}{|\overrightarrow{OD}||\overrightarrow{AB}|}$$

The absolute coordinate of a waypoint with $s = t$ is computed as follows:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \frac{|\overrightarrow{OD}|}{|\overrightarrow{AB}|} \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \left(B(t) - \begin{pmatrix} minX \\ minY \end{pmatrix} \right) + \begin{pmatrix} x_O \\ y_O \end{pmatrix}$$

B. Method for computing flight time and constructing grids

The second component of the objective function is the total flight time of all trajectories. We first discretize the airspace into a 4D grids (3D space + time). The size of each grid cell is defined by the minimum separation requirement (N_h for x-axis and y-axis, N_v for z-axis) and a given time range with length Δt. Then, for each trajectory i, we update the 4D coordinate (x, y, z, t) for each point of the grid based on Bezier curves: First, we compute the position and the corresponding aircraft speed for the current time; Then, the next trajectory sample on the Bezier curve is computed with the updated travel length. The grid cell which contains the current trajectory samples is recorded. The algorithm terminates until each aircraft arrives at the destination or gets into the forbidden area of obstacles ($\xi(x, y, z, t) = 1$).

C. Method for computing the number of conflicts

To evaluate the first component of the objective function, we need to compute the number of conflicts between N aircraft trajectories. The simplest method is to do the pair-wise comparisons with a time complexity of $O(N^2)$. However, this method cannot be scaled up. Therefore, we use a grid-based method which was proposed by [6].

After finishing recording the states of grids for all trajectories with the method described in Section IV-B, we compute the conflicts for each trajectory based on these grids. For each trajectory sample P_{ik}, let $(g_{Jx}, g_{Jy}, g_{Jz}, g_{Jt})$ be the corresponding grid cell coordinate associated to P_{ik}, where (Jx, Jy, Jz, Jt) are the corresponding indices of the cell. Then, we check the (3^3) neighborhood grid cells in the 3D space including the original cell itself. Then, all the other trajectory samples in these 27 grid cells are checked. The horizontal distance d_h and the vertical distance d_v between P_{ik} and these trajectory samples are computed. As shown in Figure 5(a–b), if $d_h < N_h$ and $d_v < N_v$, one conflict is recorded. In addition, as shown in Figure 5(c–d), some conflicts between two neighbored trajectory samples may not be detected. We use an inner-loop algorithm [6] to check the conflicts in this case. Several trajectory samples are interpolated in the time dimension. Then, each pair of these trajectory samples are checked until one conflict is identified.

D. The improved simulated annealing algorithm for improving aircraft trajectories

In this section, we propose an improved simulated annealing algorithm to optimize aircraft trajectories (see Figure 6). For each iteration, we select a trajectory according to its performance (in terms of conflict, etc) and change its associated decision. We divide the algorithm into two phases: The first phase for reducing the conflicts when the total number of conflicts $\sum_{i=1}^{N} \Phi_i > 0$ and the second phase for reducing the flight time when $\sum_{i=1}^{N} \Phi_i = 0$. The values of parameter Φ_i and parameter $Wei_i = \Phi_i + \frac{\lambda_t v_0^i T_i}{L_0^i} + \lambda_v(v_i/v_0^i - 1) + \lambda_o X_i$ are used for the roulette (in the roulette, there is a mutually-exclusive range proportional to Φ_i or Wei_i for each trajectory i. For each iteration, a value is generated randomly. If the value is in the i-th range, then trajectory i is selected.). For the selection of decision variables, four values of probabilities P_0, P_1, P_2, P_3 whose sum is equal to 1 are given initially. A random number p_r in $[0, 1]$ is generated and one decision variable is selected as follows:

$$\text{Select } \overrightarrow{w}, \text{ if } p_r < P_0; \tag{30}$$

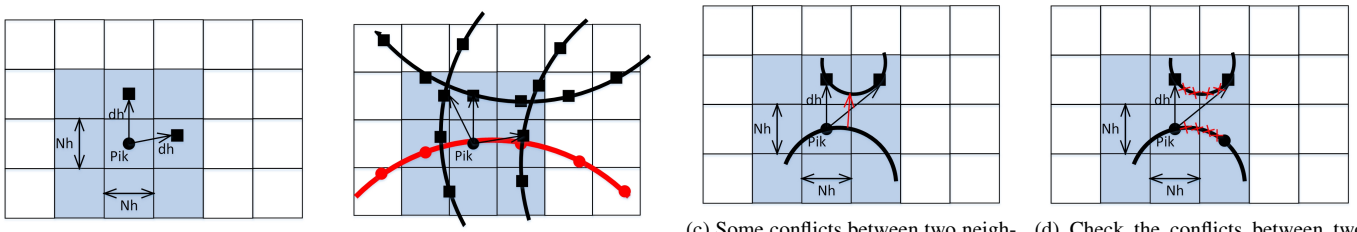
$$\text{Select } \overrightarrow{\delta}, \text{ if } P_0 \le p_r < P_0 + P_1; \tag{31}$$

$$\text{Select } \overrightarrow{alt}, \text{ if } P_0 + P_1 \le p_r < P_0 + P_1 + P_2; \tag{32}$$

$$\text{Select } \overrightarrow{v}, \text{ if } P_0 + P_1 + P_2 \le p_r \le 1. \tag{33}$$

We call this strategy for selecting decision variables, **Fixed Strategy**.

Note that there is little effect of cruise altitude on the flight time of an individual trajectory. It is more likely to obtain better solutions when trajectories are evenly distributed in different cruise altitudes. We divide the set of cruise altitudes of all trajectories into several groups. The average number of trajectories in each group is Ave_{Distri}. For a selected trajectory i, the number of trajectories that are in the same altitude group as i is represented by $Distri(alt_i)$. If the value of $Distri(alt_i)$ is much larger than Ave_{Distri}, it is better to modify the cruise altitude for trajectory i. The probabilities



(a) Detect the conflicts in the horizontal plane by checking all neighborhood cells.

(b) Compute the number of conflicts for one trajectory sample (black dot) with all other trajectories.

(c) Some conflicts between two neighbored trajectory samples may not be detected (the red arrow). It is necessary to check them.

(d) Check the conflicts between two neighbored trajectory samples using time interpolations. Several trajectory samples (red crosses) are generated.

Figure 5: The detection of conflicts in the horizontal plane [6].

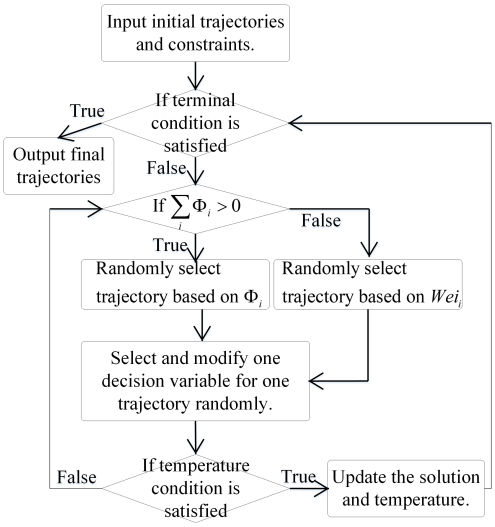


Figure 6: The flow chart for the improved simulated annealing algorithm.

for selecting decision variables are updated as follows:

$$P_{0new} = \frac{(1-\lambda)P_0}{P_0+P_1+P_3} \tag{34}$$

$$P_{1new} = \frac{(1-\lambda)P_1}{P_0+P_1+P_3} \tag{35}$$

$$P_{2new} = \lambda P_2 \tag{36}$$

$$P_{3new} = \frac{(1-\lambda)P_3}{P_0+P_1+P_3} \tag{37}$$

where $\lambda = \log(\frac{Distri(alt_i)}{Ave_{Distri}} + 1) + 1 - \log 2$. We call this strategy, **Even-Altitudes Strategy**.

Next, we compute the new state of grid cells that contain the samples of trajectory i, as well as the new number of conflicts of trajectory i and all other trajectories that have conflicts with trajectory i. The new number of conflicts of trajectory i is represented by Φ_i^c. Then, we compute the new flight time T_i, the current cruise speed for trajectory i and the new value of the objective function Obj^c. We compute the change of temperature $dTem$ as follows: $dTem = \sum_{i=1}^{N} \Phi_i^c - \sum_{i=1}^{N} \Phi_i$ if there still exist conflicts and $dTem = Obj^c - Obj$ otherwise. If $exp(\frac{dTem}{Tem}) < Ps$ ($Ps = 2$ if $\sum_{i=1}^{N} \Phi_i > 0$ and $Ps = 1$ otherwise.) is satisfied, we perform the operation and update the solutions. Otherwise, we return to the current solution. Finally, we update the temperature with $Tem = 0.99 * Tem$.

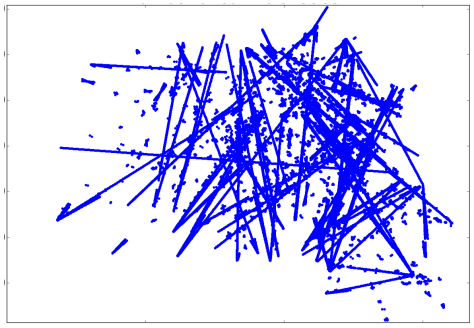


Figure 7: The distribution of 19,816 conflicts for the initial trajectories.

V. Computational Results

A. Datesets and experimental setup

To evaluate the performance of our model and algorithm, a national-size dataset which is provided by a fast time simulator (ΠRATS) is used as a case study. The dataset consists of 8,310 trajectories over the French airspace corresponding to the demand of August 17th, 2008. Because only the cruise phase is considered in our model, we filtered and modified the initial trajectories in the dataset, i.e., only the flights with cruise phases are remained and the other phases of these trajectories are removed. Finally, 8,310 trajectories are selected for the experiments.

To solve this instance, we implemented our improved simulated annealing algorithm with Python language on Fedora system with 2.7 GHz processor and 16 GB RAM. The settings of parameters in the experiments are shown in Table III. As shown in Figure 7, there are 19,816 conflicts between the initial trajectories.

B. The comparison of two strategies for selecting decision variables

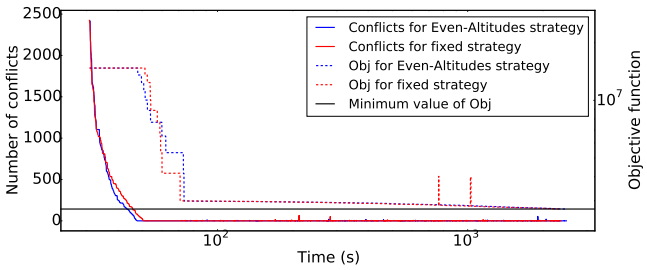
Two strategies (Even-Altitudes Strategy and Fixed Strategy) for selecting decision variables are proposed in Section IV-D. In this section, their performance for improving aircraft trajectories are compared. The dataset with $N = 1,115$ trajectories whose departure times are between 10AM and 12AM is used as a case study. $N_{obs} = 4$ moving obstacles

TABLE III. The parameter settings

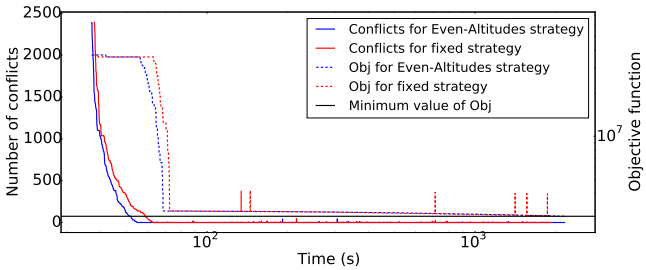
Parameter	Value
The number of obstacles, N_{obs}	4
The minimum distance for separation in the horizontal plane, N_h	5 Miles
The minimum distance for separation in the vertical plane, N_v	1000 feet
The maximum allowed advanced and delayed shifts of the departure time, $-\Delta t_a^i = \Delta t_d^i$	3600 seconds
The maximum increasing and decreasing ratios of speed for aircraft, $-\Delta v_a^i = \Delta v_d^i = \Delta v$	0.1
The coefficient that normalizes the total travel times in the objective function, λ_t	1
The coefficient that normalizes the total changes of aircraft speed in the objective function, λ_v	2
The maximum number of virtual points for Bezier curves, M	2
The allowed range for shifting the positions of virtual points, $a_i = b_i$	0.25
The number of alternative relative coordinates in tangential axis and normal axis, $n_{px} = n_{py}$	10
The size of time grid, Δt	20 seconds
The changed ratio of speed, dv	0.01
The size of altitude grid, da	1000 feet
The probability for modifying the positions of virtual points, P_0	0.3
The probability for modifying the positions of departure time, P_1	0.3
The probability for modifying the positions of cruising altitude, P_2	0.2
The probability for modifying the positions of cruising speed, P_3	0.2
The temperature condition	$Tem \leq Tem_{init}/10$

TABLE IV. The comparison of two strategies for selecting decision variables for solving the instances with moving/fixed obstacles. The dataset consists of $N = 1,115$ trajectories whose departure times are between 10AM and 12AM.

Obstacle type	Strategy	Initial number of conflicts	Number of iterations for zero conflicts	Run time for zero conflicts (s)	Total number of iterations	Total run time (s)	Objective function
Moving obstacles	Fixed strategy	2,453	558	50.7	45,022	2,363.7	2,710,839.4
	Even-Altitudes Strategy	2,453	448	47.7	47,182	2,484.2	2,708,967.9
Fixed obstacles	Fixed strategy	2,453	595	62.8	31,393	1,922.7	2,770,344.2
	Even-Altitudes Strategy	2,453	418	55.1	34,748	2,157.1	2,746,639.3



(a) The instance with moving obstacles.



(b) The instance with fixed obstacles.

Figure 8: The evolution of conflict numbers and objective functions of two strategies on both instances. It needs short time to reduce the number of conflicts to zero. Most run time is spent in reducing other criteria when conflicts are already equal to 0.

are generated randomly. We also solve the instance with fixed obstacles. The results are shown in Table IV. The initial number of conflicts is 2,453. The number of iterations and run times for obtaining zero conflicts are shown in columns 4–5. The total number of iterations and the total run times are shown in columns 6–7. The final values of objective functions are shown in column 8. The evolution of conflict numbers and objective functions of the two strategies on both problem instances is visualized in Figure 8. We have following observations:

1) Both strategies reduce the number of conflicts to zero for both problem instances.
2) Only about 1%–2% of iterations and run time are used for reduce conflicts to zero. Most of the run time is spent improving trajectories (reducing flight time and speed changes).
3) Both strategies need similar run time but Even-Altitudes Strategy provides better solutions.

C. The performance on large-scale instances

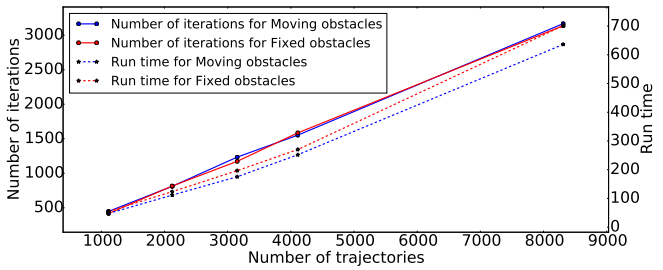
TABLE V. Instances in other time ranges with different numbers of trajectories.

Instance	Number of trajectories
10AM-12AM	1,115
10AM-2PM	2,122
10AM-4PM	3,151
10AM-6PM	4,110
0AM-12PM	8,310

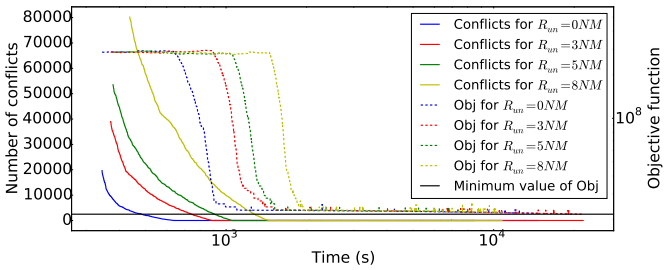
In this section, in order to show the scalability of our model and algorithm, we run the algorithm with Even-Altitudes Strategy on larger instances. As shown in Table V, the flights in several time ranges (10AM-12AM, 10AM-2PM, 10AM-4PM, 10AM-6PM, 0AM-12PM) are selected. Both instances with moving obstacles and fixed obstacles are used as case studies. The number of iterations and run time for obtaining zero conflicts for each instance are shown in Figure 9(a). With increasing number of trajectories, the number of iterations and

TABLE VI. The results for the instances with moving/fixed obstacles obtained by Even-Altitudes Strategy on the whole dataset which consists of $N = 8,310$ trajectories. The maximum run time is set to 6 hours.

	Initial number of conflicts	Number of iterations for zero conflicts	Run time for zero conflicts	Total number of iterations	Total run time (s)	Objective function
Moving obstacles	19,816	3,167	636.2	126,763	14,668.3	22,020,868.4
Fixed obstacles	19,816	3,137	701.2	175,660	21,600.1	21,594,453.1



(a) The number of iterations and run time for obtaining zero conflicts for instances in Table V with moving/fixed obstacles. The Even-Altitudes Strategy is used here. With increasing number of trajectories, the number of iterations and run time for obtaining zero conflicts increase almost linearly ($T = 0.00497 \times N^{1.308}$, $R^2 = 0.9942$, $I = 0.409 \times N^{0.992}$, $R^2 = 0.9996$, where N, T, I are the trajectory number, run time and iteration number, respectively).



(b) The evolution of conflict numbers and objective functions for the instances with moving obstacles and different values of uncertainty radius obtained by Even-Altitudes Strategy on the whole dataset which consists of $N = 8,310$ trajectories. The maximum run time is set to 6 hours.

Figure 9: (a) The performance of the algorithm for obtaining zero conflicts for different instances; (b) The evolution of conflict numbers and objective functions on the whole dataset with $N = 8,310$ trajectories.

TABLE VII. The results for the instances with moving obstacles and different values of uncertainty radius obtained by Even-Altitudes Strategy on the whole dataset which consists of $N = 8,310$ trajectories. The maximum run time is set to 6 hours.

The uncertainty radius	Initial number of conflicts	Number of iterations for zero conflicts	Run time for zero conflicts (s)	Total number of iterations	Total run time (s)	Objective function
$R_{un} = 0$ NM	19,816	3,167	636.2	126,763	14,653.6	21,620,868.4
$R_{un} = 3$ NM	39,110	5,192	889.8	192,760	21,600.1	21,720,092.3
$R_{un} = 5$ NM	53,512	6,715	1,053.8	75,634	8,446.3	22,498,778.6
$R_{un} = 8$ NM	80,237	9,046	1,451.3	96,130	10,743.6	22,438,602.7

run time for obtaining zero conflicts increase almost linearly. In addition, we run the algorithm on the whole dataset with $N = 8,310$ trajectories with the maximum run time setting to 6 hours. The results are shown in Table VI.

Therefore, on one hand, for the objective to deconflict aircraft, our algorithm is almost linearly scalable for large-scale instances; on the other hand, for the objective to further improve trajectories, our algorithm provides good solutions with a given limited run time.

D. The uncertainties of aircraft positions

In this section, we take the uncertainty of aircraft positions into account. The values of N_h and N_v are updated by the uncertainty radius R_{un} regarding Equations (12–13) in Section III-C. Here, we still use the whole dataset with $N = 8,310$ trajectories and the moving obstacles as a case study. The values of uncertainty radius are set to $R_{un} \in \{0\ NM, 3\ NM, 5\ NM, 8\ NM\}$. The maximum run time is still set to 6 hours. The results are shown in Table VII and Figure 9(b).

The initial number of conflicts are shown in column 1. With increasing values of R_{un}, more conflicts appear between aircraft initially and more iterations and run time are used to obtain zero conflicts. Therefore, the algorithm uses up 6 hours run time to provide good solution for the case with $R_{un} = 3\ NM$. However, for the cases with $R_{un} = 5\ NM$ and $R_{un} = 8\ NM$, the algorithm finds conflict-free trajectories with more deviations. It shows that the conflicts caused by the uncertainty radius is so large that trajectories cannot be improved further.

VI. Conclusions

In this study, we proposed an aircraft trajectory optimization model based on Bezier curve. The influences of winds and obstacles are also taken into account. The real speed of aircraft at each time is affected by the winds. In order to increase robustness of trajectories, the uncertainty of aircraft position was also considered. To reduce the conflicts between aircraft and improve trajectories, we can modify the departure time, the trajectory shape, the aircraft speed and the altitude for each trajectory. The proposed methodology takes into account many realistic factors of the airspace (fixed/moving obstacles, winds, etc.) To solve the model, an improved simulated annealing algorithm with two phases was proposed. A national-size dataset with 8,310 trajectories which is provided by a fast time simulator (ΠRATS) is used as a case study. Our algorithm provides conflict-free trajectories within very short time for all instances (for instance, it needs 10-12 minutes for the instance with 8,310 trajectories without uncertainty). In addition, for the objective to deconflict aircraft, the algorithm is almost linearly scalable for large-scale instances. For a

given limited run time, the algorithm provides good solutions with small values of objective function (total flight time and changes of aircraft speed). Future work could perform sensitivity analysis regarding the parameters used in this study, for instance, spatio-temporal resolution of grids and probabilities for performing actions/operations. Moreover, in order to increase the performance of the simulated annealing algorithm further, several domain-specific search heuristics could by tried for turning the blind search into a more informed search.

ACKNOWLEDGEMENT

This study is supported by the Research Fund from National Natural Science Foundation of China (Grants No. 61861136005, No. 61601013, No. 61851110763, No. 71731001).

REFERENCES

[1] A. Islami, S. Chaimatanan, and D. Delahaye, *Large-Scale 4D Trajectory Planning*. Tokyo: Springer Japan, 2017, pp. 27–47.

[2] M. Terrab and A. R. Odoni, "Strategic flow management for air traffic control," *Operations Research*, vol. 41, no. 1, pp. 138–152, 1993.

[3] I. Dhief, N. E. Dougui, D. Delahaye, and N. Hamdi, "Strategic planning of North Atlantic Oceanic air traffic based on a new wind-optimal route structure," in *12th USA/Europe Air Traffic Management R&D Seminar,*, Seattle, United States, Jun. 2017.

[4] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operations Research*, vol. 46, no. 3, pp. 406–422, 1998.

[5] C. Peyronne, A. R. Conn, M. Mongeau, and D. Delahaye, "Solving air traffic conflict problems via local continuous optimization," *European Journal of Operational Research*, vol. 241, no. 2, pp. 502 – 512, 2015.

[6] S. Chaimatanan, D. Delahaye, and M. Mongeau, "A hybrid metaheuristic optimization algorithm for strategic planning of 4d aircraft trajectories at the continental scale," *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 46–61, Nov 2014.

[7] S. Cafieri and N. Durand, "Aircraft deconfliction with speed regulation: new models from mixed-integer optimization," *Journal of Global Optimization*, vol. 58, no. 4, pp. 613–629, Apr 2014.

[8] S. Cafieri and R. Omheni, "Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes," *European Journal of Operational Research*, vol. 260, no. 1, pp. 283 – 290, 2017.

[9] A. Mateos and A. Jimnez-Martn, "Multiobjective simulated annealing for collision avoidance in ATM accounting for three admissible maneuvers," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[10] O. Rodionova, M. Sbihi, D. Delahaye, and M. Mongeau, "North atlantic aircraft trajectory optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2202–2212, Oct 2014.

[11] L. Blasi, S. Barbato, and M. Mattei, "A particle swarm approach for flight path optimization in a constrained environment," *Aerospace Science and Technology*, vol. 26, no. 1, pp. 128 – 137, 2013.

[12] K. Legrand, S. Puechmorel, D. Delahaye, and Y. Zhu, "Robust Aircraft Optimal Trajectory in the Presence of Wind," *IEEE Aerospace and Electronic Systems Magazine*, 2018.

[13] B. Girardet, L. Lapasset, D. Delahaye, and C. Rabut, "Wind-optimal path planning: Application to aircraft trajectories," in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec 2014, pp. 1403–1408.

[14] J. Rios and J. Lohn, "A comparison of optimization approaches for nationwide traffic flow," in *Management, AIAA Guidance, Navigation, and Control Conference, AIAA*, 2009.

[15] K. Ng, C. Lee, F. T. Chan, and Y. Lv, "Review on meta-heuristics approaches for airside operation research," *Applied Soft Computing*, vol. 66, pp. 104 – 133, 2018.

[16] 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.

[17] A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, and N. Mladenović, "A VNS metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes," *Journal of Global Optimization*, vol. 63, no. 3, pp. 583–596, Nov 2015.

[18] N. Dougui, D. Delahaye, S. Puechmorel, and M. Mongeau, "A light-propagation model for aircraft trajectory planning," *Journal of Global Optimization*, vol. 56, no. 3, pp. 873–895, Jul 2013.

[19] F. D. Fomeni, K. G. Zografos, and G. Lulli, "An optimization model for assigning 4D-trajectories to flights under the TBO concept," in *12th USA/Europe Air Traffic Management R&D Seminar,*, Seattle, United States, Jun. 2017.

[20] T. G. Reynolds, M. McPartland, T. Teller, and T. Seth, "Exploring Wind Information Requirements for Four Dimensional Trajectory-Based Operations," in *11th USA/Europe Air Traffic Management R&D Seminar,*, Libson, Portugal, Jun. 2015.

[21] B. Sridhar, N. Y. Chen, H. K. Ng, O. Rodionova, D. Delahaye, and F. Linke, "Strategic Planning of Efficient Oceanic Flights," in *11th USA/Europe Air Traffic Management R&D Seminar,*, Libson, Portugal, Jun. 2015.

[22] X.-B. Hu, S.-F. Wu, and J. Jiang, "On-line free-flight path optimization based on improved genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 8, pp. 897 – 907, 2004.

[23] J. A. Sethian and A. Vladimirsky, "Ordered upwind methods for static hamilton-jacobi equations," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 20, pp. 11 069–11 074, 2001.

[24] Y. Yang, "Practical method for 4-dimentional strategic air traffic management problem with convective weather uncertainty," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1697–1708, June 2018.

AUTHORS BIOGRAPHIES

Weibin Dai works as a PhD student with the School of Electronic and Information Engineering at Beihang University. He obtained Bachelor of Science in Mathematics and Applied Mathematics in Beihang University in 2015. His research interests are scalable algorithms for transportation network design problems with focus on expressive hub location problems, as well as human mobility flow analysis and prediction.

Jun Zhang was born in Anhui, China, in 1965. He received the M.S. and Ph.D. degrees from Beihang University in 1990 and 2001, respectively. He is a Professor of Beihang University, and also a member of Chinese Academy of Engineering. He has concentrated on the research and education in networked and collaborative air traffic management systems.

Daniel Delahaye is the head of the optimization and machine learning team of the ENAC Laboratory (French Civil Aviation University). He obtained his engineer degree from the ENAC school and did a master of science in signal processing from the national polytechnic institute of Toulouse in 1991. His obtained his PH.D in automatic control from the aeronautic and space national school in 1995 and did a post-doc at the Department of Aeronautics and Astronautics at MIT in 1996. He get his tenure in applied mathematics in 2012. He conducts researches on mathematical optimization for airspace design and trajectory design for strategic, pre-tactical and tactical applications. He collaborates with MIT, Georgia Tech and NASA (USA).

Xiaoqian Sun received the Ph.D. degree in aerospace engineering from Hamburg University of Technology in 2012. She is an Associate Professor with the School of Electronic and Information Engineering, Beihang University. Her research interests mainly include air transportation networks, multi-modal transportation, and multi-criteria decision analysis.