10.2 参赛论文

The Model for Measuring Complexity of Air Traffic Control Predicting and Adjusting Path Conflicts

Du Qinghe Hu Yunhua Geng Xiujuan
Advisor He Xiaoliang

Summary

Our model is based on how to measure the complexity from air traffic workload perspective. We analyze some factors concerning complexity in air traffic control, such as the number of aircraft and potential conflicts. And we give an air traffic control model to detect potential flight path conflicts and alert the controller. The results obtained from our model present a satisfactory measurement on complexity. At the same time, we can get a optimal adjust project.

We divide our model into two major parts. In the first part, we bring out a new concept about flying plane. Considering the plane as a double-cylinder model we can easily detect if two airplanes are too close and require intervention by a proximate and efficient method. According to some typical data, we obtain that the minimum distance between two planes to avoid collision is about 5 km. To guarantee more safety, we enlarge it to some extent.

In the second part, we study the complexity at three given cases. To each case, we detect potential flight path conflicts and alert the controller, and adopt different algorithms to adjust the

planes. The first one we call transient adjust model, in this model, we use both horizontal and vertical sorting adjustment. And we bring a concept "degree", which is the number of potential conflicts of one plane. We prefer adjusting the plane that has the maximum degree first. Through this principle we can reduce searching times effectively, and the solution we get is close to the optimal solution which is obtained by global searching algorithm. When seven planes cross the sector simultaneously, the time we can get the solution by computer is about 3~5(average value) seconds generally. The second one we call it dynamic adjust model by using time slice, it can improve the efficiency more. And the last one we call it particular adjust model. This model can be used to effectively handle some extreme situations when much more planes enter the airspace simultaneously.

According to our results, we think that the number of aircraft, potential conflicts and the number of planes required adjusting determine complexity. The runtime of software and total adjustment are reflection of complexity. The runtime consists of two parts, one is to judge the potential conflicts and the other is to compute the rational adjusting project. We analyze the result obtained through our algorithm, and get the changing trends of runtime and total adjustment with the number of planes and potential conflicts. We draw three curves to describe this relation. We can see that the complexity (runtime and total adjustment) would increase very rapidly with the number of planes, and more gently with the number of potential conflicts. In most cases, the additional software can reduce the complexity. We can get this conclusion through our model. If we can find more efficient algorithm, we can reduce this complexity more.

Background

In nowadays, Air traffic becomes busier and busier. And how to improve safety and reduce air traffic workload becomes more important. FAA had brought forward a new "free flight" concept according to develops trends of air traffic. This concept brings more freedom to pilots, but it had added more burden to air traffic controller. TCAS act as a system equipped by plane can reduce the conflict in airspace. But it doesn't work so well in some airspace, in which the situation is more complexity. And in this situation aircraft controller performance more important actor. The success of New York's TRACON Center is a good example. And we know that the workload of air traffic control concern about many factors, especially about the complexity of that airspace. So studying the number of plane and potential conflict, and how the software will bring affect to the air traffic controller, will have important significative on how to improve safety and reduce air traffic controller workload. To consider this, we design an optimized model to prove these relationship.

Factors Affecting Complexity

1. The number of planes in the airspace

The number of the planes in the airspace is one of the most important facts affecting complexity. It is obviously that more planes lead less space for each plane. Then the probability of flight path conflicts will rise, and it will be more difficult to adjust the planes. With the numbers of the planes rising, getting an optimal resolution to avoid conflicts become harder.

2. The number of potential conflicts

The number of flight path conflicts between every two planes is another most important factors. It does determine the air traffic workload. Assuming there are several planes simultaneously passing through the airspace. The amount of the conflicts directly determines the difficulty to predict it well and true and to acquire an ideal adjustment solution by both controller and computer. Although the numbers of planes is big, the conflicts may be less. Thus the complexity will be less, the workload will also be in a lower lever. Assuming the numbers of planes is N. We can calculate the upper bounder of conflicts. Because every two planes has the probability of conflict, the number of conflicts will be less than $C_n^2 = N(N-1)/2$. But as we know generally, the case that every planes has path conflicts will not occur, and the probability that there are not conflicts is also very low, so the conflicts lever generally remains at a average level between 0 and $N(N-1)$. If we consider the flight path as random tracks, the average number should be $N(N-1)/4$.

3. Vertical Sorting

In a general way, planes in different directions fly in different altitude levels. The neighboring level has a 300-meter-high difference. In practical, because planes fly at different altitude levels will not affect each other, assigning a set of flight level can effectively reduce the potential conflicts. And vertical adjustment is an easier and more efficient way in common uses. In some cases, this would probably add the probability of vertical conflicts to some extent, but this doesn't matter much about the complexity.

4. Angle between the Two Conflict Aircraft's Direction

For every two planes, which have path conflict, the angle between their directions can also affect the complexity and workload to some extent. Especially in recent years, FAA actively bring forward the concept "Free Flight", this new concept give more freedom for the pilot to select the flight path than before. And this may cause more planes with opposite directions appearing at the same flight altitude level. So the angle between the directions may affect the complexity.

5. Additional software

We should design a software to detect and predict the flight conflicts to help controller to judge and to reduce the air traffic load. But sometimes, the additional software will not get the right result; we call it "false alarm". If the false alarm rate is high, the complexity will arise with it; workload is heavier than real cases. So how to remain the false alarm rate at the lower lever is a major problem.

How to Measure the Complexity

1. What's air traffic workload

Assuming there are N planes simultaneously passing through the airspace sector, in order to make the planes cross the sector safely, a lot work should be done. These works include detecting and predicting the potential conflicts and acquire a proper solution to adjust the planes by human or computer. The total amount of work is **air workload**. Of course, the amount of work itself is not clear and measurable. But it can be embodied by other measurement such as **the total work time** from getting the state of the planes to find the proper solution, **the total amount of adjustment** made by planes to avoid collision. Obviously, more air workload leads more time and more adjustment.

2. The relation between complexity and air traffic workload

From workload perspective, the air traffic workload is the direct reflection of complexity. If the complexity become complex, the workload will be heavy. So we can measure complexity through the amount of workload, that is to say, we can use the total time and the total amount of adjustment to describe and measure the complexity. On one hand, complexity is mainly determined by the factors mentioned above, especially the number of the planes and the number of potential conflicts. Thus we can use the two numbers as the independent variables. On the other hand, the work time and total adjustment determine workload, which is the direct reflection of complexity. Thus we adopt them as dependent variable to represent complexity. Then we will try to find the concrete relation between them. This is one of our two major works in this paper.

3. The main procedure

The course putting our thought into practice faces mainly two problems. First, how to acquire the work time? Secondly, how to acquire the total adjustment? We are not controllers, and even the controllers can't tell the accurate data just because they judge conflicts and provide the solution of adjustment is only by his experience. Based on this fact, we design an algorithm carried out by computer to predict the potential path conflicts to alert controllers, and get a rational solution of adjustment to help controllers to adjust the planes. We use the runtime of the program as the substitute of the work time. The total adjustment is substituted by the adjustment solution made by computer. Then we design a set of data, input it to the computer, simulate the real situation, and get a set of result. Through the result we can analyze the relation mentioned above and draw a conclusion. At the same time, the adjustment solution given by computer is a rational one to solve the conflicts. So design an algorithm and write the program is our another major works in this paper. Finally we will analyze if this additional software would add or reduce the complexity.

Problems About Total Angles and Total Times of Adjustment

When the air traffic controller has to send out the adjustment message, he (or she) maybe concern about the number of planes requiring adjusting, and the adjustment angles of each plane. (Because he has to spare some airspace for planes flying into the airspace later.) In our model, we design our program for two different goals. One is for optimizing total adjustment angles, and the other is for optimizing the total adjustment times. Because the main ideas in these two programs are similar, we can easily get one algorithm from the other by few changes. That is to say, when we want to get the most optimal angles, we should search all feasible solutions and get the minimal angles for all planes. If we want to get the optimal adjustment times, we should search all feasible solutions and get the most minimal one too. And in our algorithm, we prefer more on the optimal of total angles than the times of adjustment. We design an optimal searching algorithm to solve these problems (shown in Transient Adjusting Model). And the result of all the tests show that when we get the solution of optimal angles, we have the probability more than 70% to get the optimal adjustment times at the same time. The all algorithms described, as following are all based on this consideration. So we only describe in detail how to get the optimal total angles and the optimal solution project.

Assumptions

1. We consider an airspace, Terminal Approach Control, with the altitude under 7 km. Since we think methods avoiding conflicts occurring in an airspace higher than that altitude can be concluded in our discussing range.

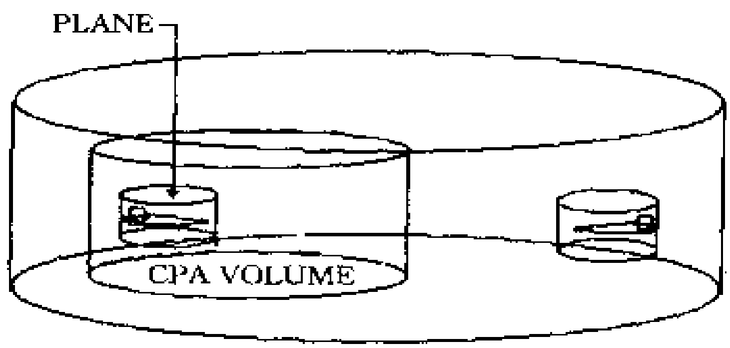
2. Assume the sector as a cuboid space, with size 160 km, altitude 7 km.

3. Airspace stratification model

Aircraft in a level fly at the same altitude, if any other plane flies into this level, it will be at the same altitude too. And we describe a clearance between any two adjacent level is 300m.

4. Double-Cylinder Model

(1) We simplify the plane as a cylinder. The assumption is easy to calculate in our later algorithms and it is suitable for the fact that the length (consider the effect of the airflow a plane generate around it) is much larger than the height of a plane. The specific data is given as follows:



R (radium of the cylinder): 600m; H (height of the cylinder): 50m.

Figure 1

(2) CPA(Closest Point of Approach) Area

We define CPA Area to be an area that once any two of them is overlap, the planes cannot avoid conflict no matter how to adjust the flight path. Also consider CPA volume as a cylinder (the followings are the same). It is an important concept. We give an algorithm to extract the size of the CPA volume. And once any two planes's CPA zone is tangent, the air traffic controller has to send alarm message to the pilots.

Place the plane at a point on the perimeter of the circle. And a plane's velocity's direction is the same as the direction of the radium through the point. The reason why we set up such a model is:

The size of the circle will be changing with two planes' relative velocity. (Assume the height of CPA volume is h, the radium of it is r. Suppose two planes fly at the speed of v_1 *and* v_2 in three conditions respectively shown in graph1.) Whether place the plane at the center of the circle or on the circle's perimeter does not affect the result of the first model very much. But when we calculate the adjust project, we consider the CPA volume as constant, and use the maximum volume to replace the changing volume, putting it on the circle seems to be more reasonable than at the center of the circle.

5. Assume only one adjustment project (horizontal or vertical adjust) allows to be made at any one instant.

6. Assume that it costs no time in getting objects' position, velocity, track, and detecting whether there are potential path conflicts or not (Since by our algorithm the runtime is very short, only about one tick.)

7. Flight path is a line in general condition. The velocity of each plane will not change during flying through an airspace sector.

8. Assume the airspace sector is a cuboid with side 160 km, height 7 km.

The more specific assumptions are given in the following specific models.

The Criterion of the Minimum Protecting Area

1. Definitions

a: the maximum acceleration centripetal.

V: the velocity of a plane.

l: the minimum distance that two planes can avoid collision.

R: the radium of the cylinder.

H: the height of the cylinder.

θ: angle described as gragh2.

P_0: the initial position of a plane.

P: the position of a plane at any instant.

V_c: the maximum climbing velocity.

V_d: the maximum descending velocity.

2. Analysis of CPA Model

1. Based on our model, we break the problem into 3 aspects. The horizontal adjustment to avoid the horizontal conflicts:

Mapping CPA into a horizontal level. At the distant l, two planes begin to change their directions at acceleration centripetal of a, the plane cylinders will be tangent at an instant during the following tracks. Further through the value of l, we can obtain the value of r. (Specific steps are given as the following algorithm.) Because in this case, the height can be any value between H and 150 m (half of the height of the vertical clearance 300m), we can suppose it as 50 m.

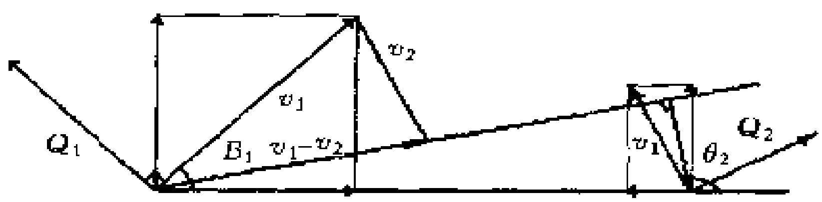


Figure 2

Subscript 0: the value at the initial time ($t = 0$)

S': relative position

V: relative position

t: time

P: position of plane at the time of t

$l = |S_0|$

$V_{x1} = V_1\cos\theta_1$; $V_{y1} = V_1\sin\theta_1$; $V_{x2} = V_2\cos\theta_2$; $V_{y2} = V_2\sin\theta_2$

$S_x = P_{x1} - P_{x2}$; $S_y = P_{y1} - P_{y2}$;

We are restricted by the condition: during the period that $t > 0$, $\min |S| > 2R$.

The two planes simultaneously change direction at acceleration centripetal of a:

$$P_{x1} = P_{x01} - \frac{V_1^2}{a_1}\left(1 - \cos\left(\frac{a_1}{V_1}t\right)\right)\sin\theta_1 + \frac{V_1^2}{a_1}\sin\left(\frac{a_1}{V_1}t\right)\cos\theta_1;$$

$$P_{y1} = P_{y01} + \frac{V_1^2}{a_1}\left(1 - \cos\left(\frac{a_1}{V_1}t\right)\right)\cos\theta_1 + \frac{V_1^2}{a_1}\sin\left(\frac{a_1}{V_1}t\right)\sin\theta_1;$$

$$P_{x2} = P_{x02} - \frac{V_2^2}{a_2}\left(1 - \cos\left(\frac{a_2}{V_2}t\right)\right)\sin\theta_2 + \frac{V_2^2}{a_2}\sin\left(\frac{a_2}{V_2}t\right)\sin\theta_2;$$

$$P_{y2} = P_{y02} - \frac{V_2^2}{a_2}\left(1 - \cos\left(\frac{a_2}{V_2}t\right)\right)\cos\theta_2 + \frac{V_2^2}{a_2}\sin\left(\frac{a_2}{V_2}t\right)\sin\theta_2;$$

We can get a minimum l that meet the condition above by Lingo software easily. It is obvious that planes with different velocity and position have different l. But we should get the l which is suit for most cases, so we give a convienient method to obtain l below.

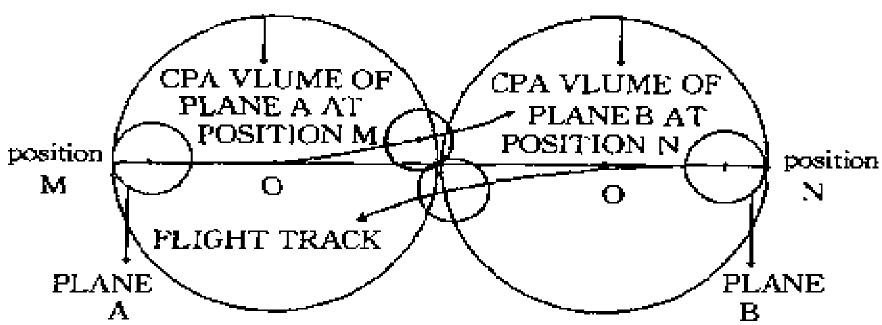


Figure 3

We assume that the two planes fly at the same speed at V, and maximum their acceleration centripetal is the same too. But their directions are opposite. Then we can calculate the minimum radius ($r_{\min}$) they turn around. From Figure 3 we can obtain an equation.

$$(\frac{1}{2})^2 + r_{\min}^2 = (R + r_{\min})^2$$

According to this equation we can get a proximate value of l. If we let $V = 800$ km/h, R = 600 m, a = 10 m/s, we will get the minimum $l \approx 5$ km.

2. The vertical adjustment to avoid the horizontal conflicts:

We assume two planes fly in the opposite direction also, their horizontal velocity is V(800 km/h). One plane climbs at the speed of V_c, another plane descends at the speed of V_d. We can get the minimum l that they have plenty time to reach an altitude difference equal to h. min $t = h/(V_c + V_d)$, thus we can obtain $l = 2tV$. We let $V_c = V_d = 10$ m/s (This parameter is typical for planes). But in fact this l is much less than 5km. So we should adopt 5km as the minimum radius of protecting area.

3. The horizontal adjustment to avoid the vertical conflicts:

If two planes have potential conflicts, so long as we make the two planes have no conflicts in the horizontal level, the two planes will not collide. So we should guarantee that when two planes fly in opposite direction with the distance of R (radius of horizontal protection area), before they are closest, they would not arrive the same altitude level. Thus we can calculate $t = R/2V$, then h could be calculate by the expression $h = t(V_c + V_d)$. We let $V_c = V_d = 13$ m/s, $R = 5$ km, then $h = 291$ m. So we use 300 m as the safety altitude difference.

Abstractly, we have given the clear expressions in three aspects. However, to simplify later calculation, we replace the changing CPA volume by the maximum value. At the same time, this replacement can guarantee the flight safety. So we think it is reasonable. We prescribe the value of r and h is respectively 8 km, and

300 m for our subsequent work. Then any planes mustn't be closer than 8 km, or the collision will probably occur.

Further, by the size of any two planes' CPA volume, the initial positions, and flight velocity, we can get a set of sufficient judgement conditions to judge whether there are potential conflicts or not. At first, Let us judge the horizontal conflicts. The main mathematical deduction is given below. Secondly, consider the vertical potential conflicts. Through the same procedure, we also obtained a sufficient judgement conditions.

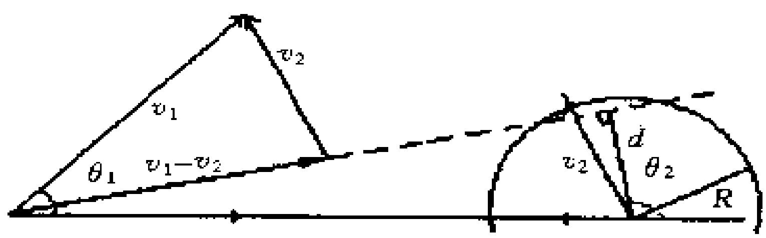


Figure 4

From the figure above, let $V = V_1 - V_2$. We can get a line equation by the position of plane 1 and the direction of relative velocity V. The minimum distance from the position of plane 2 to the line is d. If $d < R$ (here $R = l$) and $V \cdot P < 0$ ($V \cdot P$ is product of vector V and P), the two plane will have path conflicts. It is very easy to calculate by the knowledge of analytic geometry. When two planes are both in sector, and they meet the condition above, the controller would think that they need to intervention.

Strengths

Through the Double-cylinder model, we can accurately calculate the closest distant between any two planes and the dead time that a controller has to alarm the pilot to adjust the flight path to avoid the conflict. And the algorithm's runtime is very short, need only a few timeticks.

Weaknesses

In the Double-cylinder model, to simplify the model, we regard the plane's path as a line, ignoring the case that the path may be an arc. So there must be a little error, though it is allowed in real condition. And we do not consider the time lag (time separation from the controller sending message to the pilot adjusting the path).

Transient adjustment Model: Two Basic Models to Obtain the Optimal Adjustment Project at Any One Instant

Because we think a control system's task is not only to alarm planes to avoid conflicts, but also to provide a preventing collision, what's more to provide a method had best can obtain the maximum satisfaction, that is to say to made the summation of the adjusting angle to be the minimum. The basic thought is global searching. Of course to shorten the runtime, we design several algorithms, at the same time, we guarantee the solutions are approximately the optimal ones.

1. Assumptions

(1) Each plane has the same size of the maximum CPA volume, $r = 8$ km, $h = 300$ m. (The reason is also to improve the safety and to simplify the algorithm.)

(2) In one sector, there are no more than 8 planes simultaneously passing through that sector.

(3) For each plane, the maximum horizontal adjusting angles are the same, 10°.

(4) Two basic adjusting methods horizontal and vertical adjustment can not be made at the same time. When one method is made, the other one cannot be made.

2. Definitions

(1) f_i: the ith plane.

F_n: The total number of planes at any one instant.

F_u: The number of the upper level.

F_l: the number of the lower level.

(2) c_j: At the searching algorithm, the jth circle

(3) d_i: degree, which means the number of all the potential conflicts of plane f_i.

$D_j$max: the maximum degree of c_j.

(4) $d_j$max: the maximum d_i, of each searching circle, c_j.

(5) S_s: searching step, 0.1°.

(6) T_m: The global searching's maximum times

(7) θ_{max}: $\theta_{max} = +10°$ "+" means anti-clockwise, "−" means clockwise.

(8) θ_o: the optimal sum of adjusting angle, which means the accumulative total of adjusting angles which planes needed to be adjusted.

The Adjustment Methods to Avoid Horizontal Conflicts at Any One Instant

1. Horizontal adjustment steps

If only global searching method is used, our program's runtime will be too long to suit the fact. So we designed two plans to improve the global searching algorithm.

(1) Finding the Maximum Degree at Each Circle

At the beginning of each searching circle, c_j, find the f_j, whose degree is $d_{j\max}$, and to begin with f_j, search for the satisfactory adjustment angle. Finish this circle's searching, go on with the next circle. The next searching circles can be deduced by analogy, until the d_i is 0. The value of θ_o will be obtained. Though the θ_o may be not the optimal adjusting angle, according our comparing, it is close to the real optimal solution.

Adding the FMDEC algorithm can spare a lot of program's runtime. (see to the appendix)

(2) Dividing Sector to Local Searching

As each plane has two adjusting directions (anti-clockwise and clockwise), and considering the case that eight planes (the maximum number) simultaneously passing though the sector, we divide the whole adjusting range into 2^8, 256, sectors. On each condition we can obtain a local optimal solution. From the 256 local optimal solutions, choose a global optimal solution. Through this algorithm, the expectation searching times can reduce be to $(20/0.1)^{n-1}$.

Adding the two methods to the searching algorithm, our program can give a compromised optimal solution synthetically considering the performance of adjustment angle and the runtime. The specific data is shown in appendix. We guess FAA will prefer to adopting our Improved Searching (IS) algorithm.

2. Vertical adjusting model

We acquired the information from that a real sector is always separated into even number horizontal levers, and planes of the same heading fly in the same level. A plane in order to avoid conflict climbing or descending into its adjacent level is an adjusting method requiring less runtime and less complexity, which can be seen from appendix. In this model we only consider vertical adjustment between any two adjacent levels, and all conflicts can be avoided in this way.

Assume Planes in a level fly at the same altitude, if any other plane flies into this level, they will be at the same altitude too. Our

Vertical Adjusting Model does not only mean adjusting in vertical direction, but considering VA at first. If only VA cannot avoid all the conflicts, make HA method as described above.

1) The basic thought of VA algorithm

(1) Coordinate Mapping: Set up two 2D coordinates, upper and lower one. Mapping the upper coordinate to the lower one, with the origin at the same point. If the CPA of planes in upper level are overlap with the planes in lower one, they can't adjust into the same level, and these especial planes will be allocated into different level again. If the plane in the upper level which has been mapped in the lower one is allocated in the upper level too, we can think this plane has no adjustment in vertical direction. And then we adjust the planes in each level. But the planes which have been adjusted in vertical direction can't be adjust now. So our problems is degenerated into searching all allocated projects to avoid conflicts in a 2D plane, and making vertical adjusting from this level into the upper one.

(2) Balance of the Number: Try to keep the balance of the number of aircraft in the two levels, because in this case, the total amount of potential conflicts is probably small, and try to minimize the adjustment times.

2) About the result

If the distribution of planes have been changed after adjustment, it can be considered that we adjust these changed planes. For example, there are three planes in the upper level named a, b, c, and other five planes named d, e, f, g, h, which fly in the lower level. After the adjustment, there are b, c, e, f in the upper lever and others in the lower. Then the plane named a has been adjusted into lower level and planes e and f have been adjusted into upper one.

3) The basic methods

(1) Judge whether any couple of planes in each level have potential conflicts. If they have some, they need adjustment.

(2) If $d_i \neq 0$, we should make further judgement to see whether there will be the overlap of CPA volume. If there are no overlap of CPA volume among them, go along as followings(We will discuss the special situation mention in Coordinate Mapping after these steps):

<1> Select one: Move one of the eight planes into the upper planar. Use the IS (Improved Searching) algorithm(according to the rules in Assumption 4 of *Transient adjustment Model*) find the optimal resolution θ_{o1} (the minimum adjusting angle) in the lower level, $T_{m1} = C(8,1) * (20/0.1)^7$

($C(m, n)$ represent C_m^n.)

<2> Select two: Move two planes of the eight into upper planar. Calculate the adjustment time in each level. Use T_{m2} represent total time. $T_{m2} = C(8,2) * [(20/0.1)^6 + (20/0.1)^2]$.

<3> Select three: In the same way, $T_{m3} = C(8,3) * [(20/0.1)^{8-3} + (20/0.1)^3)$.

And all others can calculate in this way (T_{m4}, T_{m5}, T_{m6}, T_{m7}, T_{m8}.)

c. For each mapping, we can go as the above steps, at last we can get the optimized projection. Yet not in all cases the data should be used. To accord with the basic thought(see in **Balance of the** number of The basic thought of VA algorithm)

We draw a table to analyze all the cases:

The initial f_u	1	2	3	4
The initial f_t	7	6	5	4
T_m	$\sum(T_{m1}, T_{m4})$	$\sum(T_{m1}, T_{m4})$	$\sum(T_{m1}, T_{m4})$	$\sum(T_{m1}, T_{m4})$
The initial f_u	5	6	7	
The initial f_t	3	2	1	
T_m	$\sum(T_{m4}, T_{m5})$	$\sum(T_{m4}, T_{m6})$	$\sum(T_{m4}, T_{m7})$	

(If the f_t is 8 or 0, the maximum searching time is the same as the IS algorithm.)

Especially: If there are the overlap of CPA volume, we must put one of the two planes into the upper level. Suppose n pairs of planes' CPA volume are overlap. Because the two planes of any pair are impossible at the same lever, one of them must be at the upper level at first. Move one of the two planes into the upper level. Then there are n planes in the upper level. In accordance with thought 2, and by the described algorithm, we can obtain the optimal adjusting solutions.

Strengths of vertical adjustment

This adjustment method can reduce adjusting times comparing to the horizontal adjustment (see in table above), and does not need too much runtime. At the same time, if the controller adopts the adjustment project, the workload will be reduced

Weaknesses

In this algorithm, we assume that the vertical adjustment have no directly effect on the total adjustment of angles. In fact, the vertical adjustment is not so easy as we think. If given the weigh between vertical and horizontal adjustment, we can get more accurate project.

Dynamic Adjustment Model:

1. Model description

When no plane enters the sector, use Time-slice Model to avoid potential conflicts. We divide any given interval into several time-slices, and only give the adjustment advisory at the beginning of each time-slice. When some planes enter the sector, judge whether conflicts will occur or not. And if there will, we give two method to avoid conflicts. The first one is that only adjust flight path at the moment plane fly in. The second one is the same as Time-slice Model.

2. Assumptions

(1) It will cost a plane no more than 18 minutes to fly through this square airspace (160×160 km^2) at the speed of 800 km/h.

(2) During this given interval of time, we will not adjust the aircraft unremittingly. We divide the interval of time into several parts and only adjust it at the beginning of each part. In the follow model, if the given interval time is less than 6 minutes, we only adjust the aircraft at the beginning of each 3 minutes. Based on the fact that the time separation spent in the communication between the controller and the pilots is at most 2 minutes (suppose a communication to one pilot is 12 sec, the maximum communication time is $12 * 8$), we think choosing 3 minutes as a time-slice is suitable.

(3) The aircraft flying in this airspace will not allow to adjust more than three times, and in each adjustment the deviated angle will no more than $10°$. When the aircraft fly away from the space, it's track can't deviate $10°$ from the primary track..

3. Definitions

N: The amount of adjustment circle

According to Assumption (2), if there are F_n planes fly in this airspace, we'll get all possible combination aspects about the plane adjusting at the beginning of each interval time during the given time.

I: The adjustments in circle I

M: The given interval of time will be divided into M parts

J: The number of time parts in all M times adjustment

T: The given interval of time

T_j: The time when we make J adjustment

4. Adjust rules

1. Reduce the number of conflicts

2. Put off the happening time of conflicts

3. If at the end of the given interval time there still have potential conflicts, we'll adjust the planes with the Transient Adjust model at the end of the interval time.

5. Adjust steps

When no planes flying into the sector, steps of the Time-slice model are given as the flow process diagram below.

When there are planes flying into the sector, the adjustment steps are:

Step A: Make adjustment when a plane flying in.

At the beginning of the interval of time, we adjust aircraft with the Transient Adjusting algorithm. In the following time, if there is a plane flying in, we will judge if the aircraft have some conflicts with some others in the same level at first. If no conflicts, this aircraft need no more adjustment. If it has some conflicts in the horizontal direction, we will then judge if the aircraft has some other conflicts with the aircraft in the vertical direction. If it has some too, we only consider horizontal adjustment. If it has no vertical conflict, we consider two ways to solve this problem. The first one is to adjust planes in this level. Adjust the plane just flying in first, if the conflicts can not be avoid, then adjust others. (Because other planes have been adjusted and they have no conflicts if there is no other planes flying into.) And the second one is to adjust this plane to its adjacent level. Compare the two adjustment angles obtained from the two algorithms, and use the more optimal algorithm of the two. (see in appendix)

Step B: Also use time-slice model to avoid conflicts. The main steps are:

When a plane flies in, at the next time-slice, adjust the plane. If only adjust this one can not avoid all conflicts, adjust others, until no potential conflicts. In this condition, during a time-slice, if a plane flies in, it will be adjusted at the beginning of the next time-slice if it need be adjusted. And we assume during this time, no conflicts will occur, and all the potential conflict can be avoid after the time lag. And when another plane flies in, use the same way to avoid conflicts.

How to Detect and Avoid Potential Conflicts during Particular Time of Day?

We only consider two extreme cases. One is that there are few planes and few conflicts, the other is that the number of planes simultaneously passing the sector is large, including many planes climbing or descending, and many conflicts would occur. Through simulating the extreme cases, we find in case one, our program only need very little time to obtain result, and the result shows that the total adjustment angles are small and the total adjustment times are very short. In case two our program need a much longer runtime, the false alarm rate is high (because in this case, some flights are arc and the flight velocity does not keep the same, which does not suit our assumption). The long runtime and the high false alarm rate will increase the controller's workload. So we offer two improved methods. One is broaden searching step, the other is that during searching circle, we only need to find a feasible solution, and do not to find an optimal one. By this way, the runtime can be reduced to a large extent. But the false alarm rate does not decrease. To deal with this problem, a long interval time should be used to watch planes real flight path, and obtain the dynamic velocity changing with time. Judging by these data, the false alarm rate will be reduced.

Analysis of the Results

We implemented the model in the C programming language, making some additional assumptions. Considering the algorithm in TAM (Transient Adjustment Model) is the kernel of all specific models, we only list some results about it as following. These data were collected from the situation in which there are maximum potential conflicts. And we draw the figures shown in following.

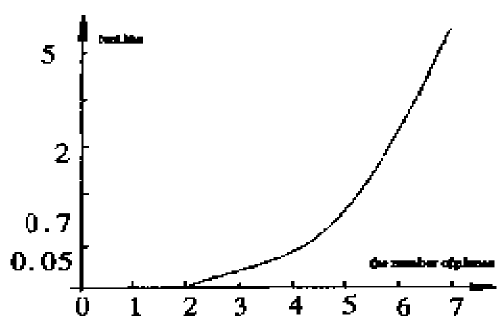


Figure 5: The relation between the number of planes and runtime.

From Figure 5, we can draw a conclusion that the more planes in an airspace sector, the more runtime needing in searching for op-

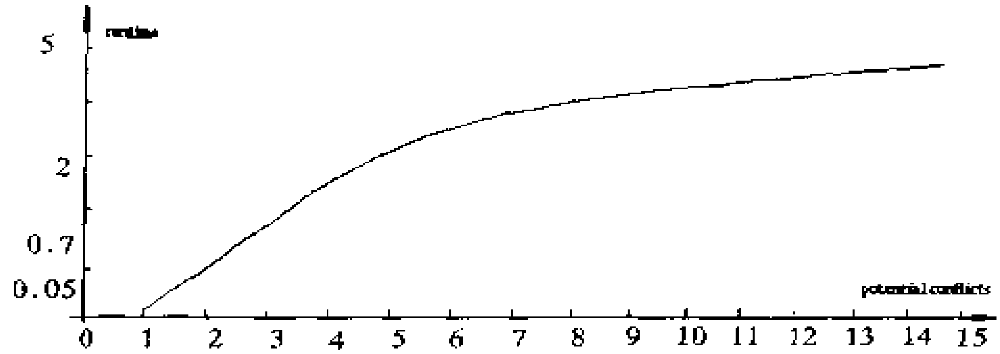


Figure 6: The relation between potential conflicts and runtime.

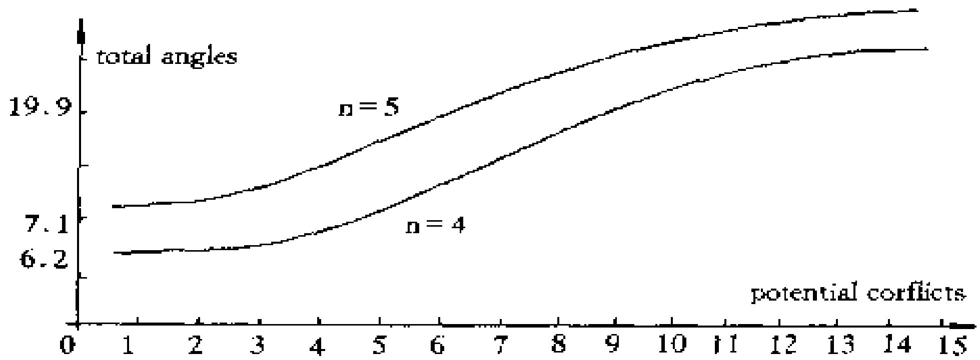


Figure 7: The relation between potential conflicts and total angles

timal project. In our model algorithms bases on searching for all feasible project and find out an optimal one. Because we have found restrict conditions in solving conflicts and we used some new optimized methods, such as searching the plane which have the maximum degree at first, partition each adjustment angle range into two (which can reduce at lease half of the runtime), and so on. This algorithm is primary through out all our models. As discussed in this paper, the runtime in **Optimal Adjustment Project at Any One Instant** is less than the time in this searching algorithm. And total runtime in **Dynamic Adjustment** is nearly less than half of this runtime. In **Particular Adjustment**, the runtime is much less.

Shown as Figure 6, we can deduce the relation between these two. In our algorithm, we adjust the plane which have the largest degree first. So the number of conflicts has directory affect on runtime. The tests proved our idea.

Shown in figure 7, we think the total adjustment angles have some relation with the number of planes and the conflicts.

We can draw a conclusion from the results about the relation between **complexity** and the number of planes and conflicts. As discussing above, we think the **complexity** can be evaluated by the total adjustment angles and runtime . So we can say, the number of planes and the conflicts are main factors affecting the workload and the complexity. With the help of software, the air traffic controller can easily know when to send alarm message and adjustment projects to the planes. With efficient software, he can reduce his workload and improve the safety of air flight.

We select some complex situation from the simulating. The number of planes ranges from 2 to 6. There lists the results in the Horizontal Adjustment in Transient Adjustment model as following.

X(m)	Y(m)	V(m/s)	Angles(°)	Adjustment angles
−75 000	0	223	0	−0.4
−23 176	71 329	222.9	−72.00	0
71 000	23 716	211.9	−160.71	6.5
71 000	237.0	211.9	160.71	−6.1
−23 716	−73 000	217.6	43.58	0

X(m)	Y(m)	V(m/s)	Angles(°)	Adjustment angles
Number of planes: 5 Potential conflicts: 6 Time: 32 tick Adjusting angles: 13°				
−75 000	0	223	0	−2.3
−37 500	64 951	223	−60	−0.9
223	−120	223	−120	9.4
37 500	64 951	223	−180	−6.2
−37 500	−64 951	223	120	−6.1
−37 500	−64 951	223	6.0	−6.2
Number of planes: 6 Potential conflicts: 15 Time: 183 tick Adjusting angles: 31.1°				
6 000	6 000	223	−120	1.3
6 000	−265	200	75	0
−20 000	50 000	210	120	0
−20 000	−50 000	230	30	0.1
9 000	9 000	200	90	0
−75 000	0	223	0	2.6
75 000	−75 000	240	135	0
Number of planes: 7 Potential conflicts: 2 Time: 61 tick Adjusting angles: 4°				
60 000	60 000	223	−135	0
−30 000	50 000	223	−70	8.2
−63 000	20 000	230	−65	0
−20 000	−75 000	235	80	0
32 000	−40 000	210	140	0
45 000	−75 000	205	30	1.978
Number of planes: 6 Potential conflicts: 1 Time: 36 tick Adjusting angles: 1.70°				
2 000	60 000	223	−135	8.8
−30 000	20 000	223	−70	0

X(m)	Y(m)	V(m/s)	Angles(°)	Adjustment angles
−63 000	20 000	230	−65	0
−2 000	−75 000	235	80	0
3 200	−40 000	210	140	0
Number of planes:5 Potential conflicts:1 Time:13 tick Adjusting angles:8.8°				
−50 000	50 000	223	−45	0
50 000	50 000	223	−120	0
50 000	−40 000	223	135	−0.8
−55 000	−50 000	223	70	0
Number of planes:4 Potential conflicts:1 Time:1 tick Adjusting angles:0.8°				
−75 000	0	223	0	−5.3
0	75 000	223	−90	−5.9
75 000	0	223	180	−0.9
0	−75 000	223	90	−7.8
Number of planes:4 Potential conflicts:6 Time:4 tick Adjusting angles:19.9°				
−75 000	0	223	0	−6.2
75 000	0	223	180	0
Number of planes:2 Potential conflicts:1 Time:0 tick Adjusting angles:6.2°				
−75 000	0	223	0	−5.3
0	75 000	223	−90	−0.9
75 000	0	223	180	−0.9
Number of planes:3 Potential conflicts:3 Time:1 tick Adjusting angles:7.1°				

Strengths

In our models, we set up algorithms respectively according almost every air traffic condition. And in each condition, we can give a proximately optimal solution through our algorithms. At the same time, using several improved algorithms, we guarantee the result of a more complex condition can be output within 5sec. Our algorithms have the high reliability. Through some random data, simulating our algorithms can give feasible solutions. And the thought in our algorithm is clear and easy to implement.

Weaknesses

Because we have to guarantee the runtime in a low range, we use some simply algorithm to search optimal solution, the solutions we get may be not the exact optimal ones. We do not make further analysis in Vertical Adjusting project. Among our assumptions, we consider the velocity of one plane does not change during flying through the airspace sector, and replace the real track by a line ignoring circle flight path. So the result may be not very precise and our program will generate false alarm in some cases.

Appendix: Flow Diagram of Transient Adjustment Model

To Jane Garvey

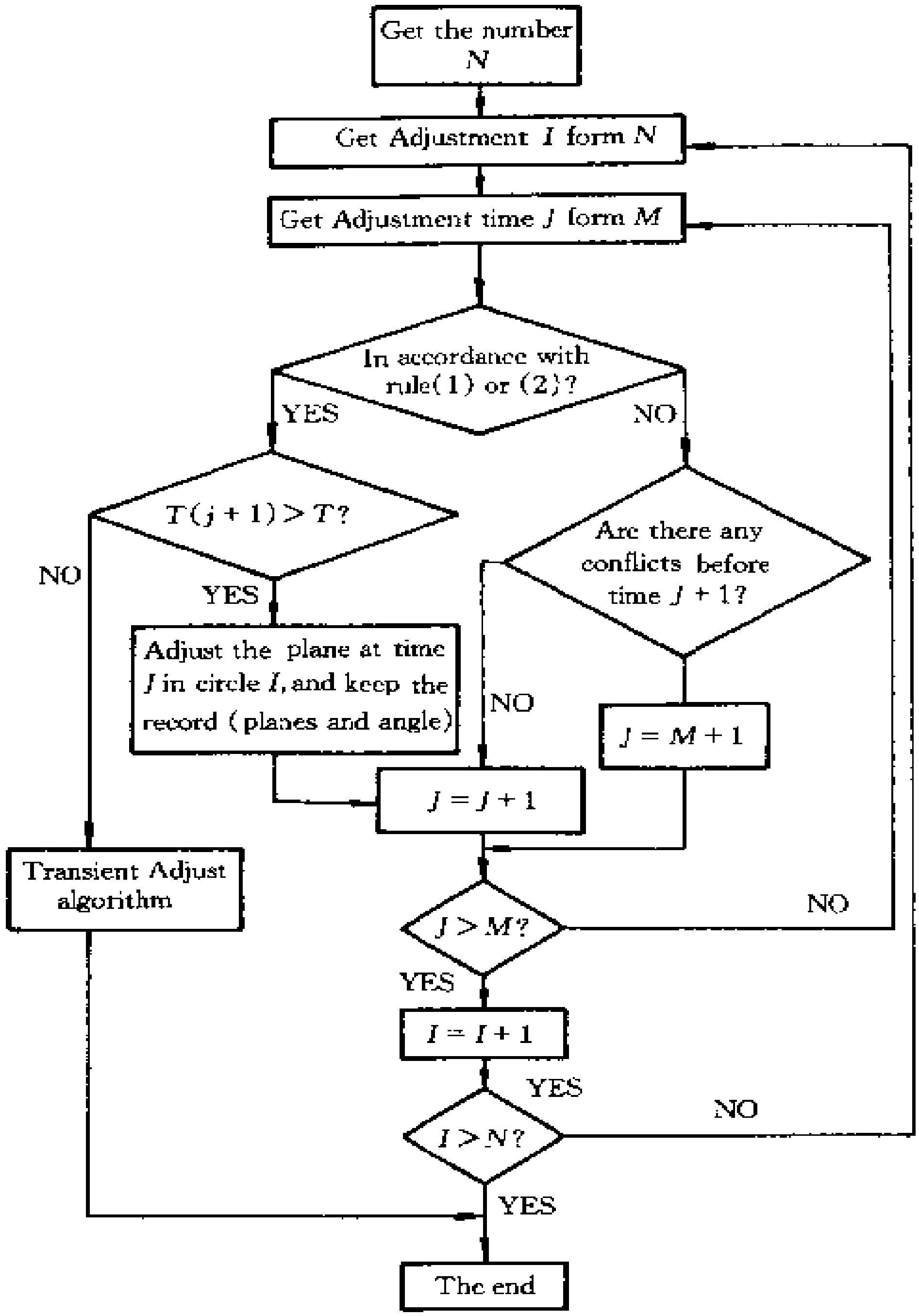
Under the background that FAA brought out "Free Flight", We have studied in depth a set of problems including how to detect potential path conflicts, how to avoid conflicts in a small adjusting range, and how to measure complexity from an air traffic workload perspective.

We set up several models and algorithms. Based on our Double-Cylinder model, we gave an algorithm and obtained accurate expressions of safety distance between any two planes flying at different velocities and in different directions. And the algorithm is very easy to calculate. At the same time we can guarantee planes' safety as long as planes fly beyond the minimum safety distance.

We not only gave algorithm to detect potential conflicts but also generated adjustment project to avoid conflicts. We analyzed all kinds of cases, and according to each case, we offered a set of feasible solutions.

First by combining horizontal and vertical adjustment, we can get the optimal coordination project. Once the state of planes including positions and velocities are given, our program can calculate the adjusting method quickly. Considering a most complex condition, seven planes simultaneous passing through the sector in the same level, within 3 — 5 sec our program can get a set of feasible solutions. And in other cases, the runtime is very short. Seen from the above example, adding our algorithm to the air traffic control system, workload can be reduced to a large extent.

Also, according to our model, we can analyze the factors affecting complexity including the number of planes flying through the sector simultaneously, the number of potential conflicts and other factors. Based on several sets of data, we deduced the approximate relation between all factors and complexity. At any one instant the total number of the planes in the sector is the main factor affecting complexity, and during any given interval time, the trend of the effect is the same. Seen from our result, suppose in a busy airspace, six planes fly through a sector at the same level, with opposite directions. Our algorithm can give a adjusting method within a very short time. And it will help the controller to draw conclusion and send advice to pilots, and alleviate the air traffic workload. During particular time of day, especially in the case that planes frequently climbing and descending, the air traffic workload is much heavier than common cases, controllers are all in a state of tension. When there is



more planes, they may not have the right judge. So our algorithm can alert them at proper time, help him to judge correctly and get a rational solution.

Our algorithm is efficient and easily to implement. But there are some weaknesses in our algorithm. Because we are not so familiar with the real situation, we may omit some factors affecting the air traffic workload. It may lead some effect. Although the effect is not important, this is future work we should do.

We suggest FAA add software to air traffic control system. Based on our analysis, general speaking, it will reduce the workload. We also expect FAA feed back some information to help us strengthen our algorithm to be more feasible.

论 文 点 评

本论文首先给出了影响复杂性的若干因素，如空间的飞机数目，存在潜在冲突的飞机数目，两架存在潜在冲突的飞机方向夹角等，并给出了度量这种复杂性的具体方法；接着给出了自己的模型，通过一个近似有效的算法可以估计出两架飞机距离是否太近而需要干涉，并根据一些典型的数据得到了具体的警戒距离。论文的后半部分，着手在三个给定的情形下研究问题的复杂性，对每一种情形，分别采用了不同的算法估计出潜在的飞行线路冲突并及时提醒管理人员进行有效的调整。文中对所得的结果给出了进一步的分析，并给出了具体的分析数据，验证了算法的可实现性及快捷性，较好地完成了问题中所要考虑的情况；不足之处是对可能的误差分析做的不够。

该论文获得美国 2000 年 MCM 竞赛一等奖。

Our algorithm is efficient and easily to implement. But there are some weaknesses in our algorithm. Because we are not so familiar with the real situation, we may omit some factors affecting the air traffic workload. It may lead some effect. Although the effect is not important, this is future work we should do.

We suggest FAA add software to air traffic control system. Based on our analysis, general speaking, it will reduce the workload. We also expect FAA feed back some information to help us strengthen our algorithm to be more feasible.

论 文 点 评

本论文首先给出了影响复杂性的若干因素,如空间的飞机数目,存在潜在冲突的飞机数目,两架存在潜在冲突的飞机方向夹角等,并给出了度量这种复杂性的具体方法;接着给出了自己的模型,通过一个近似有效的算法可以估计出两架飞机距离是否太近而需要干涉,并根据一些典型的数据得到了具体的警戒距离。论文的后半部分,着手在三个给定的情形下研究问题的复杂性,对每一种情形,分别采用了不同的算法估计出潜在的飞行线路冲突并及时提醒管理人员进行有效的调整。文中对所得的结果给出了进一步的分析,并给出了具体的分析数据,验证了算法的可实现性及快捷性,较好地完成了问题中所要考虑的情况;不足之处是对可能的误差分析做的不够。

该论文获得美国 2000 年 MCM 竞赛一等奖。

Our algorithm is efficient and easily to implement. But there are some weaknesses in our algorithm. Because we are not so familiar with the real situation, we may omit some factors affecting the air traffic workload. It may lead some effect. Although the effect is not important, this is future work we should do.

We suggest FAA add software to air traffic control system. Based on our analysis, general speaking, it will reduce the workload. We also expect FAA feed back some information to help us strengthen our algorithm to be more feasible.

论 文 点 评

本论文首先给出了影响复杂性的若干因素,如空间的飞机数目,存在潜在冲突的飞机数目,两架存在潜在冲突的飞机方向夹角等,并给出了度量这种复杂性的具体方法;接着给出了自己的模型,通过一个近似有效的算法可以估计出两架飞机距离是否太近而需要干涉,并根据一些典型的数据得到了具体的警戒距离。论文的后半部分,着手在三个给定的情形下研究问题的复杂性,对每一种情形,分别采用了不同的算法估计出潜在的飞行线路冲突并及时提醒管理人员进行有效的调整。文中对所得的结果给出了进一步的分析,并给出了具体的分析数据,验证了算法的可实现性及快捷性,较好地完成了问题中所要考虑的情况;不足之处是对可能的误差分析做的不够。

该论文获得美国 2000 年 MCM 竞赛一等奖。

Our algorithm is efficient and easily to implement. But there are some weaknesses in our algorithm. Because we are not so familiar with the real situation, we may omit some factors affecting the air traffic workload. It may lead some effect. Although the effect is not important, this is future work we should do.

We suggest FAA add software to air traffic control system. Based on our analysis, general speaking, it will reduce the workload. We also expect FAA feed back some information to help us strengthen our algorithm to be more feasible.

论 文 点 评

本论文首先给出了影响复杂性的若干因素，如空间的飞机数目，存在潜在冲突的飞机数目，两架存在潜在冲突的飞机方向夹角等，并给出了度量这种复杂性的具体方法；接着给出了自己的模型，通过一个近似有效的算法可以估计出两架飞机距离是否太近而需要干涉，并根据一些典型的数据得到了具体的警戒距离。论文的后半部分，着手在三个给定的情形下研究问题的复杂性，对每一种情形，分别采用了不同的算法估计出潜在的飞行线路冲突并及时提醒管理人员进行有效的调整。文中对所得的结果给出了进一步的分析，并给出了具体的分析数据，验证了算法的可实现性及快捷性，较好地完成了问题中所要考虑的情况；不足之处是对可能的误差分析做的不够。

该论文获得美国 2000 年 MCM 竞赛一等奖。