Aircraft Conflict Resolution by Genetic Algorithm and B-Spline Approximation
Daniel Delahaye, Clément Peyronne, Marcel Mongeau, Stéphane Puechmorel

To cite this version:
Daniel Delahaye, Clément Peyronne, Marcel Mongeau, Stéphane Puechmorel. Aircraft Conflict Resolution by Genetic Algorithm and B-Spline Approximation. EIWAC 2010, 2nd ENRI International Workshop on ATM/CNS, Nov 2010, Tokyo, Japan. pp 71-78, 2010. <hal-00956473>
Abstract  Conflict resolution has always been a sensitive matter in air-traffic management. Current European projects aim partial or total automation of air traffic control to deal with the constant growth of air traffic. Technological advances on flight management system allows us to consider an automatic conflict resolution using continuous trajectories. In this paper, we present a new methodology that, first, relies on B-splines to model trajectories, secondly models air-traffic conflict resolution as an optimization problem whose decision variables are the spline control points. Finally, we use genetic algorithms to tackle this optimization problem in order to generate optimal conflict-free situations.

Keywords  Air traffic conflict resolution, genetic algorithm, B-Spline approximation

1 INTRODUCTION

From the beginning, the most critical point of air traffic control was to ensure safety separation between airplanes. To achieve this goal a safety standard separation distance has been defined: 5 Nm (Nautical miles) horizontally and 1000 feet vertically. Air traffic controllers are responsible for ensuring the respect of these separation rules.

1.1 Context

Air traffic being constantly increasing, controllers in charge of an aviation sector must handle more and more flights. Up to now, decreasing the size of aviation sectors could compensate the growth of traffic but we are reaching the point where a decrease of the size of sectors is no longer efficient.

Nowadays, air traffic management has already used every available ressource to increase airspace capacity. However, from now to 2020, air traffic is expected to be multiplied by two or three. Consequently, air traffic management (ATM) will have to deal with this overload while ensuring at least equivalent standards of safety [1]. The SESAR European project aims at finding solutions to this problem by providing a decision support to air traffic controllers in order to decrease their workload. Considering the technological advances on the aircraft flight management system (FMS), we will explore in this paper the possibility of a full automation generating continuous trajectories that new FMS can follow.

1.2 Previous Related Works

Eversince 1981 and the famous air traffic US controller strike (where more then 6000 controllers were fired by Ronald Reagan), numerous projects of air traffic control (ATC) automation have been led. However, no definitive results have been found. Two kinds of method have become predominant for their good results : genetic algorithms (GA) [3] and navigation functions [8]. By the past, two kinds of method have
In this paper, we present a new methodology that, first, relies on B-splines to model trajectories, secondly models air-traffic conflict resolution as an optimization problem whose decision variables are the spline control points. Finally, we use genetic algorithms to tackle this optimization problem in order to generate optimal conflict-free situations.

We present our trajectory model in Section 2. Section 3 presents the optimization method we chose to solve our optimization problem: genetic algorithms (GA). We detail in Section 4 the objective function (conflict detection) of our optimization problem. Numerical results are presented in Section 5. We draw conclusions in Section 6.

2 TRAJECTORY MODEL

In this section, we will describe our B-spline trajectory model. We choose a smooth trajectory model fitted to future FMS abilities. In order to obtain FMS flyable trajectories, we use B-splines, the mathematical tool which provides the most interesting properties for our concerns. We shall first present the B-spline concept and then we shall describe how we use it in our model.

B-spline was primarily introduced to find a curve interpolating a set of points of \( \mathbb{R}^2 \) called control points. It was later extended to approximation, thereby avoiding the undesirable oscillation inherent to interpolation. In our study, we shall focus on this use of splines to approximate a set of control points. The control polygon, the piecewise linear curve linking the control points, completely defines the curve [7], [5]. The B-spline curve always remains in the convex envelope of the control points (see figure 2).

Basically, B-splines are parameterized curves generalizing the Bezier curve concept. It is an efficient approximation tool which is constructed from polynomial pieces joined at certain parameter’s values called knots, stored in a knot vector. In a very simplified way, if we consider a set of control points \((X_i, Y_i) = P_i \in \mathbb{R}^2 (i = 0, 1, ..., n)\), and a parameter \(u\), we can define the B-spline as follows:

\[
C(u) = (\sigma_x(u), \sigma_y(u)), u \in [a, b]
\]

Figure 1 Principle of navigation functions between two aircrafts (left), between an aircraft and an obstacle (right)

Figure 2 Control points, its convex envelope, control polygon and resulting B-spline

D. Delahaye, C. Peyronne, M. Mongeau, S. Puechmorel

...all computations...
where \( \sigma_i(u) \) and \( \sigma'_i(u) \) are the B-splines approximations of the couples \((i, X_i)_{i=0}^{m}\) and \((i, Y_i)_{i=0}^{m}\) for \( u \in [a, b] \). We choose to rely on B-splines modeling trajectories because it is a very efficient tool for curve approximation in terms of both approximation quality and computational time. Moreover, B-splines feature interesting properties such as \( C^2 \)-continuity (crucial for modeling smooth aircraft trajectories), robustness and flexibility (if one control point is displaced, only a small part of the curve will be affected). The last good point in favor of B-splines is its compatibility with GA: the curve is completely determined by few control points only, which will be the optimization parameters of the GA. In our study, we use a fixed maximal number of control points between the departure and the arrival points.

3 OPTIMIZATION METHOD

When several aircraft are involved in a conflict, the conflict resolution problem has been shown to be NP-hard [3]. Moreover, the optimization variables being the B-splines control-point location, we shall see that our objective function (4) is not differentiable with respect to these variables. Consequently, we must rely on black box (direct) optimization methods to address our problem. In this paper, we choose to use a stochastic global optimization method, genetic algorithms, to guide the control-point location. As we mentioned in the previous part, GA seem to be appropriate with B-splines. In this section, we explain briefly GA theory and then we shall detail how we adapt the different operators (selection, mutation and crossover) to our air traffic conflict resolution problem.

3.1 Basis of Genetic Algorithms

Evolutionary algorithms use techniques inspired by evolutionary biology such as inheritance, mutation, natural selection, and recombination (or crossover) to find approximate solutions to optimization problems [4], [6]. An individual, or solution to the problem to be solved, is represented by a list of parameters, called chromosome or genome. Initially several such individuals are randomly generated to form the first initial population (POP(k) with \( k = 0 \) on Figure 4).

Then each individual is evaluated, and a value of fitness is returned by a fitness function (generally, the objective function of our optimization problem). This initial population undergoes a selection process which identifies the most adapted individuals. The selection process which has been used in our experiments is a deterministic \((\lambda, \mu)\)-tournament selection. This selection begins by randomly selecting \( \lambda \) individuals from the current population POP(k) and keeps the \( \mu \) bests \((\lambda > \mu)\). These two steps are repeated until a new intermediate population (POP) is completed. Following selection, one of the three following operators is applied: nothing, crossover, and mutation. The associated probability of application are respectively \((1 - p_c - p_m)\), \(p_c\) and \(p_m\). Crossover results in two

Figure 3 In bold, we have the B-splines functions used to calculate the B-spline for \( u \in [1, 2] \) (the only ones with nonzeros values on this interval) Control points, its convex envelope, control polygon and resulting B-spline

\[
\sigma'(u) = \sum_{i=0}^{m} X_i B_i(u)
\]

\[\sigma''(u) = \frac{(x_{i+1} + 4x_i + x_{i-1})}{b^2} \quad \sigma''(u) = \frac{(2x_{i+1} - 3x_i + 2x_{i-1})}{b^2}
\]

We have analogous formulae for \( \sigma''(u) \). We choose to rely on B-spline modeling trajectories because it is a very efficient tool for curve approximation in terms of both approximation quality and computational time. Moreover, B-splines feature interesting properties such as \( C^2 \)-continuity (crucial for modeling smooth aircraft trajectories), robustness and flexibility (if one control point is displaced, only a small part of the curve will be affected). The last good point in favor of B-splines is its compatibility with GA: the curve is completely determined by few control points only, which will be the optimization parameters of the GA. In our study, we use a fixed maximal number of control points between the departure and the arrival points.
3.2 Chromosome Encoding and Fitness Calculus

In this section we detail the link between our modeling method and genetic algorithms. Indeed, our optimization method is using B-splines to calculate the aircraft trajectories in order to determine each individual's fitness. To define this interaction, we first define an encoding.

A chromosome will represent the trajectories of N aircrafts using a matrix of $N \times N_c$ where $N_c$ is the maximal number of control points we wish to consider. For example, the case where $N = 3$ airplanes and $N_c = 4$ we encode the trajectories as shown on figure 5.

Obviously, departure and arrival points are not encoded in the chromosome as they are not meant to move, they are not optimization variable.

Let us now define precisely how we specify the control points $P_j^i$. We want a compromise between allowing the trajectory to deviate freely (in any directions) from the business trajectory (BT) and staying as close as possible from it. For that purpose, we define a fixed width band ($D_{max}$), depending on the BT length ($D_{tot}$), around the BT trajectory where the control points are allowed to lie.

A matter for the control point location is that, if there are all gathered in the same region, it can create oscillations on the resultant trajectory. Consequently, we distribute uniformly the control points along the BT.

To summarize, a control point can be defined by a single parameter representing a bandwidth percentage ($\pm \frac{D_{max}}{D_{tot}}$ %).

Moreover, it is desirable to be able to have a different number of control points for a given chromosome. In order to satisfy this goal, we used a convention: percentage are generated within $[-200\%; 200\%]$, if the percentage is between $-100\%$ and $100\%$, then we create a control point, if it belongs to $[-200\%; -100\%] \cup [100\%; 200\%]$, we do not. Consequently, our chromosome is a $N \times N_c$ matrix of percentage. Several cases are presented in Figure 5, 6 and 7.

3.3 Crossover and Mutation

Let us now describe how we adapted the genetic operators we use in our GA.

![Figure 5](image_url)

Figure 5 Chromosome coding. In such example 3 aircraft are involved in a conflict and for each of them the coding contains the list of the control points which define the associated trajectory (here 4 control points).

![Figure 6](image_url)

Figure 6 Chromosome encoding and corresponding B-spline with one percentage exceeding 100%.

![Figure 4](image_url)

Figure 4 Genetic algorithm with tournament selection. The first step consists in the selection of the best individuals from population $POP(k)$. Afterwards, recombination operators are applied in order to produce the $POP(k+1)$ population.
3.3.1 Slicing crossover

The crossover operator aims at finding better solutions by mixing two good individuals of the previous generation. Therefore, we decide to keep trajectory consistencies by inverting only complete trajectories using a slicing crossover (we do not separate control points belonging to the same trajectory). Accordingly, the crossover consists in picking a plane number randomly and inverting the parent trajectories corresponding to the next aircrafts. An example is given in Figure 8:

3.3.2 Mutation

The mutation operator aims at diversifying the genes in the population in order to explore as much as possible the problem space. Thus, mutation consists in choosing randomly one control point in the chromosome and to assign to it a new random number (using a uniform distribution) in the interval [−200%; 200%]. (see Figure 9).

4 OBJECTIVE FUNCTION: CONFLICT DETECTION METHOD

In order to evaluate each chromosome fitness, we decode it into $N$ trajectory curves (one per aircraft) and evaluate two quantities. First, how many conflicts the situation engenders and secondly what is the total extra distance induced with respect to the BT. To calculate these quantities, we discretize the airspace into square cells of size half the standard separation (noted $d$). Our conflict detection is performed in two steps:

- First, for each airplane, we store the grid’s cells through which the aircraft flies, the aircraft number (its label), the entry and exit times in and out each of the stored cell,
- Then, we select each stored cell and we check whether any other airplane goes through any of the eight neighboring cells for other airplanes. If there are such airplanes, we check the time to see whether there is a conflict between these two airplanes. If so, we calculate the conflict duration.

An example of a conflict detection with two airplanes is given in Figure 10. The circle represents the approximate conflict zone, the bold cells are the cells where spatial conflict is detected and where time conflict must be checked. $d = \frac{ sep_{standard}}{2}$

Our conflict detection procedure send back the chromosome fitness to the GA. Here is the formula we use to calculate the fitness:

$$f(X) = -(CN + \sum_{i=1}^{N} \frac{NR_i}{BT_i} - N)$$
Figure 11 The 16-aircraft roundabout configuration

Figure 12 The roundabout configuration after resolution

Figure 13 The fitness evolution with respect the generation

where \( CN \) is the number of conflict, \( NR_i \) the length of the new route calculated by the algorithm for the aircraft \( i \), while \( BT_i \) the length of the business trajectory (straight line from the departure to the arrival point) and \( N \) the number of aircraft. High fitness corresponds to good individuals. Indeed, the lower are the number of conflict and the route lengthening, the better is the chromosome.

5 RESULTS AND PERSPECTIVES

In this section, we present several results we obtained using our methodology. First, we test our method on the roundabout test problem which is a very common test on automatic conflict-detection method. It consists in making a certain number of planes fly to the diametrically opposed point at a common speed (each point on the circle has an outgoing and an incoming trajectory).

A diagram of the situation is shown in Figure 11 (each point on the circle has an outgoing and incoming trajectory):

We choose to fly \( N = 16 \) aircrafts for results’ readability) equidistributed on a circle of 100Nm(= 185200m) radius. To address this conflict resolution, we use the following parameters:

- Number of generation : 100
- Size of the population : 500
- Mutation probability : \( p_m = 0.3 \)
- Crossover probability : \( p_c = 0.6 \)
- Maximal number of control points : \( N_c = 2 \)

To compute the conflict resolution, we used a 2.53GHz Intel(R) Core(TM)2 Duo on a Windows Vista Operating System and we coded in Java. The obtained conflict-free situation is presented in meter-scale (1Nm = 1852m) in Figure 12. This whole resolution computation required approximately 4 minutes.

The fitness’ evolution with respect to generations is shown in Figure 13. The fitness is meant to increase to 0 as formula (4) shows. Moreover, when the fitness is in \([-1, 0]\) the situation is conflict-free. Consequently, the algorithm stops only when the best individual fitness is in \([-1, 0]\).

Although one can easily solve intuitively this academic problem due to its symmetry, our automatic implementation does not exploit any symmetry here. This result shows that our methodology is promising as the obtained conflict resolution is consistent with experts’ experience.

We also test our methodology on a more realistic operational problem where \( N = 15 \) aircrafts are involved in a fuzzy convergence. In this situation, for each aircraft, we have its departure point, its heading and its speed. To calculate the arrival points, we make the 15 aircrafts fly for one hour at a constant speed, on a straight line following the initial direction (heading). We present the conflict-free situations our algorithm obtains is displayed in Figure 14.

We can see on Figure 15 that on an operational application, our method converges far more rapidly in
three generations (90 generations for the roundabout test problem). This results shows that our method can be very efficient from an operational point of view. Indeed, in operational problems, there are never more than four aircrafts involved in the same conflict. Consequently, our algorithm will be able to deal very efficiently with operational air traffic. For this situation, we used a different number of generations because of its extremely quick convergence (number of generation: 20). Moreover, we can see that few aircrafts are deviated, which is consistent with a controller behaviour, who would prefer to deviate largely one aircraft instead of deviating lightly several aircrafts to solve conflicts.

6 CONCLUSION

We have shown in this paper that the combination of B-splines and genetic algorithms can be a promising methodology for automatic conflict resolution in air traffic control. However, we have in mind several developments to improve our approach such as using sharing in GA (deals with equirepartition of the population on the different maximums) in our GA or implementing a self-adaptive GA (every parameter such as bandwidth, control points’ number, etc will be considered within the chromosome encoding, as proper optimization variables). Furthermore, we plan to exploit our B-spline model of trajectory to address the conflict resolution problem with deterministic derivative-free optimization methods [2]. Indeed, despite the local aspect of these methods, they can also be adapted to global optimization.

References


Acknowledgements

This work has been supported by French National Research Agency (ANR) through COSINUS program (project ID4CS ANR-09-COSI-005) and Capgemini through the first author PhD funding.

Copyright

The authors confirm that they, and/or their company or institution, hold copyright of all original material included in their paper. They also confirm they have obtained permission, from the copyright holder of any third party material included in their paper, to publish it as part of their paper. The authors grant full permission for the publication and distribution of their paper as part of the EIWAC 2010 proceedings or as individual off-prints from the proceedings.