Optimizing B-splines using Genetic Algorithms Applied to Air-traffic Conflict Resolution

Clément Peyronne, Daniel Delahaye, Marcel Mongeau, Laurent Lapasset

OPTIMIZING B-SPLINES USING GENETIC ALGORITHMS APPLIED TO AIR-TRAFFIC CONFLICT RESOLUTION

Clement Peyronne, Daniel Delahaye
Applied Mathematic Laboratory (LMA), ENAC, 7 rue Edouard Belin, Toulouse, France
clement.peyronne@capgemini.com, delahaye@recherche.enac.fr

Marcel Mongeau
Universite de Toulouse ; UPS, INSA, UT1, UTM ; Institut de Mathematiques de Toulouse ; F-31062 Toulouse, France
CNRS; Institut de Mathematiques de Toulouse UMR 5219; F-31062 Toulouse, France
marcel.mongeau@math.univ-toulouse.fr

Laurent Lapasset
Air Traffic Management, Capgemini, Toulouse, France;
llapasse@capgemini.com

Keywords: Air-traffic conflict resolution, B-splines, Genetic algorithms

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.

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 (Alliot and de Verdière, 2003). 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 Work

By the past, two kinds of method have become predominant in automatic air traffic conflict resolution for their good results : genetic algorithms (GA) (Durand, 2004) and navigation functions (Zeghal, 1994).

Genetic algorithms are part of what is known as natural computation or evolutionary methods. This optimization method is based on the evolution theory and uses concepts such as mutation, crossover and, of course, selection. Each possible solution of our optimization problem is encoded as a chromosome via a specific encoding (for example, number and duration of simple maneuvers). The algorithm creates randomly a first chromosome population. Each chromosome ability to solve the problem (*fitness*) is then evaluated. The best individuals (according to

their fitness) are selected and mutation, crossover are applied to obtain a new population of chromosome (next *generation*). The user chooses the necessary generation number for the algorithm to converge (this choice is usually done empirically).
The obtained trajectories' velocity stays within acceptable bounded range with respect to ATM operational constraints. However, these techniques have not been tested yet with curved trajectories.

Navigation functions use a different representation. Indeed, the airspace is considered as a potential field, and airplanes like particles navigating in it. Naturally, negative charges represent obstacles to the airplane (other aircrafts, congested areas) and positive charges, its destination. As a result, each airplane is attracted to its destination while being repulsed by obstacles which enables to the automatic generation of conflict-free trajectories.

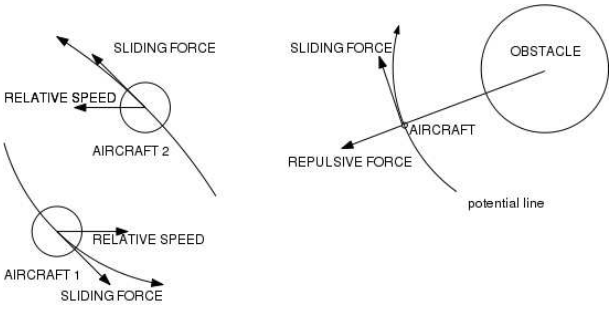

Figure 1: Principle of navigation functions between two aircrafts (left), between an aircraft and an obstacle (right)

It has been demonstrated that navigation function methods ensure collision avoidance and connection between departure and destination. The major drawback of this method is that the obtained solution does not respect the ATM constraints such as limited speed, or trajectory smoothness. Evenmore, they can lead to major delays and overcosts as they tolerate any deviation from the *business trajectories* (BT : the straight line between the departure and arrival points).

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

B-spline was primarily introduced to find a curve interpolating a set of points of 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 (Rabut, 2008).

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 $(X_i,Y_i) = P_i \in R^2 (i = 0,1,...,n)$, and a parameter u, we can define the B-spline as follows :

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

where $\sigma_x(u)$ and $\sigma_y(u)$ are the B-splines approximations of the couples $(i,X_i)_{i=0:n}$ and $(i,Y_i)_{i=0:n}$ for $u \in [a,b]$.

Theoritically, the curve is calculated by multiplying the B-splines basis functions by the control points coordinates. For σ_x:

$$\sigma_x(u) = \sum_{i=-1}^{n+1} X_i B_i(u) \tag{1}$$

However, in practice, B_i construction is made by recurrence. Consequently, we used a more numerically convenient way to calculate the B-spline values only using control points values. The B-spline and its derivative values at knot u_i are defined by :

$$\begin{aligned} \sigma_i &= \tfrac{X_{i+1}+4X_i+X_{i-1}}{6} \\ \sigma_i' &= \tfrac{X_{i+1}-X_{i-1}}{2h} \\ \sigma_i'' &= \tfrac{X_{i+1}-2X_i+X_{i-1}}{h^2} \\ \sigma_i''' &= \tfrac{X_{i+2}-3X_{i+1}+3X_i-X_{i-1}}{h^3} \end{aligned} \tag{2}$$

Therefore, by using Taylor series, we obtain the B-spline value for all $u \in [u_i, u_{i+1}]$:

$$\sigma_x(u) = \sigma_{x_i} + (u-u_i)(\sigma'_{x_i} + (u-u_i)(\frac{\sigma''_{x_i}}{2} + (u-u_i)\frac{\sigma'''_{x_i}}{6})) \quad (3)$$

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 $\mathcal{C}^2$-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 (Durand, 2004). 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 (Goldberg, 1989), (Koza, 1992). 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 2). 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 (λ,μ)-tournament selection. This selection begins by randomly selecting λ individuals from the current population POP(k) and keeps the μ bests $(\lambda > \mu)$. These two steps are repeated until a new intermediate population (POP_i) 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 new child chromosomes, which are added to the next generation population. The chromosomes of the parents are mixed during crossover. These processes ultimately result in the next generation population of chromosomes (POP(k+1) on Figure 2) that is different from the initial generation. This generational process is repeated until a termination condition has been reached. The next section presents the application of this algorithm to our specific problem.

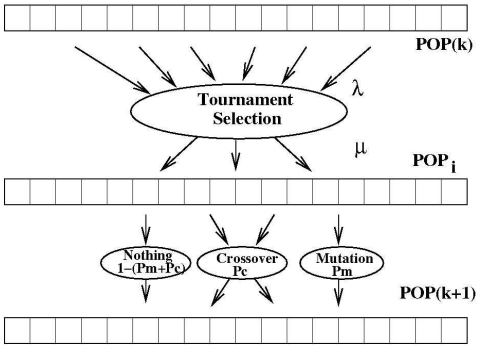

Figure 2: Genetic algorithm with tournament selection. The first step consists in the selection of the best individuals from population $POP(k)$. Afterwards, recombination operators are applied in order to produce the $POP(k+1)$ population.

3.2 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 individuals 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 $N = 20$ airplanes and $N_c = 2$ we encode the trajectories as follows :

$Airplane1$	P_1^1	P_2^1
$Airplane2$	P_1^2	P_2^2
...	...	...
$Airplane19$	P_1^{19}	P_2^{19}
$Airplane20$	P_1^{20}	P_2^{20}

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_i^j. 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 ($\pm\frac{d}{D_{max}}\%$).

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 3 and 4.

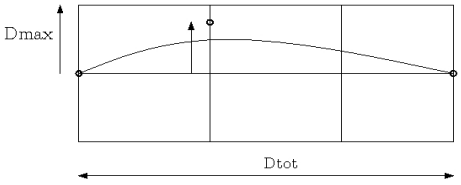

Figure 3: Chromosome encoding and corresponding B-spline with one percentage exceeding 100%

3.3 Crossover and Mutation

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

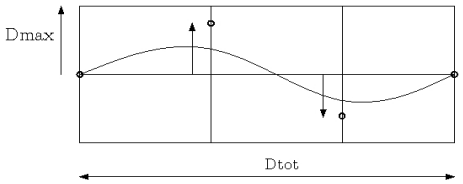

Figure 4: Another chromosome encoding and its corresponding B-spline with both percentages within $[-100\%,-100\%]$

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 5 :

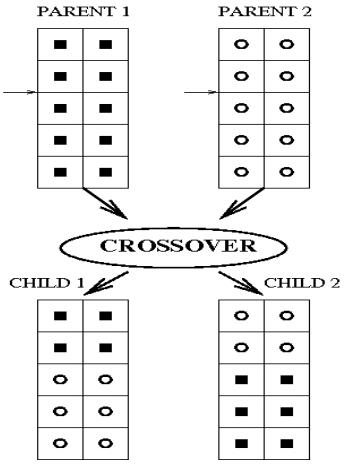

Figure 5: Parent and children chromosomes

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 6).

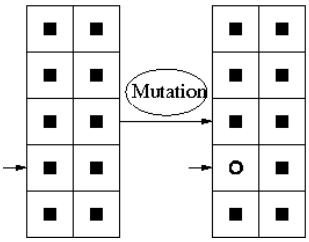

Figure 6: Initial and mutated chromosome

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.

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

$$f(X) = -(CN + (\sum_{i=1}^{N} \frac{NR_i}{BT_i} - N)) \qquad (4)$$

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).

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 7. This whole resolution computation required approximately 4 minutes.

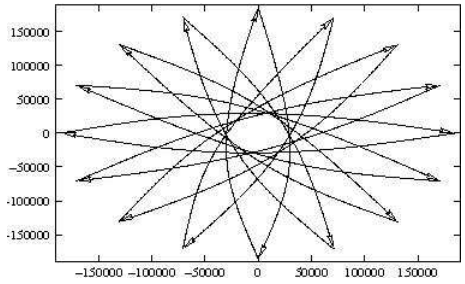

Figure 7: The roundabout configuration after resolution

The fitness' evolution with respect to generations is shown in Figure 8. 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 $[-1,0]$.

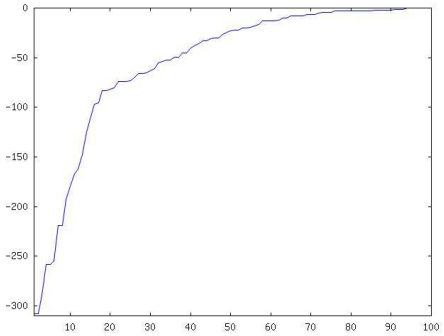

Figure 8: The fitness evolution with respect the generation

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 9.

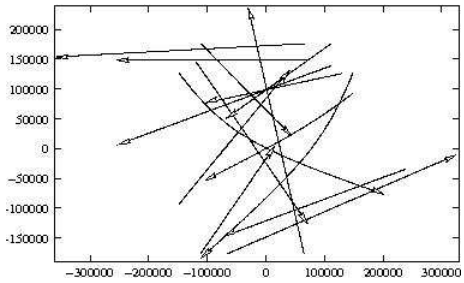

Figure 9: The operational problem after resolution

We can see on Figure 10 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.

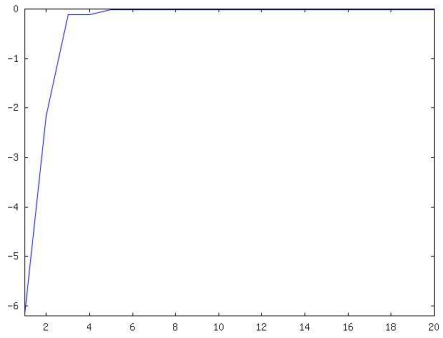

Figure 10: The fitness evolution with respect the generation

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-adaptative* 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 (Conn et al., 2009). Indeed, despite the local aspect of these methods, they can also be adapted to global optimization.

ACKNOWLEDGEMENTS

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

REFERENCES

Alliot, J. and de Verdière, D. C. (2003). L'ATM : 20 ans d'efforts et perspectives. In *Vers l'automatisation du vol et de sa gestion*.

Conn, A., Scheinberg, K., and Vicente, L. (2009). *Introduction to derivative-free optimization*.

Durand, N. (Octobre 2004). *Algorithmes génétiques et autres outils d'optimisation appliqués à la gestion du trafic aérien*. PhD thesis, CENA, Toulouse.

Goldberg, D. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Reading MA Addison Wesley.

Koza, J. (1992). *Genetic Programming*. MIT press.

Rabut, C. (2008). Courbes, outils pour la CAO, approximation à une variable. INSA-Toulouse lecture notes.

Zeghal, K. (1994). *Vers une théorie de la coordination d'actions, application à la navigation aérienne*. PhD thesis, Université Paris VI.