Air-traffic conflict resolution via B-splines and semi-infinite programming

Clément Peyronne[1,2] Daniel Delahaye[2] Laurent Lapasset[1]
Marcel Mongeau[2]

[1] Capgemini, 15 Avenue Docteur Maurice Grynfogel, 31000 Toulouse, France ;
{clement.peyronne,laurent.lapasset}@capgemini.com
[2] Ecole Nationale de l'Aviation Civile, 7 rue Edouard Belin, BP 54005, 31055 Toulouse, Cedex 4, France ; {marcel.mongeau, daniel.delahaye}@recherche.enac.fr

Mots-clés : *Conflict Resolution, Air Traffic Management, Optimization.*

1 Introduction

The most critical point of air-traffic control is 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. Air traffic being constantly increasing, controllers in charge of an aviation sector must handle more and more flights. Nowadays, air-traffic management (ATM) has already used every available ressource to increase airspace capacity. From now to 2030, air-traffic is expected to be further multiplied by a factor two or three [1]. Consequently, ATM will have to deal with this overload while ensuring at least equivalent standards of safety [1].

Recently, two projects have been launched to find solutions for the future of air-traffic control : SESAR (Single European Sky ATM Research) in Europe, and NextGen in the United States. These projects aim at including more automation in ATM, introducing new technologies. Both SESAR and NextGen consider a new ATM concept where the objective is to keep each airplane as close to its original trajectory as possible. The SESAR European project aims at finding solutions to this problem by several actions on different time frames. One of the solutions SESAR encourages is to introduce more automation in air-traffic control, creating an automatic conflict resolution tool to provide a decision support to the controller. In this paper, we choose to begin with full automation as a first step, delaying the problem of providing conflict-free solutions to the controller to a later stage.

In the next section of this paper, we present our air-traffic conflict resolution problem. We describe our B-spline trajectory model in Section 3. Then, we introduce in Section 4 the new optimization formulation of our problem and the semi-infinite approach to tackle it. In Section 5, we detail the necessary reformulation to use derivative-free optimization method. After that, in Section 6, numerical results are presented and discussed and compared with previous results obtained using a stochastic optimization method (genetic algorithms denoted GA) presented in [2]. We draw conclusions and present our future work in Section 7.

2 The air-traffic conflict resolution problem

Our air-traffic conflict resolution problem is defined as follows. We are given the number of aircraft, noted N, involved in the conflict, and for each aircraft, the start and end points are given. We aim at calculating N *optimal* conflict-free trajectories. Optimality can be defined in several ways : for example, optimality in distance or in fuel consumption. Nevertheless, the particular choice of an objective function does not change essentially the way to tackle the problem. Remark that our previous stochastic methodology [2] attempted at minimizing the number of conflicts detected in a discretized space. All aircraft are assumed to start from their starting point at time $t = 0$. In this study, we work in two dimensions as we assume that each aircraft is constrained to follow its given altitude profile. In this study, we aim at solving conflicts considering a 20-minute time horizon and en-route aircraft flying at approximately 400 miles per hour.

3 Trajectory modeling

In this section, we introduce a B-spline trajectory model that relies on only one free control point per trajectory (more details in our previous paper [2]). The B-spline approximation is to find a curve approximating a set of points of R^2 called *control points* which completely defines the curve [5]. We choose to rely on B-spline for 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).

Basically, B-splines are parameterized curves generalizing Bezier curve concept [2]. We define a parameter $s \in [0, n]$ varying from 0 to n, n being the number of control points of the curve. In our model, we choose $n = 3$ control points : the start and end points (which are given) and a third control point which will be determined by our optimization variable vector. We can define the B-spline curve as the following 2D parameterized curve C :

$$C(s) = (\sigma_x(s), \sigma_y(s)), s \in [0, n]$$

Our optimization method will decide the location of the middle control point. In this case we choose to use only one control point subject to optimization for each trajectory (for each aircraft). We consider that more than one turn in this time window would be inconsistent with ATM requirements. The i-th aircraft trajectory is driven by a set of three control points called $\beta_i(u)$ where $i = 1, \dots, N$ the aircraft index.

4 Semi-infinite programming formulation

In this section, we introduce a new optimization model for the optimal conflict resolution problem. Instead of going through this conflict-detection phase as in [2], the new idea here is to consider a continuous separation-distance constraint, making the approach much more direct. This constraint must be respected at all times, leading to an infinite number of constraints. Therefore, we propose modeling the problem with an objective function involving a finite number of optimization variable subject to an infinite number of constraints, giving rise thereby to a semi-infinite optimization problem as in [4]. In this study, we focus on minimizing the total relative distance increase with respect to the direct route.

Recall that to model a trajectory we use B-splines involving one movable control point (the start and end points of the trajectory are not meant to move). Let u be the vector containing all these central control point locations (u_i is the central control point of aircraft i). Let $\gamma^{\beta^i}(s(t))$ be the 2D-position of aircraft i considering its control point locations and time t, described by $\beta^i(u)$ (to clarify notation, from now on we will denote $\beta^i(u)$ as β^i).

Thus, our relative distance increase the objective function is :

$$f(u) = \sum_{i=1}^{N} \frac{T^i(u_i) - L^i}{L^i}$$

where $T^i(u_i)$ is the total travelled distance of the aircraft i following its modified trajectory and L^i is the direct (straight line) route length from the start point to the end point.

Constraints must guarantee conflict-free situations. They have to ensure that all aircrafts are, at least, $\tau := 5$ Nm away from each other (5 Nm is the *standard separation* norm in ATM). For two aircraft i and j, this constraints reads :

$$c^{ij}(u;t) := \|\gamma^{\beta^i}(s(t)) - \gamma^{\beta^j}(s(t))\|_2 \geq \tau, \qquad \forall t \in [0, t^{ij}_{min}],$$

where $||\cdot||_2$ is the l_2 norm, $t^{ij}_{min}(u) = min(t^i_{end}(u), t^j_{end}(u))$, and $t^i_{end}(u)$ is the arrival time of aircraft i to its end point and is calculated from the B-spline trajectory defined through u. Therefore, we choose to formulate our problem as :

$$\begin{aligned} &\min_u f(u) = \sum_{i=1}^{N} \frac{T^i(u_i) - L^i}{L^i} \\ &s.t. \qquad c^{ij}(u;t) \geq \tau \qquad\qquad ,\forall t \in [0, t^{ij}_{min}(u)]; i = 1, ..., N; j = i+1, N. \end{aligned} \tag{1}$$

5 Reformulation for derivative-free optimization methods

Our problem clearly fits into the scope of semi-infinite programming (SIP), with time as the semi-infinite parameter. We first test one of the modern derivative free optimization (DFO) methods [3] in order to have a rough idea of the viability of our problem formulation. DFO do not attempt at approximating derivatives. More precisely, we choose a trust-region based DFO method using simple bounds and a minimum Frobenius norm model. For that purpose, we adapt our formulation to eliminate the constraints, since DFO methods are not designed to handle constraints (for now). We therefore include the constraints as a penalization in the objective function. In order to do that, we first reformulate the constraints to have a *finite* number of constraints In an analogous manner to [4], we rewrite our constraint functions $c^{ij}(u;t)$ as :

$$c^{ij}(u) = \int_0^{t^{ij}_{min}(u)} \max\{\tau - c^{ij}(u;t); 0\} \quad dt = 0.$$

We then include the new constraints in the objective function as a penalization, yielding to the new objective function f_{DFO} as follows :

$$f_{\text{DFO}}(u) = f(u) + \omega \sum_{i=1}^{N} \sum_{j=i+1}^{N} c_{ij}(u)$$

where ω is a user-defined positive weighting parameter. We shall see in the next section that the results we obtain are good.

6 Numerical results

In this section, we compare results obtained with the two optimization approaches (SIP and GA) on simple situations involving a very large number of aircraft. We perform all our tests on a 2.53GHz Intel(R) Core(TM)2 Duo on a Windows XP Operating System. We set $\omega = 0.01$ for the weighting parameter that balances the trade-off between the distance and the conflict criteria. In this study, we consider simple, but difficult, academic conflict situations as test problems for our methodology. Each instance involves N aircraft uniformly distributed on a circle of radius 100 Nm in the first two situations, and 400 Nm for the last ones, where each aircraft is heading towards the center of the circle. We consider four test problems : $N = 8, 16, 32$ and 64.

In the following table, we display the objective-function value $f(u) = \sum_{i=1}^{N} \frac{T^i(u_i) - L^i}{L^i}$ obtained, the number of objective-function evaluations required (denoted *fevals*), and the calculation time (*time*) for both GA and the SIP approaches (the latter relying on DFO and denoted dfoSIP). Remark that, although the objective function is not exactly the same, we compare the same value : the total relative distance increase generated by the conflict resolution.

N	8		16		32		64	
$f(u)$	0.1338	0.0241	0.9556	0.1701	0.9435	0.0777	F	0.223
fevals	10000	321	10000	1984	10000	1448	25000	1879
time	53”	1’19"	8’38”	7’ 33"	13’30”	17’ 50"	*	1h 37’ 24"
method	GA	dfoSIP	GA	dfoSIP	GA	dfoSIP	GA	dfoSIP

In the above table, “F” denotes a failure (no conflict-free solution found) of the method within the allowed budget in terms of number of function evaluations (10000 for $N \leq 32$ and 25000 for $N = 64$).

It appears clearly that the dfoSIP approach performs better than genetic algorithms in terms of solution quality and number of function evaluations. The calculation time however, is similar for both methods (inferior with GA on the 32 aircraft situation) . We also see that, as the size of the problem increases, the difference in quality between GA and SIP increases as well. One cause for this difference in performance, independent of the optimization method itself, is the way we implemented conflict detection in the genetic algorithms : the objective function used in GA (the number of conflicts computed via a discretization of space [2]) always overestimates the separation distance, making thereby the constraints harder to respect. This may be why GA does not find a solution for the 64-aircraft instance. To conclude, we remarked that GA and SIP yield similar solution on each of the smaller instances (8, 16 aircrafts) in terms of traffic organization with better length, CPU time and function-evaluation budget for SIP.

7 Perspectives and conclusion

We introduced a semi-infinite programming of the air-traffic conflict resolution problem that yields very encouraging results even though it does not seek for a global optimum. It appears that our simple dfoSIP approach already performs better than the GA methodology we developed in a previous study [2], where we also modeled trajectories via B-splines. B-splines allows us to reduce drastically the search space dimension.

We already have some work in progress on testing, less symmetric, test instances. We are currently implementing the objective and constraint analytic derivatives in order to be able to apply a classical local optimization method. We then intend to test both of our approaches on real air-traffic and especially on a complete day of traffic over France with a common way to compute the conflict detection (in other words, using exactly the same objective function for both GA and SIP).

Références

[1] J. Alliot, D. Colin de Verdière. L’ATM : 20 ans d’efforts et perspectives. *Vers l’automatisation du vol et de sa gestion*, technical report, ONERA, Toulouse, 2003.

[2] C. Peyronne, D. Delahaye, M. Mongeau, L. Lapasset. Optimizing B-splines using genetic algorithms applied to air-traffic conflict resolution. *ICEC Proceedings*, Valencia, Spain, 2010.

[3] A. Conn, K. Scheinberg, L. Vicente. Introduction to derivative-free optimization. *MPS-SIAM Series on Optimization*, 2009.

[4] C. Visweswariah, R.A. Haring, A.R. Conn. Noise considerations in circuit optimization. *IEEE transactions on computer aided design of integrated circuits and systems*, 2000.

[5] T.Guillod. Interpolations, courbes de Bézier et B-Splines *Bulletin de la Société des enseignants Neuchâtelois de Sciences*, 2008.