AUTOMATED CONFLICT-FREE PLANNING: EXPERIMENTS ON REAL AIR TRAFFIC DATA

Jérémy Omer* , Thomas Chaboud*
* Onera, the French Aerospace Lab
jeremy.omer@onera.fr; thomas.chaboud@onera.fr

Keywords: *automation, numerical optimisation, air traffic control, conflict solving, experiments*

Abstract

This paper focuses on confronting the automated air traffic conflict solving algorithms we developed, to experiments on situations representative of the operational practice. The whole process goes from reading traffic data to using the solver results as aircraft trajectory corrections. Studying its feasibility gives indications on our algorithms' adequate behaviour and the necessary caution in setting the solver constraints. Experiments are done on recorded flight plans with actual time-overs, and a few representative examples are described.

1 Objective

European and world air traffic may be on a 20 years doubling trend, as anticipated by such large-scale strategic concepts as SESAR or NextGen, or could end up stagnating due in part to energy costs and environmental constraints ([1]). Either way, the challenges the air transport industry will have to face can only grow stronger, and technological advances are needed in the coming years.

Foremost among the expected improvements of current Air Traffic Management (ATM) is the extended or even new roles automated systems have to assume in operations. The Air Traffic Control (ATC) capacity being a known bottleneck of ATM efficiency justifies research and development of automated aids for this operational function.

Our work in this area consisted mainly in studying the theoretical framework of optimal and fast enough computational methods deconflicting predicted aircraft trajectories. The needs and limits of convex or linear modellings of the problem constraints were examined, along with the development of several algorithms based on MILP (Mixed Integers Linear Programming) and NLP (Nonlinear Programming) optimisation techniques, and hybrids thereof. In the following, we may simply refer to any of these variants as "the solver".

They were first validated, in terms of performances, cost and trajectory correctness, using difficult but artificial traffic cases.

The capacity of a solver to correctly deal with real air traffic conflicts entails that

- it is able to generate trajectories that are devoid of conflict in terms of its internal modelling, in acceptable time;
- it takes all the parameters relevant in describing the operational situation, as an input;
- the proposed solution can be transformed into flyable and de-conflicted trajectories through speed and heading orders to the aircraft.

These points are an integral part of the practicality of the tool's operational implementation, just as much as the algorithmic complexity, the pertinence of the optimization criteria or the quality of the solutions found. Examining them touches upon the crucial matter of operational constraints modelling and results interpretation, as e.g. the interplay of the aircraft performance models involved.

Note that we do not aim here for an actual operational assessment. Conflict solving automation is only seen as a long-term goal in SESAR ([2]). Current European ATM operations still lack many features that are either indispensable (e.g. agreement on human/machine responsibility sharing in ATC) or very useful (e.g. data-link exchange of flight intentions, performance models and trajectory prediction sharing (cf. SESAR project P5.5.2, 2012)) to automated conflict solving integration. Furthermore, Onera is not working in close technical cooperation with air traffic managers, hence, among other things, lacking in mass traffic data for statistical assessments.

This is not to say that these conflict-solving algorithms must remain a theoretical toy. Indeed, they can be used in low-TRL (Technology Readiness Level) R&D studies based on modular traffic simulations frameworks. For instance, a working conflict solving software helps evaluate a new strategic planning concept through simulations. The advantage is obvious from the point of view of traffic safety; also, it is very difficult otherwise to say anything of the fuel consumption overhead due to ATC actions.

Another instance is the UE FP7 project 4DCo-GC ([3]); in this study of an ATM based on strategical 4D contracts, our solver can be used in pre-simulations of initial flight plans to obtain fully de-conflicted trajectories at the planning stage.

This feasibility work encompasses the whole process of reading traffic data, computing the algorithm input, solving the potential conflicts and using the solutions as trajectory corrections. It gives indications on the algorithms' behaviour and the necessary caution in setting the solver constraints (separation distance, 4D entry and exit points, aircraft performances).

2 Numerical conflict solver

Automated ATC was treated in several studies since 1990 [4, 5, 6, 7, 8]. The automated tool we recently developed is described in [9]. Its specificity is to be conceived with a view to including non linear constraints while still giving some guarantees of convergence. This was achieved by coupling two resolution methods associated with two different models of the same problem.

More details are given on the solver in order to help understanding the results of the experiments conducted in the last sections of this article.

2.1 Formulation of the problem

Given a set $\mathcal{A}$ of aircraft and their reference trajectories on a 10-15 minutes time horizon T, the objective of the solver is to find conflict free trajectories that stay as close as possible to the reference trajectories and minimise fuel consumption. This research may be formulated as the minimisation of a cost function subject to a set of constraints. The cost should represent fuel consumption while constraints should guarantee that aircraft are separated, that their trajectories are actually flyable and that they stay close to the reference trajectory.

As the aircraft motions are assumed to be planar, the trajectories are two-dimensional (2D). The state of an aircraft i at time t is described by its position and speed vectors $\mathbf{p}_i(t)$ and $\mathbf{v}_i(t)$. The aircraft is then controlled by its acceleration $\mathbf{u}_i(t)$, which was judged more realistic than directly controlling speed.

As fuel consumption is somehow related to the aircraft acceleration, the cost function was chosen as:

$$\int_0^T \left(\sum_{i\in\mathcal{A}} \|\mathbf{u}_i\| \, dt \right) \tag{1}$$

In order to keep the model simple, the realism of the trajectories is only ensured by bounding accelerations and speeds.

$$\underline{V_i}^2 \leq \|\mathbf{v}_i(t)\|^2 \leq \overline{V_i}^2 \text{ and } \|\mathbf{u}_i(t)\|^2 \leq \overline{U_i}^2, \forall i \in \mathcal{A} \tag{2}$$

Regarding 4D planning, the cost function already ensures that trajectories deviate as little as possible from the reference trajectories when avoiding loss of separation. The remaining issue is to put the aircraft back on their reference trajectories after manœuvring. This is achieved by constraining the aircraft to recover their initially planned positions and speeds, $\mathbf{p}_i^{end}$ and $\mathbf{v}_i^{end}$, at time T:

$$(\mathbf{p}_i(T), \mathbf{v}_i(T)) = \left(\mathbf{p}_i^{end}, \mathbf{v}_i^{end}\right), \forall i \in \mathcal{A} \qquad (3)$$

The trajectories are conflict-free if separation is maintained between each pair of aircraft in potential conflict. This means that for any pair $(i, j) \in \mathcal{C}$ the set of potential conflicts:

$$\left\|\mathbf{p}_j(t) - \mathbf{p}_i(t)\right\|^2 \geq D^2, \forall t \in [0, T], \qquad (4)$$

where D is the horizontal separation norm (a typical value is $D = 5$ NM).

It was also considered that segregated areas may have to be avoided. For instance, this is necessary if some portions of the airspace are reserved, at least on a temporary basis, for military use. Such area was modelled as a polygonal obstacle o described by affine inequalities $a_e x + b_e y + c_e \leq 0$, $e \in \mathcal{E}_o$. Avoidance of an obstacle is guaranteed by the constraints:

$$\max_{e \in \mathcal{E}_o}(a_e p_{i,x}(t) + b_e p_{i,y}(t) + c_e) \geq 0, \ \forall t \in [0, T], \qquad (5)$$

where $(p_{i,x}(t), p_{i,y}(t))$ are the coordinates of aircraft i at time t.

The optimisation problem containing these criteria and constraints resides in the framework of optimal control. No analytical solution was found yet without further simplifications. This involves that the model should be solved numerically. A discretisation process called *direct transcription* was implemented for this purpose.

2.2 Numerical resolution through NLP

The state and control variables are functions taking their values in a continuous time interval. As no numerical techniques may handle such data, these functions had to be parametrised with a finite number of variables. The most straight forward option was chosen, which means that the time window was sampled into a sequence of $K+1$ instants $0 = t_0 < t_1 < \ldots < t_K = T$. The samples are uniformly distributed according to a time step Δ. The variables are then represented by the finite set of values taken at each time step $\{\mathbf{p}_i^k\}$, $\{\mathbf{v}_i^k\}$ and $\{\mathbf{u}_i^k\}$, $k \in \{0, \ldots, K\}$.

This process is called direct transcription. The first difficulty arising during transcription is due to the differential relations between the variables, as $\mathbf{v} = \dot{\mathbf{p}}$ and $\mathbf{u} = \dot{\mathbf{v}}$. These differential equations have to be integrated numerically. Based on a study made by Paielli on modelling manœuvres [10], this difficulty was alleviated by assuming that acceleration is a piecewise constant function. Variables are then calculated as

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \Delta \mathbf{v}_i^k + \frac{\Delta^2}{2}\mathbf{u}_i^k, \forall i, k \qquad (6)$$

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \Delta \mathbf{u}_i^k, \forall i, k \qquad (7)$$

The second difficulty is due to non convex constraints. For such constraints, it is not sufficient to check their validity at each time step if they have to be respected on the whole time interval. This case is mostly an issue for the separation constraints. Figure 1 illustrates a situation where non convexity leads to a loss of separation that would not be detected with a simple check at each time step. It is however possible to find a simple analytical expression of the minimum distance on each time interval $[t_k, t_{k+1}]$ when speed is constant [9]. By comparing speed with a well chosen constant function it was then possible to find a good approximate constraint guaranteeing that separation is maintained on the whole interval.

The resulting model is a NLP which may be solved by several efficient solvers such as IPOPT[1]. It is important to highlight that, due to non convexity of constraints, the convergence to a global optimum may not be proved. This remark may seem secondary as finding the best set

[1] IPOPT is an open libray distributed on the COIN-OR website https://projects.coin-or.org/Ipopt

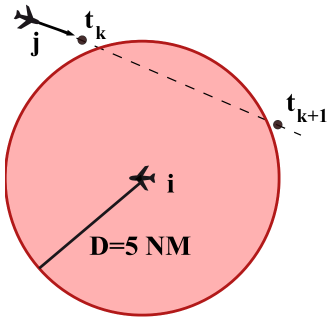

Fig. 1 Concavity of separation constraints

of trajectories is not essential as long as it is conflict free. However, without such proof, it is not even possible to affirm that conflict free trajectories are computed when they exist.

The necessity to provide some guarantees of convergence was then achieved by modelling the problem with a MILP. This family of optimization problem is characterised by a linear objective, linear constraints and both continuous and binary variables. This choice was done because it may be used to get a good approximation of the problem and because such model may be solved optimally. Solving the MILP first guarantees that conflict free trajectories are found when the MILP is feasible. It also provides a good initial solution for solving the NLP which is a more faithful representation of the real problem.

2.3 Modelling ATC as a MILP

ATC was already modelled as a MILP by several authors [6, 7, 8]. The difference with the previously developed approaches is that this model should allow several modifications of speed and heading while assuming that speed is a linear continuous function of time. Most other options did not allow for heading changes [8] or only allowed one speed change [7] or assumed that speed changes were instantaneous and only checked separation at each time step [6].

The formulation of the problem with linear constraints is not natural though. Most constraints are non linear and separation constraints are not even convex. In order to make the constraints linear, it is helpful to notice that they might be represented geometrically thanks to circles. A separation constraint is a circle centred on an aircraft inside of which no other aircraft must go. Upper bounds on speed and acceleration are circles inside of which the extremities of speed and acceleration vectors must remain. The process of linearisation may thus be undergone by approaching the circle with tangents when the forbidden area is inside (Figure 2) or with chords when it is outside (Figure 3).

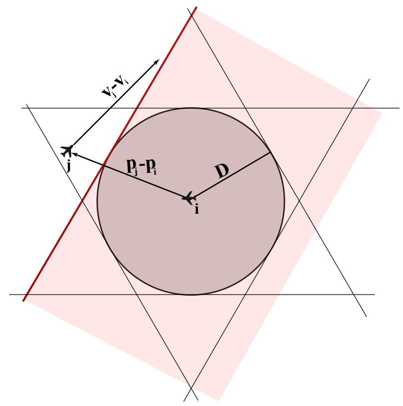

Fig. 2 Approximation of the separation circle with tangents

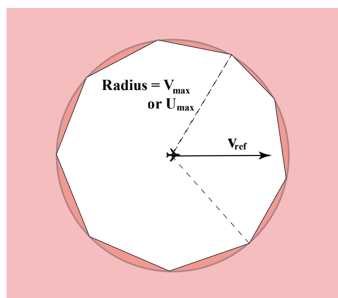

Fig. 3 Approximation of upper bounds with chords

Chords and tangents are straight lines and as such, their equations are linear. Upper bounds are respected if the vectors satisfy all the constraints delimited by the chords. On the opposite, separation is guaranteed if at least one of the constraints delimited by the tangents is respected. Separation is thus approached with a disjunction of linear constraints. This disjunction may only be modelled with binary variables. These variables add a combinatorial complexity to the resolution, as a tree, whose size grows exponentially with the number of binary variables, has to be explored to get the optimal solution. The expression of the constraints is given in [9].

It is also important to notice that lower bounds on speed are similar to separation constraints as they might be represented as a circle outside of which the extremities of speed vectors must lie. It was however observed on theoretical data sets that lower bounds are not necessary because of the constraints on final positions and speeds. It is indeed impossible or very costly to recover the reference trajectory when very small velocities are chosen. Lower bounds on velocities were thus deleted from the model because they would make it much more complex without adding much information to it. As a consequence, there is a need to confirm on real data sets that these constraints may rightfully be neglected.

Experiments were conducted to test the solver on virtual data sets. The results were encouraging as they showed that the solver was able to generate conflict-free trajectories on very complex situations in reasonable computation times. Reality remains a source of unexpected and very specific situations which cannot be found in artificial configurations. The next section of this article aims at building experiments drawn from real traffic.

3 Solving “real” air traffic conflicts : generating problem instances

Our main goal is to test the solver’s ability to handle conflict situations from actual traffic. At this stage we only want to deal with local situations with realistic aircraft trajectories, roughly as they would be presented to a human ATCO monitoring a sector. This means that sets of conflicts taken in isolation are extracted from air traffic data and input to the solver. The solutions are directly analysed without reconstructing the complete trajectories.

Working in tactical (short time notice) conditions renders considering the trajectory predictions as deterministic a reasonable approximation. Under these conditions, a single run of the solver is adequate, though it was primarily studied and developed to be part of an updating control-loop of the traffic.

As the solver represents separation and performance constraints in its own internal model, necessarily different from their operational reality, the translation from one to the other, and the conformity of the solutions proposed, are important matters to study. Working on a set of conflicts taken in isolation is a first step in this direction.

3.1 Traffic data, pretreatment

Large-scale traffic and reliable aircraft performance data are requisite for our purpose. Recorded flight data is a logical choice considering the target applications of our solver and our limited access to operational providers. Radar tracks files may seem the best choice, thanks to their high sampling rate, but are not easily obtained and their size is massive even on a national scale. Moreover, taken in isolation, they are already de-conflicted; one would have to correlate them with the relevant flight plans to get an idea of the aircraft intended trajectories, so as to detect ATCO (ATC Officer) actions to resolve conflicts.

We found an easily adaptable performance model in IESTA aircraft performance module [11] based on BADA [12], and a few ETFMS (Enhanced Tactical Flow Management System) data files from CFMU (Central Flow Management Unit) were available. These files contain about 28000 continental flight plans each, along with both initial (as requested by the airlines) and actual take-off (TOT) and time-over (TO) times. Our solver is not able to handle take-off sequencing. Actual TO have the advantage of providing realistic airport regulation to the traffic.

The next step is then to use these waypoint-to-waypoint flight plans with actual TO as the aircraft original 4D flight intentions. As compared to radar tracks, these ‘trajectories’ do not include direct routings nor conflict resolutions through vectoring. The problems they leave for our solver to manage are thus representative enough, in terms of traffic patterns (time and space distribution), of those handled by ATCOs.

However, ETFMS format rounds TOT and TO dates to the nearest minute. Because of this, some aircraft may seem not to conform to their performance category's speed limits. Moreover, some pairs of aircraft seem to pass the same initial departure fix with very low longitudinal separation. Some pretreatment adjusting TOT and TO is required as unfeasible constraints (speeds, boundary values) do not constitute well-posed problems for the solver. The adjustments were kept to a smooth, and whenever possible small, nudging of TOT and TO to appropriate second-accurate dates.

3.2 Conflicts, conflict clusters

Everywhere two aircraft's trajectories come within separation distance of each other (in 3D), we define a potential conflict if both aircraft are within 2 minutes of their respective closest points. Making a list of every potential conflict in a day's worth of flight data on a continental scale can be computationally challenging. We sped the process up using 2D-hashing (quadtree) techniques on the set of more than 500000 segments (20-30 segments per flight). When using dimensional hashing for geo-location, the complexity of conflict detection remains quadratic, but is reduced by a large constant factor (proportional to the quadtree size) in average cases. Note that hashing initially adds a linear overhead.

For each potential conflict detected, the solver is input a list of parameters describing the traffic situation in the chosen time window. This description includes the horizontal separation distance, the time window duration, and for every aircraft involved, along local coordinates:

- its initial and final positions;
- its initial and final speeds;
- its maximal and minimal speeds in the current context;
- its maximal acceleration.

The performance limits, depending on altitude and temperature around the conflict center point are computed from the aircraft's performance categories.

We did not consider conflicts below FL195, as the solver is specialised in 2D resolution of en-route conflicts. In the case of conflicts just above this limit, the time window (15 minutes, centred on the time of closest approach) may reach some of the initial or final segments of the flights involved. This stresses the need for the TOT and TO adjustments mentioned in 3.1.

What are the aircraft to consider in a given conflict situation? In theory, to generate a conflict-free traffic, one needs to:

- simulate the primary conflicts solving
- detect eventual secondary conflicts created by the avoidance manœuvres
- add the aircraft involved before solving the new, bigger instance
- and so on, until a fixed point is reached

More formally, let A and B be aircraft, A_{T_0,T_1} be the partial trajectory of A between times T_0 and T_1. Let A_{T_0,T_1} be related to partial trajectory B_{T_2,T_3} if B, between times T_2 and T_3, is potentially affected by the solver's actions on A's trajectory between times T_0 and T_1. The total set of aircraft trajectories that must be taken into account in any resolution involving aircraft A between T_0 and T_1 is the transitive closure on $\{A_{T_0,T_1}\}$ of this relation. Such sets are often called conflict *clusters* (e.g.[13]), though the exact definition of a cluster may differ from one author to another.

If the traffic is dense enough, computing a conflict's cluster may already be costly. Worse, the resulting cluster's size cannot be bounded *a priori*, as it can snowball in every dimension. This could create unmanageable situations for the ATCO/solver, due to lack of time (computational complexity) and room in airspace.

This problem potentially exists in current ATM, and is managed by pre-emptively limiting clusters growth through ATFCM (Air Traffic flow and Capacity Management) measures that create buffer space: air route design and flight level forcing leave some geographical buffer, whereas strategic regulations such as take-off slots or miles-in-trail procedures leave time buffers. In terminal areas where traffic density may reach critical levels at peak hours, special procedures, such as stacks and space reserved for 'tromboning', create extra buffers [14].

In our experiments, we built the clusters around the partial trajectories involved in the primary conflict by detecting potential extended separation loss, that is, using our regular conflict detection with larger horizontal separation distance. In practice, a single pass of this detection with extended separation set to $15NM$ proved sufficient for the traffic samples tested, returning maximum clusters of 4 aircraft only: the actual en route space-time traffic density is rather low.

The upside is that it shows the European ATM filters, strategic (ATFCM) and pre-tactical (inter-sectors or inter-control centres traffic delivery), do smooth the density and improve safety. Note also that the solver minimises the aircraft's acceleration through its cost function (cf. 2.1), which means that in the absence of effective conflicts, its action on the trajectories will tend to be a spatially contracting function.

The downside is that these tests leave the solver well below the traffic amount it showed it could handle in virtual data sets (cf. 2.3). Building more challenging examples from realistic data will need applying traffic increase methods (Section 5) on current samples.

4 Experiments and results

Over a day's worth of traffic (about 28000 flights), we detected 1580 potential conflicts above FL195. Of course, none of them was frontal, thanks to the orientation scheme of flight levels. It was surprising that no crossing conflict occurred either, but it is due to our trajectory generation method which forces the aircraft to stay strictly on the route network. Its design, and this particular day's flight schedule, were such that all crossings could be dealt with through 'merge-and-split' structures. Figure 6 gives a good illustration of one.

The conflicts were all either of the 'catching-up' (two aircraft on the same track with the same heading) or 'merging' (two aircraft converging at low angle to join the same track) types.

Once the traffic samples pre-treated so that their boundary conditions were feasible with respect to the performance and separation constraints, the solver had no trouble finding correct solutions. A few marginal performance infringements appeared in the resulting proposed trajectories. They are due to too small margins left, both in the pre-treatment process (some aircraft too close to the edges of their flight envelope) and in the performance constraints input to the solver. The current progress of our analysis does not allow to conclude on this matter.

A few examples illustrate the solver's favoured resolutions' geometry.

Fig. 4 Short-cut resolution

Figure 4: Both aircraft have the same initial trajectory (red, white); the conflict is due to the second one catching up on the first. The solver proposes short-cuts for both, with speed adjustments (lower speeds), that meet the exit 4D target. The original route detour is probably due to a military segregated area; we do not have data about these. Even then, depending on the local FUA (Flexible Use of Airspace) implementation,

the solution may be licit. Otherwise, with relevant information, the solver is able to take forbidden zones into account in its resolutions.

Fig. 5 Three take-off from Heathrow

Figure 5: Original trajectories are represented in yellow and red, the corrected ones in green and blue. The first segment is fictitious, as ETFMS data does not describe the SID (Standard Instruments Departure) procedure geometry. The three flights take off from Heathrow, heading to Düsseldorf, Amsterdam and Budapest. Note that the solver puts the 'yellow' aircraft away, on the longer green trajectory, before to let it back on its destination track. In effect, this choice is symmetric to the usual approach trombone maœuver.

Figure 6: Original trajectories are represented in red and yellow (from Heathrow) and orange

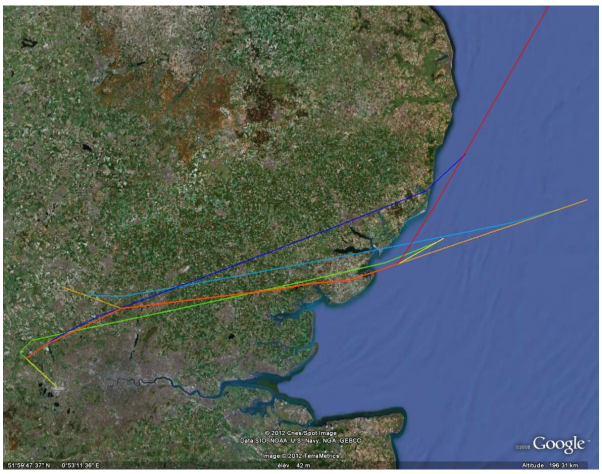

Fig. 6 From merge-and-split to crossing

(from Luton), the corrected ones in dark blue, green and light blue respectively. The solver overrides the air route network 'merge-and-split' design on these tracks and suggests a safe crossing. In general the route network increases traffic predictability, but is sub-optimal in terms of fuel efficiency.

Pictures were created thanks to Google Earth™mapping service.

5 The way forward

Working on a few samples was enough to reckon the practical feasibility of providing examples from actual traffic to the solver, and observe the behaviour of the overall conflict processing software. The results are sufficient to take the next steps, beginning with statistical analysis on the processing of much larger sets of traffic data.

As mentioned in the comments to Figure 4, gathering and exploiting data on segregated airspaces would increase realism, and pose more challenging riddles to the solver around them. Still, the status schedule of these zones must be valid in the period covered by the traffic data sample, as the flight plans input in the solver take them into account.

The same aircraft trajectories we used can also be formatted as a chronological stream; impending conflicts could be computed on-the-fly, and the solver be part of a continuous control loop. Such a setting is also a natural way to include uncertainties on wind, aircraft performance and trajectory prediction. It will thus constitute a test bed for the stochastic programming variants of the solver (work in progress).

Increasing the traffic density two- or threefold would allow to demonstrate that our algorithms can handle much higher complexity, besides the need to evaluate 2030+ ATM concepts in optimistic scenarios. It is not a trivial task if one wants to keep some measure of realism regarding demand patterns. The easiest part is respecting operational conditions (airport capacity, matching of aircraft types to their mission profiles). In addition, the traffic distribution must at least conform to current or forecast statistical trends. For instance it is important to reflect peak or low hours, relative business or quietness of certain airports, airlines categories, and so on. A possibility resides in the aircraft cloning techniques that have been used for some years by the Eurocontrol FAP team [15]. Once a traffic demand is obtained, one still needs to devise and apply ATFCM measures at a strategic level (routing, regulation) to prevent traffic bunching and limit conflict clusters.

In order to prevent clusters from growing too much, it could be efficient to fence up the manœuvres produced by the solver inside a limited space. This is made possible by the 'segregated area' constraints included in the solver model: the inclusion of virtual no-fly zones, with well chosen boundaries, in a conflict set input to the solver, would force it to find a solution in restricted space.

Finally, we do not have a conflict solver specialised in arrival procedures at our disposal yet, though we are part of an internal Onera effort towards designing one. Departure management is relatively easier, being subject to fewer uncertainties: in practice, the aircraft are separated in initial climb thanks to the runway take-off frequency.

References

[1] EREA, "Erea vision for the future - towards the future generation of air transport system," Association of European Research Establishments in Aeronautics, Tech. Rep., 2010.

[2] Eurocontrol, "Sesar definition phase: The concept of operations at a glance," SESAR, Technical Report, 2008.

[3] A. Joulia and C. LeTallec, "Aircraft 4d contract based operation: the 4dco-gc project," in *Congress of International Council of Aeronautical Sciences (ICAS 2012)*, sept. 2012, to appear.

[4] N. Durand, J.-M. Alliot, and J. Noailles, "Automatic aircraft conflict resolution using genetic algorithms," in *Symposium on Applied Computing*, 1996.

[5] A. Bicchi and L. Pallottino, "Optimal cooperative conflict resolution for air traffic management systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 100–109, 2000.

[6] A. Richards and J. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference*, vol. 3, 2002, pp. 1936–1941.

[7] A. Vela, S. Solak, J. Clarke, W. Singhose, E. Barnes, and E. Johnson, "Near real-time fuel-optimal en route conflict resolution," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 826 –837, dec. 2010.

[8] A. Alonso-Ayuso, L. Escudero, P. Olaso, and C. Pizarro, "Conflict avoidance: 0-1 linear models for conflict detection & resolution," *TOP*, pp. 1–20, 2011.

[9] J. Omer and J.-L. Farges, "Automating air traffic control through nonlinear programming," in *International Conference on Research in Air Transportation (ICRAT), Berkeley, California, 22/05/2012-25/05/2012*. http://www.icrat.org: ICRAT, mai 2012, p. (on line).

[10] R. A. Paielli, "Modeling maneuver dynamics in air traffic conflict resolution," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 3, pp. 407–415, May-June 2003.

[11] S. A. et al., "Evaluating the local environmental impact of air traffic with iesta: Outputs and validation walkthrough," in *Congress of International Council of Aeronautical Sciences (ICAS 2010)*, sept. 2010.

[12] "User manual for the base of aircraft data (bada)," EUROCONTROL, Tech. Rep. 11/03/08-08, 2011.

[13] N. Durand, "Algorithmes génétiques et autres outils d'optimisation appliqués à la gestion de trafic aérien," Ph.D. dissertation, HDR de l'Institut Polytechnique de Toulouse, 2004.

[14] K. Roy, *A hierarchical framework for air traffic control*. ProQuest, 2011.

[15] APN/CEF, "Capacity assessment & planning guidance," Eurocontrol, Tech. Rep. Edition September 2007, 2007.

5.1 Copyright Statement