Optimal resolution of en route conflicts

Géraud Granger, Nicolas Durand, Jean-Marc Alliot

Optimal Resolution of En Route Conflicts

Géraud Granger, Nicolas Durand, Jean-Marc Alliot
Laboratoire d'Optimisation Globale*
7, av E. Belin 31055 Toulouse Cedex France
tel: (33) 5 62 17 40 54 - fax: (33) 5 62 17 41 43
email: granger@recherche.enac.fr - durand@cena.dgac.fr - alliot@dgac.fr

Abstract

Designing a Control Simulator involves many difficult problems: modeling conflict detection, trajectory uncertainties, solving conflict inside sectors, respecting military areas constraints, coordinating aircraft between sectors, etc...

Moreover, the n-aircraft conflict resolution problem is highly combinatorial and cannot be optimally solved using classical mathematical optimization techniques. The set of admissible solutions is made of many unconnected subsets enclosing different local optima, but the subset enclosing the optimum cannot be found a priori.

In this paper, we present a conflict solver and its implementation in an Air Traffic simulator, with statistical results on real traffic in a French Sector. This solver, which takes into account real flight plans, solves every conflict inside a sector or over the French airspace on a loaded day.

Introduction

Whereas civil aviation authorities all around the world are engaging more and more controllers as the traffic increases, it seems that the available power of computers is not used and principles of control have not changed for the last 30 years: tools provided for Air Traffic Control (ATC) remain very basic, whereas aircraft are highly automated and optimized systems.

When building a new aircraft, engineers are now able to simulate the aircraft behavior before building a prototype. Many choices can thus be made before tying up expenses.

The same principle should be applied for designing future control systems or simply test minor changes on existing ones. The CATS/OPAS traffic simulator was designed to answer to this objective.

The first versions of the simulator [DA97] were able to handle conflict detection and resolution in a Free-Route context: aircraft were supposed to fly direct routes from origin to destination, the airspace was not divided in sectors, conflicting aircraft were dynamically clustered before resolution. The simulator was able to test Free Flight hypotheses but not to simulate the controller's context and task.

A new modeling is introduced in this article to handle uncertainties on aircraft speeds, climbing and descending rates on flight plan routes.

The first part of the paper presents the state of the art for problem solvers and discusses the constraints hypothesis and goals chosen. Modeling is introduced in the second part. These two parts widely rehearse the article presented at the first USA/Europe ATM $R\&D$ Seminar in Saclay [DA97], but were added for the sake of completeness and clarity. Part three details the maneuver modeling in the flight plan routes context and brings in the conflict solver. Some examples of resolution on real traffic and statistical results are given in part four.

1 Automatic conflict resolution

1.1 State of the art

Many people have studied the two aircraft problem (for example [JHS99]) which is certainly the most frequent but the easiest to solve. Projects, as for example AERA [NFC+83, Nie89b, Nie89a] or Free-R [VNDN97] failed on the question: what happens if more than two, three, four ... aircraft are involved in the same conflict or how can you guarantee this will never happen ?. Durand and Alliot showed [Dur96] that in such cases, conflict resolution becomes a very complex and combinatorial mathematical problem. Feron [JHOF98, EFF99] gives a method to solve n aircraft conflict but the modeling required to simplify the problem is completely unrealistic.

*The LOG is a common laboratory of the Centre d'Etudes de la Navigation Aérienne and the Ecole Nationale de l'Aviation Civile

Other project failed on the modeling. ARC-2000 [K+89, FMT93] optimized aircraft trajectories using 4 dimensional cones and priority rules between aircraft. Optimum was not reached, and the system relied on the availability of FMS-4D for all aircraft, with no uncertainty on speeds. Zeghal [Zeg94], with reactive techniques for avoidance, gave a solution to the problem of automation which was robust to disturbance, but completely disregarded optimization. Furthermore, the modeling adopted implied a complete automation of both on board and ground systems and required speed regulation which could not be handled by human pilots and would probably be very difficult to apply to aircraft engines without damaging them.

A first approach to conflict resolution by stochastic optimization algorithms (genetic algorithms) was done by Alliot and Durand [AGS93]; more advanced results are presented in [DASF94, DAN96].

1.2 Specifications of the system

The main idea, guiding the design of the solver introduced in this paper, is to be as close as possible to the current ATC system:

Constraints: the solver has to handle the following constraints:

- Conflict free trajectories must respect both aircraft and pilot performances. Considering the evolution of ATC toward automation [DAM93], trajectories must remain simple for controllers to describe as well as for pilots to understand and follow.
- Trajectories must take into account uncertainties in aircraft speed due to winds, turbulence, unusual load, etc. Vertical speed uncertainties are particularly important.
- Flight plans must be respected as much as possible.
- Maneuver orders must be given with an advance notice to the pilot. When a maneuver has begun, it must not be called into question.

Goals: We want to achieve the following goals:

- find conflict free trajectories
- Simultaneously minimize different criteria :
  1. the number of maneuver orders
  2. the conflict resolution duration
  3. the delay due to maneuvers
- compute these trajectories in real time.

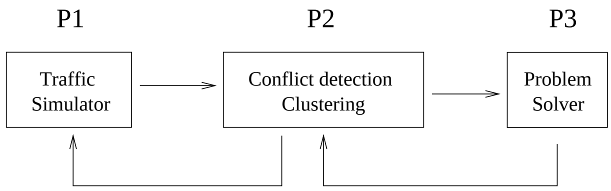

Figure 1: General architecture

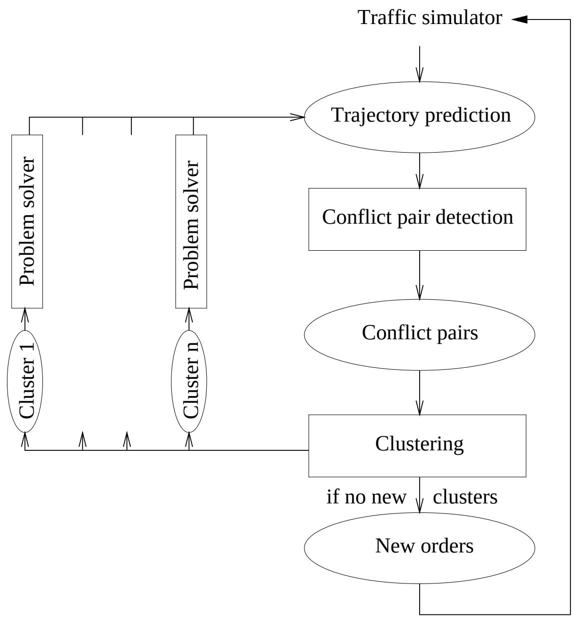

Figure 2: Detailed architecture of the prototype

2 Modeling

2.1 General architecture of the system

We just sketch here the architecture of the simulator; each part will be detailed in the following sections. The system architecture is presented in figure 1 and 2. The system relies on three main processes P1, P2, and P3:

- P1 is the traffic simulator.
- P2 is in charge of conflict pair detection, clustering of pairs, and verification of new trajectories built by the solver.
- P3 is the problem solver.

P1 sends current aircraft positions and flight plans to process P2. Process P2 builds trajectories forecast for T_w minutes, does conflict detection by pairs and transforms 1-to-1 conflicts in n-aircraft conflict. Then, process P3 (the problem solver) solves in parallel each cluster, as aircraft in each cluster are independent from aircraft in the other clusters. The problem solver sends to P2 new orders and P2 builds new trajectories forecast based on these orders. Then P2 once again runs a conflict detection process to check that modified trajectories for aircraft do not interfere with aircraft in another cluster, or with new aircraft. If no interference is found, new flight orders are sent to P1. If there are interferences, interfering clusters are joined and the problem solver is used again on that (these) cluster(s). The process is iterated until no interference between clusters remains, or no new aircraft is concerned by modified trajectories. The new orders are sent back to the traffic simulator.

The above process is iterated and all trajectories are optimized each δ minutes. However, during the computation time, aircraft are flying and must know if they must change their route or not. δ should be large enough to compute a solution, send it to the pilot and let him time enough to begin the maneuver. Consequently, for each aircraft, at the beginning of the current optimization, trajectories are determined by the previous run of the problem solver and cannot be changed for the next δ minutes.

2.2 The Air Traffic simulator

One of the main goals of this project was to test the algorithms on real traffic. The Air Traffic Simulator takes as input flight plans given by companies and pilots: no pre-regulation is done neither on departure time nor on requested flight levels. Consequently, flight plans only have to be deposited T_w minutes before take off.

The simulator uses a tabulated model for modeling aircraft performances: for a given aircraft type, it gives a vertical speed and a ground speed which depends on the aircraft attitude (whether it is climbing, leveled or descending). For example, a B747 leveled at FL-300 has a GS of 490 kts. If it is climbing, its GS will be 480 kts and its VS 1000 fts/mn. At FL-150, values would be respectively 430, 420 and 1800. Aircraft performances are in tabulated form describing ground speed, vertical speed, and fuel burn as a function of altitude, aircraft type and flight segment (cruise, climb or descent.) The main dataset for aircraft flight performance used is the base of aircraft data (BADA) performance summary tables derived from the total energy model of EUROCONTROL. 69 different aircraft types are described. Synonym aircraft are used to model the rest of the fleet. The Airbus A320 (EA32) is used as default aircraft.

All aircraft speeds are modified by a random value to take into account uncertainties on different factors (aircraft load, winds, etc…) This value can be either computed once at aircraft activation and remains the same for all the flight, or can be modified anytime during the flight. The conflict detector and the conflict solver are impervious to the way this value is computed as long as it remains inside a given interval. Uncertainty modeling for conflict detection and resolution is discussed later in the article.

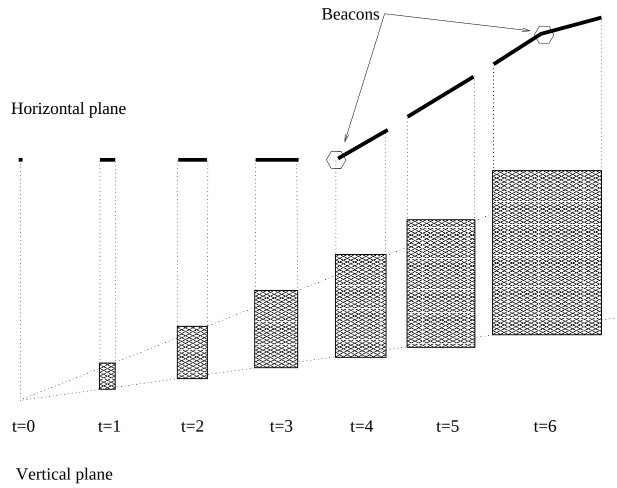

Figure 3: Modeling of speed uncertainties.

Aircraft follow classical routes (from way-point to way-point). The flight model is simple: an aircraft first climbs up to its RFL, then remains leveled till its top of descent, then descends to its destination.

Aircraft fly with a time step that can be chosen at the start of the simulation. The time step is always chosen in order to guarantee that two aircraft face to face flying at 500 kts could not cross without being closer than one standard separation at at least one time step. For most of our simulation, we use a $15s$ time step.

2.3 Conflict detection and clustering

2.3.1 Trajectory forecast and 1-to-1 conflict detection

As described above, the P2 process does trajectory prediction for T_w minutes. This trajectory prediction is done again by a simulation on a slightly modified version of the Air Traffic simulator. But, as stated above, we assume that there is an error about the aircraft's future location because of ground speed prediction uncertainties[1]. The uncertainties on climbing and descending rates are even more important. As the conflict free trajectory must be robust regarding these and many other uncertainties, an aircraft is represented by a point at the initial time. But the point becomes a line segment in the uncertainty direction (the speed direction here, see figure 3). The first point of the line "flies" at the maximum possible speed, and the last point at the minimum possible speed.

[1]Uncertainties on ground track will not be considered, as they do not increase with time and will be included in the standard separation

When changing direction on a beacon, the heading of the line segment's "fastest point" changes as described on figure 3.

To check the standard separation at time t, we compute the distance between the two line segments modeling the aircraft positions and compare it to the standard separation at each time step of the simulation.

In the vertical plane, we use a cylindrical modeling (figure 3). Each aircraft has a mean altitude, a maximal altitude and a minimal altitude. To check if two aircraft are in conflict, the minimal altitude of the higher aircraft is compared to the maximal altitude of the lower aircraft.

Let's take an example. A B747 is leaving its departing airport (altitude 0) at $t = 0$. Its climb rate is 1800 fts/mn and its ground speed is 175 kts. If we suppose that ground speed uncertainty is 5% and vertical speed uncertainty 20%, maximal and minimal climb rate are $1800 \times 1.2 = 2160$fts and $1800 \times 0.8 = 1440$ fts/mn and ground speeds are respectively 184 and 166 kts. This means that $15s$ later, the fastest and higher point has traveled 0.76 Nm and 540 fts while the slowest and lowest has only traveled 0.69 Nm and 360 fts. But this time, when computing maximal and minimal speeds, the difference of altitude of both points must be taken into account. At 540 fts, the tabulated model gives a standard ground speed of 197 kts, so max ground speed is $197 \times 1.2 = 237$ kts. At 360 fts, standard ground speed is 189 kts, with a minimal ground speed of 151 kts. So, the height of the segment grows much faster than the 20% factor for some aircraft.

Duration T_w can be changed, but must be at least equal to $2 \times \delta$. A good evaluation of T_w is difficult. With a perfect trajectory prediction, the larger T_w, the better. However, this is not true as soon as uncertainties are included in the model. A large value of T_w induces a large number of 1-to-1 conflict, as sizes of segments modeling aircraft positions grow quickly with time. Therefore, the conflict solver can become saturated.

2.3.2 Clustering

After pair detection, P2 does a clustering which is a transitive closing on all pairs. Each equivalence class for the relation "is in conflict with", is a cluster.

For example, if aircraft A, B are in conflict in the T_w window, and if B is also in conflict with C in the same time window, then A, B, C is the same cluster and will be solved globally by the conflict solver.

The conflict solver sends back to P2 maneuvers orders for solving conflicts. Then P2 computes new trajectories for all aircraft and checks if new interferences appear. For example, if the new trajectory given to aircraft B to solve conflict with A and C interferes with cluster D, E and with aircraft F, then A, B, C, D, E, F will be sent back to the problem solver as one conflict to solve.

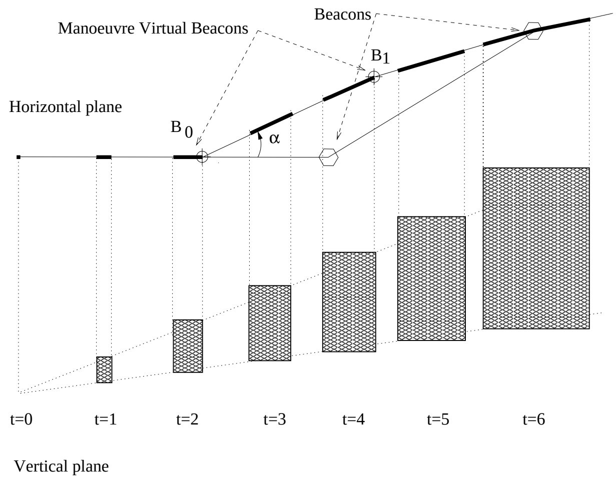

Figure 4: Horizontal maneuver modeling.

The process will always converge: in the worst case, P3 will have to solve a very large cluster including all aircraft present in the next T_w minutes. However, this technique is usually efficient as a very large number of clusters can be solved very quickly in parallel.

3 The conflict solver

3.1 Maneuver modeling

In the horizontal plane, classical maneuvers given to aircraft are heading deviation. In the simulator, 10, 20 or 30 degrees deviations will be allowed. The deviation starts on a virtual beacon created on the route (see figure **??**). This beacon is defined by the position of the head of the segment at some time t_0. It ends on a second virtual beacon, position of the head of the segment at time t_1. An angle criteria is defined to find on which beacon the modified and initial routes should connect.

A maneuver will be determined by:

- t_0 which defines the first virtual beacon B_0.
- the deviation angle α.
- t_1 which defines the second virtual beacon B_1.

In the vertical plane, the aircraft trajectory is divided in 4 periods (figure 5):

- Climbing period. In this period, aircraft can be leveled at a lower than requested flight level to solve a conflict. The aircraft climb is stopped at flight level

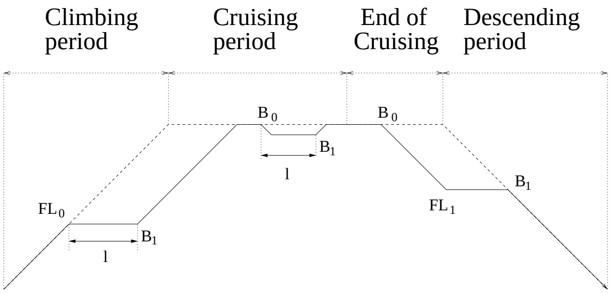

Figure 5: Vertical maneuver modeling.

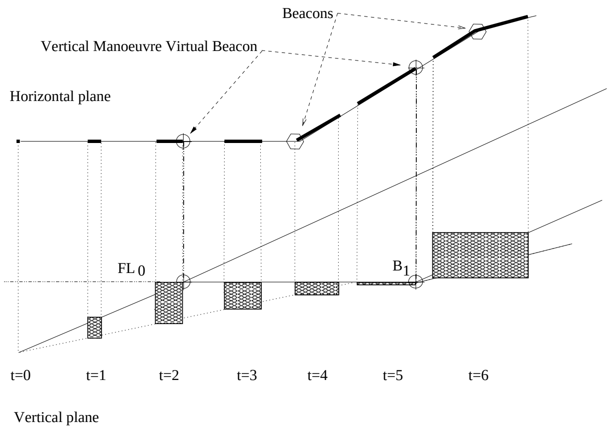

Figure 6: Vertical maneuver during the climbing period.

FL_0 and starts again on a virtual beacon B_1 as stated on figure 6. FL_0 and B_1 are defined by the position of the head of the uncertainty segment at time t_0 and t_1.

- Cruising period. When aircraft have reached their desired flight level, they may be moved to the nearest lower level to resolve a conflict. Aircraft starts descending when reaching a virtual beacon B_0 and starts climbing at B_1 ($\alpha = 0$, B_0 and B_1 are defined by the position of the head of the uncertainty segment at time t_0 and t_1). An example of maneuver is represented on figure 7.
- End of Cruising period. When aircraft are about 50 nautical miles from beginning their descent to destination, they may be moved to a lower level to resolve a conflict. Aircraft start descending on B_0 and are leveled at FL_1 ($\alpha = 0$) (see figure 8). B_0 and FL_1 are defined by the position of the head of the uncertainty segment at time t_0 and t_1.
- Descending period. During this period no vertical maneuver is possible.

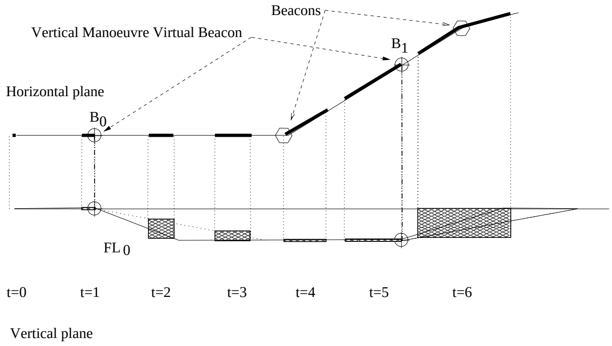

Figure 7: Vertical maneuver during the cruising period.

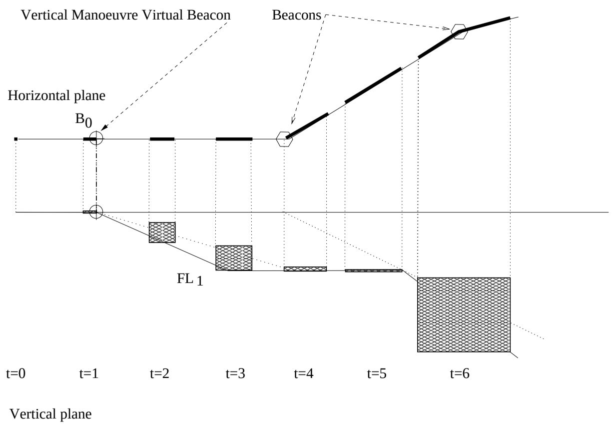

Figure 8: Vertical maneuver during the end of cruising period.

No maneuver will be simultaneously done in the horizontal and vertical plane. This model has the great advantage of reducing the size of the problem. In order to solve conflict due to aircraft taking off or entering the airspace simultaneously at the same point, a variable of delay t_d is introduced.

For a conflict involving n aircraft, the dimension of the search space is $4\,n$. This will allow us to solve very difficult conflicts with many aircraft without investigating a large solution space.

3.2 Maneuver decision time

Because of uncertainties, a conflict that is detected early before it could occur may finally not happen. Consequently, deciding to move an aircraft in that case could sometimes be useless, and could even generate other conflicts that would not occur if no maneuver had been decided. This explains why controllers do not solve conflicts too early. With the turning point modeling, when there is no uncertainty, the earlier the maneuver is started, the lower the delay. However, if speed is not strictly maintained, the earlier the conflict is detected, the lower the probability it will actually happen. Thus, a compromise must be reached between the delay generated and the risk of conflict.

3.3 Choosing the model

Initially, aircraft are allowed to use their flight plan routes.

If we do not want to call into question previous maneuvers and be able to solve very large conflicts, we must try to start maneuvers as late as possible with respect to the aircraft constraints. This argument is enforced by the fact that we allow aircraft to have large uncertainties on their speeds[2].

For example, the first trajectory of figure 9, at $t = 0$, cannot be modified before $t = \delta$. At the end of the first optimization run, at $t = \delta$, the current position of the aircraft is updated. The maneuver that occurred between $t = \delta$ and $t = 2\delta$ is kept as a constraint for the second optimization run (on the example, no maneuver is decided). In the above example, we can see that the maneuver described on line 2 (resulting from an optimization at $t = \delta$) is more penalizing than the maneuver described on line 3 (resulting from an optimization at $t = 2\delta$). This phenomenon occurs

[2]We do not plan to solve conflicts by speed modifications. Theoretical study shows that optimal En Route conflict resolution by speed modifications would require large anticipation time (anticipation time depends on different parameters such as angle of convergence, speed margins for each aircraft, standard separation etc; more details can be found in [Dur96]). This is quite unrealistic due to aircraft speed uncertainties.

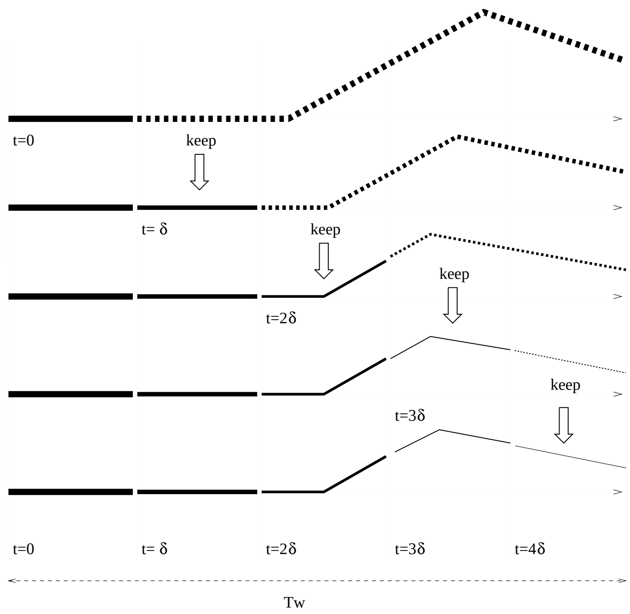

Figure 9: The model and real time optimization.

because of uncertainties. If uncertainties on speed are important, having a small δ will be very helpful to minimize the resolution costs in the real time situation.

Pilots should only be given maneuver orders that will not be modified; if no conflict occurs, no order will be given.

3.4 The function to optimize

One of the principal algorithm design challenges is to define a suitable function to optimize. A multiple-criteria function is required that simultaneously attempts to:

- minimize the delay due to deviations imposed on aircraft.
- minimize the total number of resolution maneuvers required and the total number of aircraft that will be moved[3].
- minimize the maneuver duration so that aircraft are freed as soon as possible for maneuvers that may be necessary subsequently.
- enforce all separation constraints between aircraft.

Instead of considering a single scalar value that takes into account the different lengthenings of trajectories, the number of maneuvers and the conflicts between the aircraft, the contributions from each separate aircraft pair are

[3]Thus, instead of sharing the global delay on all the aircraft, some aircraft will support a part of the delay and others will not.

maintained in a matrix F of size $n \times n$ (where n is the number of aircraft):

- If $i < j$, $F_{i,j}$ measures the conflict between aircraft i and j in the optimization time window T_w. It is set to 0 if no conflict occurs in this period and increases with the severity of the conflict. At each time step t, we compute $C_{t,i,j}$ as the difference (when positive) of the standard separation and the distance between the polygons i and j describing aircraft i and j position at time t. These values are added and give a measure of the conflict between i and j.

$$F_{i,j} = \sum_{t=0}^{total\,time} (C_{t,i,j})$$

- If $i > j$, $F_{i,j}$ measures the efficiency of the resolution between aircraft i and j. It is set to 0 if no conflict can happen between i and j after the optimization time window T_w. If a conflict may remain after this period, $F_{i,j}$ gives a bad mark to pairs of aircraft for which the difference of heading and speed are small (these conflicts are difficult to solve).

- $F_{i,i}$ (see equation 1,2,3) measures the takeoff (or entering) delay given to aircraft i (C_d is a constant), the maneuver duration time $(t_1 - t_0)$ and trajectory lengthening (C_s is a constant depending on the maneuver angle s), and the number of maneuvers (C_m is a constant multiplied by 1 if a maneuver is supposed to become definitive and 0 if not):

$$F_{i,i} = C_d\, t_d \quad (1)$$
$$+ \; C_s\,(t_1 - t_0) \quad (2)$$
$$+ \; C_m\,[(t_0 \leq 2\,\delta)\,\&\,(t_1 > t_0)] \quad (3)$$

This matrix contains much more information than a scalar global value F, and is useful in the optimization algorithm used.

However, a global scalar value is required, and can be defined as follows:

$$\exists(i,j),\; i \neq j,\; F_{i,j} \neq 0 \Rightarrow F = \frac{1}{2 + \sum_{i \neq j} F_{i,j}}$$

$$\forall(i,j),\; i \neq j,\; F_{i,j} = 0 \Rightarrow F = \frac{1}{2} + \frac{1}{2 + \sum_{i \geq j} F_{i,j}}$$

The choice of this function guarantees that if the value is larger[4] than $\frac{1}{2}$, no conflict occurs in the optimization time window. If a conflict remains, the function does not take into account the delays induced by maneuvers. When the value is smaller than $\frac{1}{2}$, maximizing the function minimizes the remaining conflicts. When the value is larger than $\frac{1}{2}$, maximizing the function minimizes the possible remaining conflicts after the optimization time window, the number of maneuvers, their duration, and the delays induced by maneuvers. When no conflict and no maneuver occurs, the function is equal to 1.

[4]Our priority is to find trajectories without conflict.

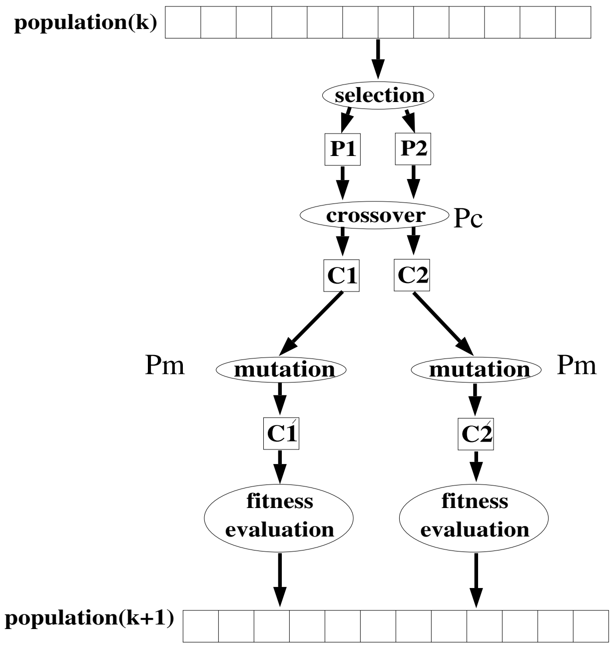

Figure 10: GA principle

3.5 A global optimization problem

Use of local methods, such as gradient for example, is useless here, because these methods rely on the arbitrary choice of a starting point. Each connected component may contain one or several local optima, and we can easily understand that the choice of the starting point in one of these components cannot lead by a local method to an optimum in another component. We can thus expect only a local optimum.

3.6 Genetic Algorithms applied to conflict resolution

Genetic algorithms (GAs) are global stochastic optimization technics that mimic natural evolution. They were initially developed by John Holland [Hol75] in the sixties. The subject of this paper is not GAs and the interested reader should read the appropriate literature on the subject [Gol89]. The general principles are given on figure 10.

Genetic algorithms are a very powerful tool, because they do not require much information and are able to find many different optima that can be presented to a human operator.

Moreover, we know much about the function to optimize and this information can be used to create adapted crossover [DAN96] and mutation operators, an other advantage of GAs over other optimization technics.

Genetic algorithms are very efficient for solving global combinatorial optimization problems but are not very efficient for solving local searches with a good precision. Consequently, in the last generation of the genetic algorithm, a local optimization method is used to improve the best solution of each chromosome class defined above: a simple hill-climbing algorithm is applied to the best chromosome at the end of the GA run.

4 Results

We present here examples of resolution that illustrate[5] the performance of the algorithm. These examples were computed on a Pentium III 933. In the following, the time window for prediction is fixed at 12 minutes ($T_w = 12$ mn) and an optimization is computed every 3 minutes ($\delta = 3$ mn).

No uncertainty is considered in the following examples. Implementation of the uncertainty modeling is still in progress for the moment. The final version of the paper will include uncertainty parameters.

4.1 Example of Two-Aircraft Conflict

In this first application (see figure 11), at 15:06:00 UT a conflict is detected between two aircraft numbered 6744 and 6648. Aircraft 6744 is descending from Flight Level 350 to its destination and aircraft 6648 is flying at FL 230. The conflict appears near the beacon. To solve the conflict, aircraft 6648 is turned left (30 degrees) on a point created 14 nm before the beacon. A second turning point is created at a distance of 9 nm from the first one. After the second turning point aircraft is directed to the next beacon of its flight plan.

4.2 Complex conflict involving 4 aircraft

In this example, at 05:45:00 UT, 3 aircraft are cruising at FL 370 and one is cruising at FL 350. Aircraft 2298 is first in conflict with aircraft 1057, then in conflict with aircraft 82. It cannot be moved vertically because of aircraft 238 cruising at FL 350. Three horizontal maneuvers (figure12) are given to aircraft 2298, 82 and 238.

[5]The label gives the number of the aircraft, its heading, its flight level and its horizontal speed

4.3 A complete test

First simulations have been done with the new modeling without uncertainties. Simulations with uncertainties will be available within six month. A complete experiment done with unregulated flight plans of the 21th of May 1999 is described here. It involves 7540 aircraft over France. We only detect and solve conflicts above 10000 feet, as we are only interested in En Route conflicts. Aircraft entering Paris TMA control area are sequenced on the TMA entry points, but no control is done inside the TMA. Slots are given to aircraft in order to prevent simultaneous take off and landing.

When running this one day test with a very basic conflict detection algorithm (only actual conflicts are detected, with no uncertainty on speed) and with no conflict resolution, 2758 conflicts are detected with flight plan routes and 1803 conflicts are detected with direct routes. The mean flight duration is 51 minutes with flight plan routes and 46 minutes with direct routes.

When running the complete simulation with detection and resolution, fixing $\delta = 3$ minutes and $T_w = 12$ minutes, the P2 process finds 8025 clusters of different size (see figure 1. There is no unsolved cluster and consequently no conflict remains.

Cluster size	Number
2	6325
3	1158
4	331
5	112
6	52
7	19
8	15
9	8
10	1
11	3
12	0
13	1

Table 1: Cluster size

Only 1970 aircraft are given 2551 maneuvers which represents 1.3 maneuver per aircraft. The mean duration of a maneuver is $2mn14s$. Details on maneuvers are given in table 2.

The mean maneuver duration per maneuvered aircraft is $45s$ which represents 1.47% of the flight duration. Maneuvered aircraft are delayed of $19s$ on average. The global mean delay is $4s$.

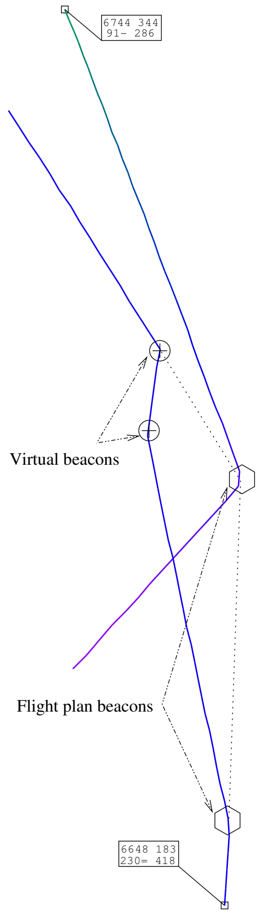

Figure 11: Conflict resolution at time 15:06:00 UT

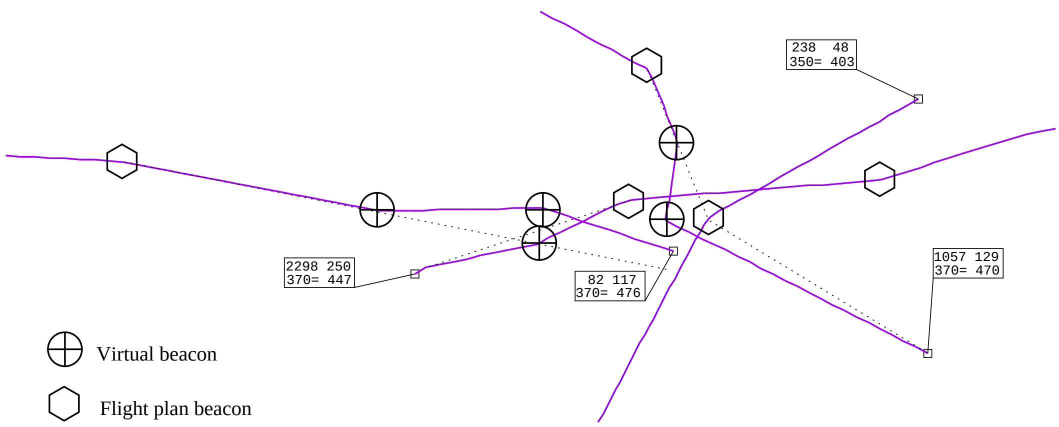

Figure 12: 05:45 UT - 4 aircraft conflict before resolution

type	number	mean duration	max duration
vert	575	2mn 38s	22mn 15s
10^0	199	2mn 16s	14mn
20^0	555	2mn 31s	21mn 15s
30^0	1222	1mn 55s	13mn 15s

Table 2: Maneuvers repartition.

With direct routes, the number of maneuvers falls to 1667 and the number of maneuvered aircraft is 1373. The mean maneuver duration per maneuvered aircraft is $18s$ which represents 0.67% of the flight duration. Maneuvered aircraft are delayed of $12s$ on average. The global mean delay is $2s$.

4.4 Further work:

Within the next six month, results with the uncertainty modeling will be available and added in the article.

The next step will include in the modeling the division of airspace into sectors. Clusters will have to be solved inside sectors and a coordination process should be defined.

The modeling should also take into account military zones.

5 Conclusion

The conflict solver introduced in this paper is a step toward simulation of en route control.

The goal of this work was to show that previous work done on automatic traffic resolution could help designing a controlled traffic simulator that could lead to the development of future ATC systems.

The simulator remains small in size (5000 lines of code in Caml language).

References

[AGS93] J. M. Alliot, Hervé Gruber, and Marc Schoenauer. Using genetic algorithms for solving ATC conflicts. In *Proceedings of the Ninth Conference on Artificial Intelligence Application*. IEEE, 1993.

[DA97] N. Durand and J.M. Alliot. Optimal resolution of en route conflict. In *Proceeding of the 1st Air Traffic Management R&D Seminar*, 1997.

[DAM93] Patrick Dujardin, J. M. Alliot, and Paul-Henri Mourlon. Different paths to automation. In *IFAC'93*, 1993.

[DAN96] Nicolas Durand, J. M. Alliot, and Joseph Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing, Philadelphia*. ACM, 1996.

[DASF94] Daniel Delahaye, J. M. Alliot, Marc Schoenauer, and Jean-Loup Farges. Genetic algorithms for partitioning airspace. In *Proceedings of the Tenth Conference on Artificial Intelligence Application*. IEEE, 1994.

[Dur96] Nicolas Durand. *Optimisation de Trajectoires pour la Résolution de Conflits en Route.* PhD thesis, ENSEEIHT, Institut National Polytechnique de Toulouse, 1996.

[EFF99] J.-H. Oh E. Frazzoli, Z.-H. Mao and E. Féron. Resolution of conflicts involving many aircraft via semidefinite programming. Technical report, MIT, March 1999.

[FMT93] Xavier Fron, Bernard Maudry, and Jean-Claude Tumelin. Arc 2000 : Automatic radar control. Technical report, Eurocontrol, 1993.

[Gol89] David Goldberg. *Genetic Algorithms*. Addison Wesley, 1989. ISBN: 0-201-15767-5.

[Hol75] J.H Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan press, 1975.

[JHOF98] J.M. Shewchun J.-H. Oh and E. Féron. Design and analysis of conflict resolution algorithms via positive semidefinite programming. Technical report, MIT, 1998.

[JHS99] Maria Prandini Jianghai Hu, John Lygeros and Shankar Sastry. Aircraft conflict prediction and resolution using brownian motion. In *Proceeding of the 38st Conference on Decision and Control, Phoenix, Arizona*, 1999.

[K+89] Fred Krella et al. Arc 2000 scenario (version 4.3). Technical report, Eurocontrol, April 1989.

[NFC+83] W.P. Niedringhaus, I. Frolow, J.C. Corbin, A.H. Gisch, N.J. Taber, and F.H. Leiber. Automated En Route Air Traffic Control Algorithmic Specifications: Flight Plan Conflict Probe. Technical report, FAA, 1983. DOT/FAA/ES-83/6.

[Nie89a] W.P. Niedringhaus. Automated planning function for AERA3: Manoeuver Option Manager. Technical report, FAA, 1989. DOT/FAA/DS-89/21.

[Nie89b] W.P. Niedringhaus. A mathematical formulation for planning automated aircraft separation for AERA3. Technical report, FAA, 1989. DOT/FAA/DS-89/20.

[VNDN97] Eric Hoffman Vu N. Duong and Jean-Pierre Nicolaon. Initial results of investigation into autonomous aircraft concept. In *Proceeding of the 1st Air Traffic Management R&D Seminar*, 1997.

[Zeg94] Karim Zeghal. *Vers une thèorie de la coordination d'actions. Application à la navigation aérienne.* PhD thesis, Université Paris VI, 1994.