Scheduling heuristics for on board sequential air conflict solving

Nicolas Archambault, Nicolas Durand

To cite this version:

Nicolas Archambault, Nicolas Durand. Scheduling heuristics for on board sequential air conflict solving. DASC 2004, 23rd Digital Avionics Systems Conference, Oct 2004, Salt Lake City, United States. pp xxxx, 10.1109/DASC.2004.1391285 . hal-00938062

HAL Id: hal-00938062

https://hal-enac.archives-ouvertes.fr/hal-00938062

Submitted on 24 Apr 2014

SCHEDULING HEURISTICS
FOR ON-BOARD SEQUENTIAL AIR CONFLICT SOLVING

Nicolas Archambault, CENA/LOG/INPT, Toulouse, France
Nicolas Durand, CENA/LOG, Toulouse, France

Abstract

The resolution of conflicts between n aircraft is highly combinational and cannot be optimally solved using classical mathematical optimization techniques. Using a priority order to solve a n-aircraft conflict is much easier but the solution is not optimal. FACES (Free-Flight Autonomous Coordinated En route Solver) is a model for coordinated on-board sequential conflict solving. In this project, conflict-free trajectories are obtained by applying elementary maneuvers to each aircraft, sequentially, according to some priority order. In this paper, a comparative study on a set of proposed priority heuristics to provide a suitable priority order is presented, and aggregated heuristics are compared according to some criteria. The conflict solver FACES using heuristics is tested in en-route upper airspace within an air traffic simulator using real traffic data.

Introduction

The Air Traffic Management System aims at allowing aircraft to fulfill their flight preferences (as regards timetables, speed, flight level) with security constraints to be satisfied. In particular, Air Traffic Control has to ensure separation between aircraft, avoiding “conflict” situations. Reliable tools for conflict detection and resolution are at stake to increase control capacity and and future airspace capacity.

Conflict solving

The conflict resolution consists in avoidance maneuvers applied by the concerned aircraft. These maneuvers can be heading angle changes (i.e. horizontal deviation), velocity changes, or vertical maneuvers, such as flight level changes for stable aircraft.

Various algorithmic approaches have been investigated in this context, using different categories of methods: optimization with genetic algorithms [1] [2], optimal control, multi-agent systems [3] [4], semidefinite programming [5], model with repulsive forces [6], mixed integer programming [7].

Objective

This study aims at optimizing a sequential, *1 versus n*, resolution method by improving the resolution priority order and extending the available maneuvers [9].

In order to study the influence of the scheduling on the resolution efficiency, we conduct an evaluation of this method with CATS [10], a traffic conflict simulation engine based on a discrete time slice execution model, using actual flight plan data.

Context

As in the FreeR project [11] [12] and “Free-Flight” context, we consider partially delegated control to aircraft. In specific airspaces, aircraft could ensure the respect of separation minima and optimize their trajectories. Aircraft are allowed to follow direct routes between their origin and destination. They are then to realize a conflict-free trajectory by solving the potential conflicts in a tactical short-term anticipation window (a few minutes).

Principle

The principle of the method is a sequential method based on a distributed model. In a sequence, each aircraft concerned computes a conflict-free trajectory for itself.

We assume that aircraft can detect each other within a certain range and send to each other a trajectory prediction and other information needed to determine a priority order (See part “Priority order”).

The aircraft with highest priority computes for itself an optimal trajectory without any constraint and broadcasts it to the other aircraft. Then, in turn, following the priority order, each aircraft computes such a trajectory for itself and broadcasts this information. Yet for these optimizations, the trajectories of preceding aircraft are taken into account as constraint: the new trajectories must avoid the already fixed ones to prevent any conflict situation.

The efficiency criterion for conflict-free trajectory is related to the lengthening of the trajectory. For instance, regarding the heading change maneuvers, the shorter the maneuver is, and the smaller the angle is, the smaller is the increase in fuel consumption and the smaller the delay.

Definitions

We define an anticipation window F_a (for instance 5 minutes) and a refresh duration Δ. The conflict detection and resolution phase is done every Δ minutes and concerns the F_a next minutes. Within the refresh duration Δ, the aircraft have to detect conflicts, exchange information and compute conflict-free trajectories.

Two aircraft are considered in potential conflict iff there exists an instant within the anticipation window when then simultaneously break the vertical separation and the horizontal separation minima. The conflict detection works in symmetrical pairs. The detection radius must be large enough to ensure that any two aircraft which would conflict within the next F_a minutes are within the radius at detection time and so labeled as conflicting.

In order to evaluate the complexity of the resolution, two parameters can be considered.

- For a given aircraft, the number of constraints, which are the trajectories of the neighbor aircraft having higher priority,
- The cluster size.

The cluster is defined from the set of potential conflict pairs as the transitive closure of the conflict relationship, as defined above. It must be noted that there need not exist a moment when all the aircraft of the cluster are simultaneously in conflict situation. In fact, our resolution method relies on neighborhoods and does not require to compute the clusters. However, conflict detection on the F_a time window frequently involves potential conflict clusters of more than 10 aircraft.

Maneuvers

At last three maneuvers types can be considered:

- horizontal deviation maneuvers, i.e. heading change by a given angle. We here consider a finite set of discrete angle value: 10°, 20° or 30° to the right or to the left. At t_0, the heading is changed by the given angle, and at t_1 the maneuvers ends and it heads towards the destination (see Figure 1)
- vertical maneuvers depend on the flight phase (see Figure 2). For instance, a stable aircraft descends to the lower flight level at t_0 and starts to climb back at t_1.
- speed regulation maneuvers could be modeled quite the same way, with two times t_0 and t_1 for the beginning and end of the maneuver, and the amplitude of the maneuvers, being an increase or decrease by a given fraction of the air speed. The speed regulation maneuvers are not used in this study, but including these is considered at a further stage.

Discretizing the angle values of the heading angle change and the speed regulation is realistic as regards the execution of the maneuvers by the pilots and the visualization by the other aircraft pilots and the controllers.

The maneuvers can be described by 3 variables: t_0 and t_1, plus the amplitude of the maneuver for horizontal deviation and speed regulation maneuvers. The current maneuvers must be either horizontal or vertical, and no combination of both. An aircraft cannot start a maneuver during another, nor change its current maneuver (except by shortening or extending the maneuvers, by reconsidering the ending moment t_1 of the maneuver at the next detection/optimization iteration).

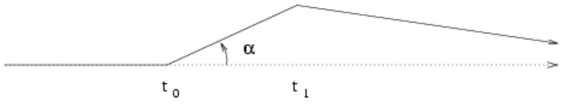

Figure 1. Horizontal maneuver

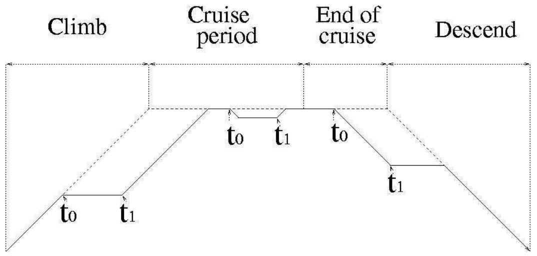

Figure 2. Vertical maneuvers

Resolution algorithm

Problem representation

The solution space is described by t_0, t_1 and a third variable corresponding to the maneuver type and amplitude.

In the resolution process as well as in the traffic simulation, time is discretized. 15 seconds is a suitable value for the time step. As the maneuver type can be represented by a variable with a finite number of values (6 values for deviation angle, 1 for vertical maneuvers, and possibly a discretized speed change from a set of value), finding a conflict-free trajectory can be modeled as a classical shortest path search in a tree.

Indeed, let us consider 4 states, such as, at every time step, an aircraft is in one of those 4 maneuvering states:

- in the state E_0, the aircraft follows its current trajectory until t_0,
- from t_0, the aircraft is in the state E_1 and is given a maneuver, from a discrete set of available maneuvers; its remains in E_1 state until t_1,
- from t_1, in the state E_2, the aircraft ends its maneuvers and follows the trajectory towards its destination,
- the aircraft is in the state E_3 iff it has reached its destination.

The resolution process for an aircraft is as follows:

While in the state E_0, at every time step, the aircraft has 8 available choices:

- stay in E_0 state,
- start a maneuver, chosen from the 6 horizontal and 1 vertical possible maneuvers (speed regulation maneuvers would only add a few choices in this model); it changes its state to E_1.

In the state E_1, the aircraft has only 2 choices, as it can:

- stay in E_1 state, continuing its maneuver,
- end its maneuver and change its state to E_2.

An aircraft in state E_2 changes its state iff it reaches its destination, to E_3 state.

In the tree representation, each tree node has a given time step and a state, including the aircraft position and speed, and its maneuvering state. The depth of a node is its time step. E_0 nodes generate 8 sons, E_1 and E_2 nodes generate 2 sons. Nodes whose state is E_3 and nodes whose time step is F_a (end of anticipation window) are tree leaves, which limits the tree depth to $F_a / \delta_{timestep}$.

The transition cost between a node and one of its son corresponds to the maneuver cost. It includes a part relative to the trajectory lengthening, as well as an arbitrary part to balance between the different type of maneuvers. Indeed, vertical maneuvers induce a small delay and an excess in fuel consumption, but they must not realistically outnumber the horizontal maneuvers, as they seem not as easy to execute. Thus this arbitrary cost is bigger for vertical maneuvers.

Also, when a node corresponds to an aircraft position where it conflicts with one of the constraint aircraft (aircraft whose trajectory is already fixed), it has an arbitrary large transition cost to prevent further exploration.

The tree-search algorithm

In this model, the optimization of a conflict-free trajectory is the search of a shortest path in the tree, as regards the cumulated cost of the father-son transitions. Standard tree search algorithm, such as A*, is suitable for the optimization. A* is a "best node first" strategy, where at each step only the best node found yet gets developed. The node comparison criterion is the sum of the computed cost so far and an heuristic estimate of the cost to a final state.

The chosen heuristic function, being the distance between the current aircraft position in the node and its destination, is underestimating the actual cost, which is a known sufficient condition to allow the algorithm to find the optimal trajectory.

Scheduling algorithm

Token allocation

Let us assume that we have a total priority order that can consistently compare any two aircraft; this priority order is discussed later. A simple (and dumb) example of such an priority order is the transponder (unique) code of the aircraft.

The algorithm used to define a global resolution order from this priority order, using only local information, is described in [9]. We recall its main features here. The following strategy is used:

1. Every aircraft broadcasts its predicted trajectory, and any other information used for the priority order, to its neighbors. Each aircraft knows whether it is conflicting with another aircraft in the F_a time window.
2. Each aircraft receives a token from every conflicting aircraft which has a higher priority in its detection zone. Aircraft that are not in conflict never receive any token.
3. Then, each conflicting aircraft with no token computes a conflict-free trajectory, taking into account the trajectories of the aircraft that have already decided their trajectory (at the first iteration step, there are none, so they will just go straight). It does not take into account aircraft that still have one or more tokens.
4. When this trajectory has been computed, the aircraft broadcasts its new trajectory; all aircraft which have received a token from this aircraft take this new trajectory into account, and dismiss the token they received from this aircraft.
5. Steps 3 and 4 are repeated until no token remains.

Provability

We recall the following properties of the algorithm, as proved in [14].

The allocation-resolution method described above cannot lead to situations where all aircraft would have at least one token, or to situations where two aircraft within sight of each other without any token would have to solve simultaneously. This is guaranteed by the use of a total priority order on aircraft. At each step, an aircraft with no token cannot have any other conflicting aircraft (which has not already solved) with no token in its detection area. Indeed, in such a case, one of these two aircraft would have given a token to the other. At each step, among the conflicting aircraft that have not already solved, there is one that has the highest priority. This aircraft cannot have any token. It can solve and get back its tokens.

Priority order

The principle of prioritizing aircraft in avoidance situations is already used in the Visual Flight Rules (VFR). Extending these rules in the "Free-Flight" Context is discussed in [15]. In [16], scheduling of aircraft for conflict resolution is compared to the classical task scheduling problem.

Aircraft scheduling is a crucial factor for the global resolution efficiency, considering all concerned aircraft. Indeed, depending on the priority used, the algorithm can bring an efficient solution, an inefficient one, or no suitable solution (that is, some conflicts remain).

We present here a method for obtaining an efficient priority order (characterized as a priority order leading to an efficient solution). This method relies on heuristic comparison criteria. These criteria should give high priority to aircraft whose resolution problem is potentially difficult, so that they decide their trajectory sooner. Indeed, late decisions are hindered by the previously decided trajectories which are constraints.

Several heuristics can be considered in order to estimate a priori the potential resolution difficulty for an aircraft: number of direct conflicts in which it is implicated, remaining time to first conflict, severity of the conflicts, aircraft speed and intrinsic maneuverability... All these characteristics are, or can be mapped to, numerical values, and a priority order can easily be created, following any one of these.

Notations

We denote as follows the independent heuristic criteria. Of two compared aircraft, the one that gets the higher priority is the one:

- *F*: with the longer time to first conflict
- *M*: with the more already scheduled maneuvers
- *N*: involved in the more conflicts
- *V*: involved in conflicts with the higher severity (average on conflicts)
- *W*: involved in conflicts with the higher severity (sum on conflicts)
- *X*: whose first conflict is the more severe
- *G*: with the higher ground speed
- *E*: with the higher vertical speed (absolute value)
- *I*: with higher ID number (such as transponder number)

The value of these criteria for each aircraft can be computed at the beginning of the detection step. The conflicts considered for some of these criteria are the predicted potential conflicts within the anticipation window.

All these are denoted in upper case; in this case the priority is higher when the value of the considered criterion is the higher. When the criterion is denoted in lower case (*f*, *m*, *n* ...), the priority is higher when the value of the considered criterion is the lower. Thus, according to criterion *f* for instance, the aircraft with higher priority is the one with the shorter time to first conflict (opposite of *F*).

Any of the defined criteria and their opposites can define a priority order; many other could be found, but it seems that in each case, either the one listed or its opposite can be relevant. With the exception of *I*, which is arbitrary and should not be relevant and so is a comparison base.

Tests

In order to estimate the relevance of these criteria, all were tested, and opposite were compared in pairs: *F* with *f*, *M* with *m*, etc...

The test is a simulation of the conflict resolution on a day of heavily loaded traffic in France, applying the conflict resolution algorithm in the upper airspace. The priority order used to define the scheduling in the sequential solving is the one tested. A test for an entire day generally runs in less than 2h30 on a Pentium IV 3 GHz, with no optimization as regards parallelization. Indeed at each resolution step, many independent aircraft can solve at the same time (being far from each other, they do not interfere).

Different priority criteria produce different resolution scheduling and thus different solution, and so as the future trajectories of aircraft along the rest of their flights. Therefore the heuristic criteria are not compared on the same set of conflicts. This is to notice for its side effects: for instance, the total number of comparisons made depends on the comparison criterion.

When two aircraft in a pair cannot be compared according to the criterion, the arbitrary criterion *i* is used (all aircraft have distinct IDs) in order to provide a total, reproducible order.

The results follow. Table 1 shows, for each criterion, the number of comparisons made, the fraction of comparisons where the criterion was discriminating, and the number of remaining conflicts. A criterion can be considered efficient if it performs better than its opposite. Opposite pairs producing similar results lead to dismiss the criterion, however this may ot be the case if the criterion was not very discriminating. For instance, *M* (resp. *m*) only concerns 0.14% (resp. 0.12%) of the comparisons, thus the difference in the global result, however slight, is noticeable, in favor of *M*. The similar results of the arbitrary criteria *I/i* are expected. The 3 criteria pairs *V/v, W/w, X/x*, based on the severity of the conflicts, show similar results (comparing a pair to another). *F/f* (time to first conflict) is moderately discriminating, and *f* performs a lot better than its opposite and is the best candidate. *G/g* is very discriminating and the low difference in the result tends to dismiss the criterion.

Table 1. Elementary criteria - Results

Criteria	Comparisons	Discriminated	Conflicts
e	12420	0,26	137
E	11918	0,22	111
f	10179	0,52	41
F	12783	0,57	163
g	11747	0,84	119
G	12287	0,84	132
i	11901	1,00	118
I	12135	1,00	127
m	12144	0,14	129
M	11322	0,12	109
n	11934	0,17	134
N	12061	0,14	109
v	11822	0,62	112
V	11077	0,61	91
w	11916	0,62	127
W	11418	0,62	94
x	11824	0,58	115
X	10973	0,57	92

Table 2. Elementary criteria - Maneuvers

	Conflicts	Number	Average	Longest
e	137	671	189	2070
E	111	708	173	960
f	41	774	170	915
F	163	642	188	2070
g	119	700	187	2820
G	132	657	180	1605
i	118	686	181	1215
I	127	674	199	3315
m	129	684	179	1215
M	109	744	164	960
n	134	655	184	1215
N	109	706	177	1995
v	112	699	174	2070
V	91	712	181	960
w	127	662	180	2070
W	94	732	176	915
x	115	693	176	2070
X	92	707	182	915

The table 2 recalls the results of the criteria (remaining conflicts) and also analyzes the maneuvers used to solve the conflicts : number of maneuvers given, average duration of maneuvers and duration of the longest maneuver. The average length of maneuvers varies little among criteria. Exceptionnally long maneuvers are rare. Still *f* is noted to perform well, as the longest maneuver is the shortest among the criteria.

Following these first tests, it seems that no criterion alone is both discriminating and relevant enough, leading the idea of combining the criteria.

Multi-criteria aggregation

This case of multi-criteria aggregation is not classical. In the classical meaning, multi-criteria aggregation aims at balancing between different optimality criteria, whereas in the present case we are not dealing with optimality criteria as such. We do not know whether the order given by one of our criterion is good in the general case. It may provide a suitable scheduling in some conflicts and not in some others.

Many multi-criteria aggregation methods needs both normalization and some form of balance between criteria. We use here, for preliminary experiments, a simple aggregation method which does not require normalization. In the multi-dimensional space, where each criterion is used as a dimension, the priority order is defined as the *pseudo-lexicographic* order. We first define an order over the criteria/axes, and the aircraft are successively compared along each axes until they are discriminated. The meta-criterion is defined as the sequence in which elementary criteria are used. For instance, the priority order denoted by *fMNi*, used to compare aircraft A and B, gives A higher priority than B:

- if its first conflict happens sooner (*f*)
- when both have the same first conflict time: if A has more maneuvers scheduled (*M*)
- when both have as many scheduled maneuvers: if A is involved in more conflicts (*N*)
- when both are involved in as many conflicts: if the ID number of A is lower (*i*).

This method still needs a form of balance between the criteria. In the previous example, *f* is more important as it is considered first and so intervenes in every comparison, whereas the other elementary criteria are only second choices.

Results

Many heuristic meta-criteria were tested, of which we select only a few highlights. Table 3 is similar to Table 1, except that it shows the fraction of comparisons where each elementary criterion was discriminating. For instance, for *fMEX*, *f* was discriminating in 52% cases, *M* in 20% (of the remaining 48%), E in 22%, *X* in 9% (the remaining cases are dealt with arbitrary order *i* on aircraft IDs).

The few results highlighted here are representative examples among the experiments.

Table 3. Results for aggregated criteria

Criteria	Discriminated	Conflicts
fMEX	0.52 0.20 0.22 0.09	29
fMXE	0.52 0.20 0.09 0.22	29
MEX	0.14 0.24 0.58	95
MXE	0.15 0.58 0.23	90
fMN	0.52 0.18 0.08	31
MNf	0.15 0.12 0.51	42
MN	0.12 0.13	104
fv	0.52 0.13	38
fV	0.52 0.13	42
fMWE	0.52 0.18 0.14 0.21	31
fWEM	0.52 0.13 0.19 0.21	32
fMEV	0.52 0.20 0.21 0.14	32
fXEM	0.52 0.08 0.19 0.22	32
fGX	0.53 0.71 0.06	33
fXG	0.53 0.08 0.70	34
fVgM	0.51 0.15 0.70 0.32	46
fgVM	0.51 0.71 0.11 0.32	51

The results raise several comments :

- The efficiency varies, but combining criteria led to better results than for individual criteria (*fMEX* leaves less unresolved conflicts than f alone).
- *f* remains a favorite candidate (*fMN* better than *MNf* or *MN*).
- Postponing an elementary criterion within the meta-criterion can decrease the discrimating power in some cases (*V/v* first discrimated in 62% or 61% cases; after *f*, it is only 13%), increase it (*M/m* first is about 14%, but in fVgM is 32%) or have no significative effect (*fMWE* and *fWEM*, *G* and *fGX*).

Priority inversion

Some of the remaining conflicts are caused by a priority inversion phenomenon.

As we modeled the trajectory optimization as a step by step process, the solver can reconsider maneuvers at the next step if the environment has changed. However, aircraft priority may change between two steps (for instance, when using criterion *M*, a aircraft which is given a maneuver may gain higher priority and may decide its trajectory before the aircraft previously viewed as a constraint) and lead to unnecessary maneuvers and even to an unsolvable conflict.

In order to avoid priority inversion, the partial priorities, between maneuvering aircraft and their constraint neighbors, could be maintained at the following steps. However maintaining partial priorities is problematic, as local comparisons cannot ensure global order any more. This is illustrated in figure 3 below, where plain arrows represent the priority order.

Step 1. A has higher priority than B (A > B). C is outside the visibilaty radius and thus not compared.

Step 2. Priority inversion. C gets closer to A and B. The computed priorities are B > C > A. Maintaining local priority from previous step (dotted arrow between A and B) breaks order transitivity and prevents resolution.

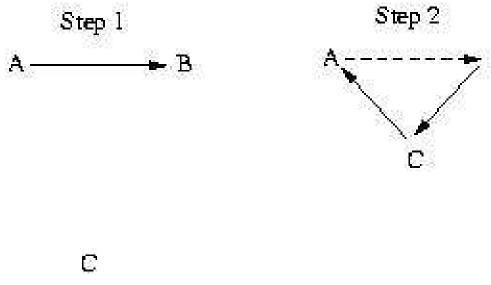

Figure 3. Priority inversion

Conclusion

This paper shows the importance of the priority order for scheduling in this case of sequential resolution, displaying high variability among the orders used.

Some of the heuristic criteria tested proved comparatively more promising for further study : time to first conflict (*f*), maneuverability (*M*), severity of conflicts, whereas others proved of lesser interest.

Further work will be done on more extensive comparative analysis on the resolution results, on the search of a solution for the priority inversion problem, and especially and the variability in the performance of heuristic criteria with respect to the available maneuvers (with the use of speed regulation maneuvers).

References

[1] Alliot, Jean-Marc, Hervé Gruber, Marc Schoenauer, 1993, Using genetic algorithms for solving ATC conflicts, 9th IEEE Conference on Artificial Intelligence Application.

[2] Durand, Nicolas, 1996, Optimisation de trajectoires pour la résolution de conflits en-route, PhD dissertation, Toulouse, INPT.

[3] Tomlin, Claire, George Pappas, Shankar Sastry, 1996, Conflict resolution for Air Traffic Management: a case study in multi-agent hybrid systems, University of California at Berkeley.

[4] Goodchild, Colin, Miguel A. Vilaplana, Stefano Elefante, 2000, Cooperative optimal airborne separation assurance in Free-Flight airspace, Napoli, 3rd Usa/Europe ATM Seminar.

[5] Frazzoli, Emilio, Zhi-Hong Mao, JH Oh, Feron Eric, 1999, Resolution of conflicts involving many aircraft via semidefinite programming, MIT.

[6] Zeghal Karim, 1998, A review of different approaches based on force fields for airborne conflict resolution, Boston, AIAA Guidance, Navigation, and Control Conference.

[7] Pallottino, Lucia, Eric Feron, Antonio Bicchi, March 2002, Conflict resolution problems for air traffic management systems solved with mixed integer programming, IEEE Transactions on Intelligent Transportation Systems, vol 3, pp 3-11.

[8] Kosecka Jana, Claire Tomlin, George Pappas, Shankar Sastry, 1997, Generation of conflict resolution maneuvers for Air Traffic Management, International Conference on Robotics and Intelligent Systems.

[9] Granger, Geraud, Nicolas Durand, Jean-Marc Alliot, 2001, Token allocation strategy for Free-Flight conflict solving, Seattle, IJCAI.

[10] Alliot Jean-Marc, Nicolas Durand, J. Bosc, L. Maugis, 1997, Cats: a complete air traffic simulator, Irvine, 16th AIAA/IEEE Digital Avionics Systems Conference.

[11] Duong Vu, Eric Hoffman, Jean-Pierre Nicolaon, 1997, Initial results of investigation into autonomous aircraft concept (FREER-1), Saclay, 1st USA/Europe ATM Seminar.

[12] Duong Vu, Eric Hoffman, 1997, Conflict resolution advisory in autonomous aircraft operations, Irvine, 16th AIAA/IEEE Digital Avionics Systems Conference.

[13] Durand, Nicolas, Jean-Marc Alliot, 1997, Optimal resolution of en-route conflicts, Saclay, 1st USA/Europe ATM Seminar.

[14] Granger, Geraud, 2002, Détection et résolution de conflits aériens: modélisations et analyse, Ph. dissertation, Toulouse, INPT.

[15] Duong, Vu, Eric Hoffman, Jean-Pierre Nicolaon, Laurent Floc'hic, Andre Bossu, 1996, Extended Flight Rules (EFR) to apply to the resolution of encounters in autonomous airborne separation, Eurocontrol Experimental Center.

[16] Carlier Jacques, Vu Duong, Dritan Nace, Huy-Hoan Nguyen, 2002, Using disjunctive scheduling for optimally solving en-route conflicts, Irvine, AIAA/IEEE 21st Digital Avionics Systems Conference.

Email Address

archamba@recherche.enac.fr