Optimal Resolution of En Route Conflicts

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: durand@cena.dgac.fr - alliot@dgac.fr

Abstract

Automatic Control has been a subject of studies for the last twenty years. It involves many difficult problems that have to be solved: conflict detection, modelling of uncertainties on trajectories, clustering of 1-to-1 conflict to find unconnected n-aircraft problems, 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 an automatic conflict solver and its implementation in an Air Traffic simulator, with statistical results on real traffic over France. This solver, which takes into account speed uncertainties and allows aircraft to fly on direct routes, solves every conflict on a loaded day, and gives each aircraft its requested flight level and departure time.

Introduction

As traffic keeps increasing, En Route capacity, especially in Europe, becomes a serious problem. Aircraft conflict resolution, and resolution monitoring, are still done manually by controllers. Solutions to conflicts are empirical and, whereas aircraft are highly automated and optimized systems, tools provided for Air Traffic Control (ATC) remain very basic. When comparing the current capacity and the standard separation to the size of controlled space, the conclusion is easy to draw: while ATC is overloaded, the sky is empty.

The need for an automatic problem solver is also a serious concern when addressing the issues of free flight. It is still very unclear how conflicts will be solved in free flight airspace. Human controllers frequently rely on standard routes and traffic organization for avoiding conflicts; they quickly become overloaded when controlling aircraft flying on direct routes. Free flight traffic, the aim of which is to permit each aircraft to fly its preferred trajectory, results in an unorganized structure, probably requiring automated, computer based, solvers. The Airborne Collision Avoidance System (ACAS) is certainly not a solution to the problem: it has only a limited view of the traffic, and moreover, should only be looked upon as a security system to prevent aircraft collision.

The first part of the paper presents the state of the art for problem solvers and discusses the constraints hypothesis and goals chosen. Modelling is introduced in the second part. Part three details the conflict solver. Part four presents examples of resolution on real traffic and statistical results.

1 Automatic conflict resolution

1.1 State of the art

Conflict resolution is a very complex mathematical problem involving trajectory optimization and constraint handling. This problem has many facets: conflict detection, clustering, conflict resolution and optimality of the solution regarding different criteria. There have been many attempts to reach these objectives.

- AERA 3 [NFC+83, Nie89b, Nie89a] considered optimum results in the "Gentle-Strict" function for a two aircraft conflict, but the "Maneuver Option Manager" only searches for acceptable solutions and does not focus on the optimum. Moreover, the MOM behavior is poorly described and the way it handles n-aircraft conflict to divide them into problems that the GS algorithm can solve is unclear.

- Karim Zeghal [Zeg94], with reactive techniques for avoidance, gives a solution to the problem of automation which is robust to disturbance, but completely

*The LOG is a common laboratory of the Centre d'Etudes de la Navigation Aérienne and the Ecole Nationale de l'Aviation Civile disregards optimization. Furthermore, the modelling adopted implies a complete automation of both on board and ground systems and requires speed regulation which cannot be handled by human pilots and would probably be very difficult to apply to aircraft engines without damaging them.

- ARC-2000 [K+89, FMT93] optimizes aircraft trajectories using 4 dimensional cones and priority rules between aircraft. Optimum is not reached, and the system relies on the availability of FMS-4D for all aircraft, with no uncertainty on speeds[1].

- A first approach to conflict resolution by stochastic optimization algorithms (genetic algorithms)[2] was done by Alliot and Gruber [AGS93]; more advanced results were presented in [DASF94b, DAN96]. Another approach, also using genetic algorithms, was tried by Kemenade, Hendriks, Hesseling and Kok [vKHHK95].

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.
- 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.

[1] It must be noted that only the ARC-2000 system has been tested on "almost" real traffic.

[2] It must be noted that genetic algorithms were also applied to airspace sectorization with promising results [DASF94a].

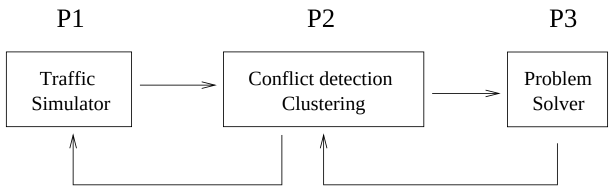

Figure 1: General architecture

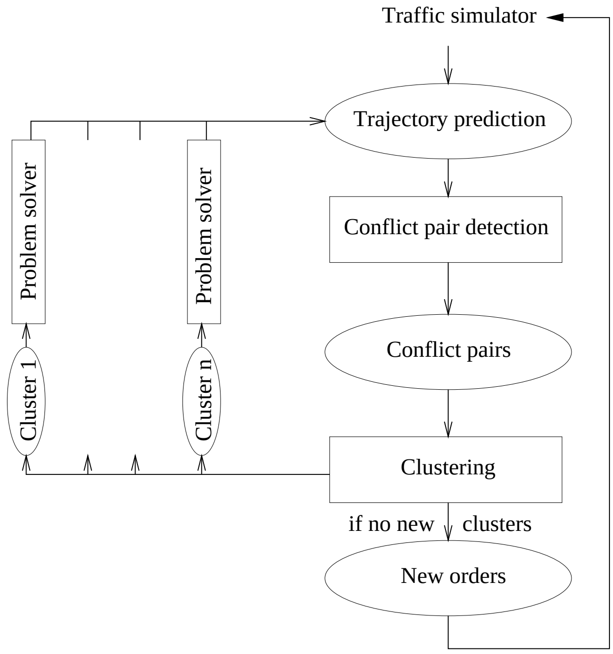

Figure 2: Detailed architecture of the prototype

2 Modelling

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 modelling 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. Performance data comes from the French operational CAUTRA system. There are currently around 250 different aircraft models available.

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 modelling for conflict detection and resolution is discussed later in the article.

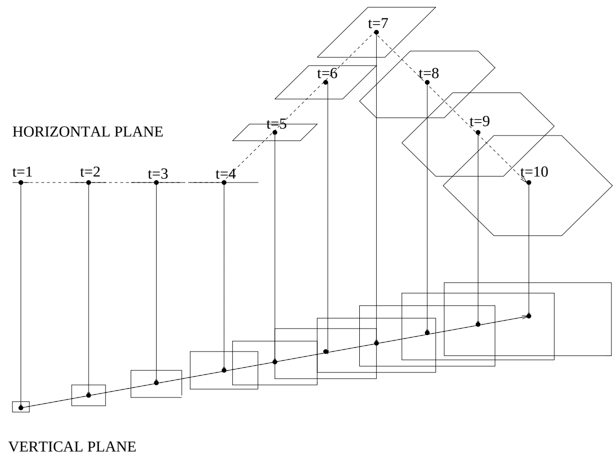

Figure 3: Modelling of speed uncertainties.

Aircraft follow either classical routes (from way-point to way-point) or direct route (from the departure, or entry point in the French airspace to their destination or leaving 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 timestep that can be chosen at the start of the simulation. The timestep 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 timestep. For most of our simulation, we use a $15s$ timestep.

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[3]. 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. When changing direction ($t = 4$),

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

the segment becomes a parallelogram that increases in the speed direction. When changing a second time direction ($t = 7$), the parallelogram becomes an hexagon that increases in the new speed direction, and so on. To check the standard separation at time t, we compute the distance between the two polygons modelling the aircraft positions and compare it to the standard separation at each timestep of the simulation.

In the vertical plane, we use a cylindrical modelling (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 gspeed is 175 kts. If we suppose that gspeed uncertainty is 5% and vspeed uncertainty 20%, maximal and minimal climb rate are $1800 \times 1.2 = 2160$fts and $1800 \times 0.8 = 1440$ fts/mn and gspeeds 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 gspeed of 197 kts, so max gspeed is $197 \times 1.2 = 237$ kts. At 360 fts, standard gspeed is 189 kts, with a minimal gspeed of 151 kts. So, the size of the convex 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 convexes modelling 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.

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 Theoretical results

The two aircraft conflict problem has been widely studied theoretically using Optimal Command Theory.

Optimal Command Theory with State Constraints ([E.K82]), lead to the following conclusions exposed by Durand, Alech, Alliot and Schœnauer in [DAAS94]. For a conflict resolution involving two aircraft: at the optimum, as long as the standard separation constraint is not saturated, aircraft fly in straight lines. When saturating, aircraft start turning, and as soon as the separation constraint is over, aircraft fly straight again. This result can easily be extended to the case of n aircraft, with $n \geq 2$. When moving only one aircraft, it can be proved (see [Dur96]) that trajectories are regular (they do not include any discontinuous point).

Numerical resolutions show that the length of the conflict free trajectory increases when:

- the angle of incidence between the two aircraft decreases.
- the speed ratio gets close to 1.
- aircraft are closer to the conflict point.

The previous mathematical study leads naturally to simplify the conflict free trajectory (see figure 4). The turning point trajectory is very close to the optimal trajectory and much simpler to describe. It will be used in the following.

It can also be mathematically proved that if aircraft parameters (speed and heading) are constant at intervals, and if aircraft trajectories don't loop, the set of conflict free trajectories has two connected components. In one of the two sets, one of the aircraft always passes the other one on its right side, whereas in the other set, it passes it on its left side. For n aircraft, the theoretical number of conflict free trajectories sets expands to $2^{\frac{n\,(n-1)}{2}}$: if $n = 9$, there is more than 268 million possibilities (see [MDA94]).

3.2 Maneuver decision time

Because of uncertainties, a conflict that is detected early before it should occur may finally not happen. Conse-

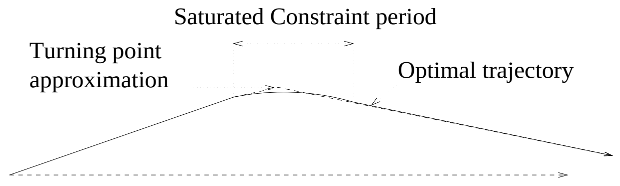

Figure 4: Turning point approximation.

quently, 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 modelling, 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

In this paper, it was decided to allow direct routes to aircraft. In a first time, only turning points were considered in the horizontal plane. After the turning point execution, aircraft were directed to their destination.

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[4].

For example, the first trajectory of figure 5, 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 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.

[4] 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.

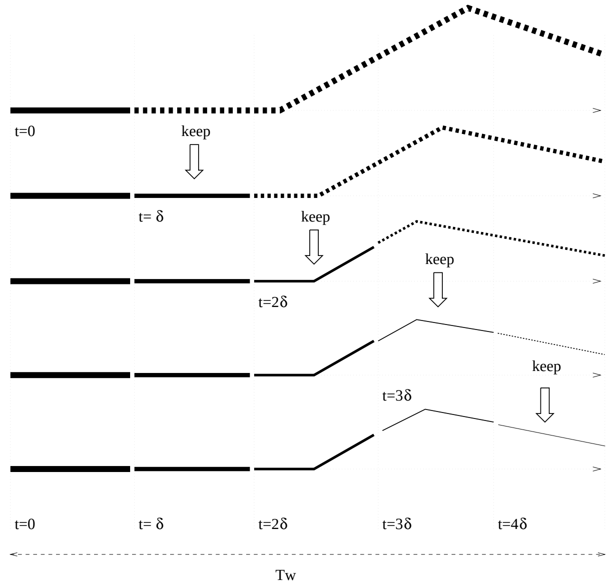

Figure 5: The model and real time optimization.

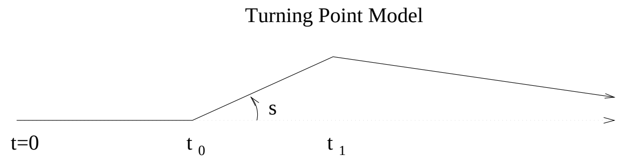

Figure 6: Horizontal maneuver modelling.

The turning point angle will be 10, 20 or 30 degrees. The previous elements lead us to choose the following model (figure 6). A maneuver will be determined by:

- the maneuver starting time t_0.
- the turning point time t_1.
- the deviation angle s.

In a second time, vertical maneuvers were introduced. Therefore, the aircraft trajectory is divided in 4 periods (figure 7):

- Climbing period. In this period, aircraft can be leveled at a lower than requested flight level during a moment to resolve a conflict. The maneuver starts at t_0. Aircraft start climbing again at t_1 and $s = 0$.
- Cruising period. When aircraft have reached their desired flight level, they may be moved to the nearest lower level to resolve a conflict. Aircraft start descending at t_0 and start climbing at t_1 ($s = 0$).

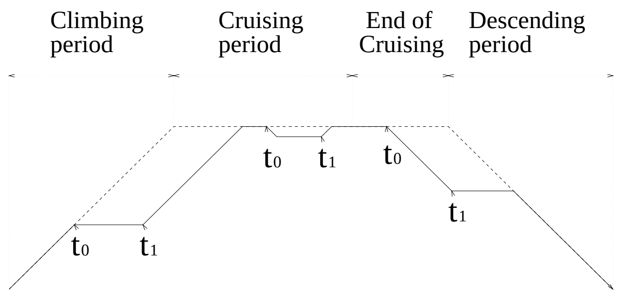

Figure 7: Vertical maneuver modelling.

- End of Cruising period. When aircraft are about 50 nautic miles from beginning their descent to destination, they may be moved to a lower level to resolve a conflict. Aircraft start descending at t_0 and are leveled at t_1 $(s = 0)$.
- Descending period. During this period no vertical maneuver is possible.

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.

3.4 Complexity of the problem

The complexity of the problem is exposed by Medioni, Durand and Alliot in [Dur96]. Let's consider a conflict between two aircraft. We can easily prove that the minimized function is convex, but the set of conflict free trajectories is not. It is not even connected. If trajectories don't loop, the set of conflict free trajectories has two connected components. For a conflict involving n aircraft there may be $2^{\frac{n\,(n-1)}{2}}$ connected components in the free trajectory space which strongly suggests that any method which requires exploring every connected component is NP[5]. It is important to note that this complexity is independent of the modelling chosen (see [Dur96]).

3.5 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[6].
- 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 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):

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

[5] A Non deterministic Polynomial (NP) problem belongs to a class of problem for which there are no polynomial-time algorithm known to solve the problem.

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

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[7] 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.

3.6 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.

3.7 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 8.

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.

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

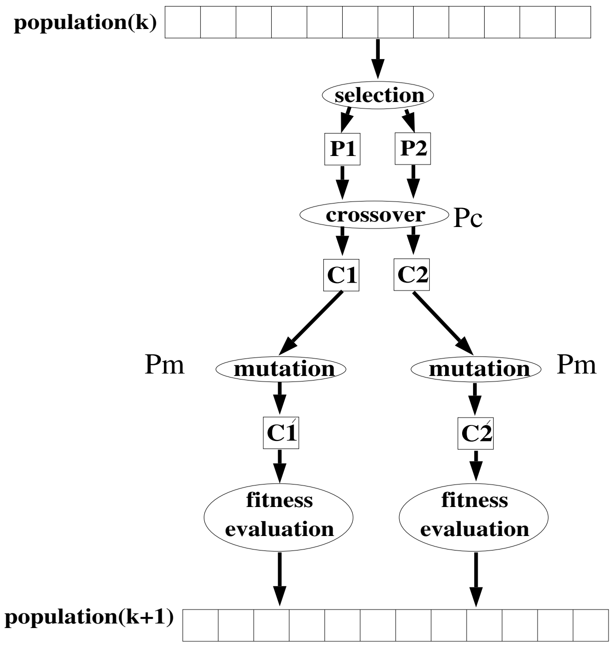

Figure 8: GA principle

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 chromosomes at the end of the GA run.

4 Results

We present here examples of resolution that illustrate[8] the performance of the algorithm. These examples were computed on a Pentium 200. 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).

4.1 Example of Two-Aircraft Conflict

In this first application, at 09:36:00 UT a conflict is detected between two aircraft numbered 2294 and 2261 flying at level 310 (see figure 9). Because of uncertainties, the horizontal predicted speed of aircraft 2294 is 475 kts plus or minus 5% (its real speed is 470 kts), whereas the horizontal predicted speed of aircraft 2261 is 445 kts plus or

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

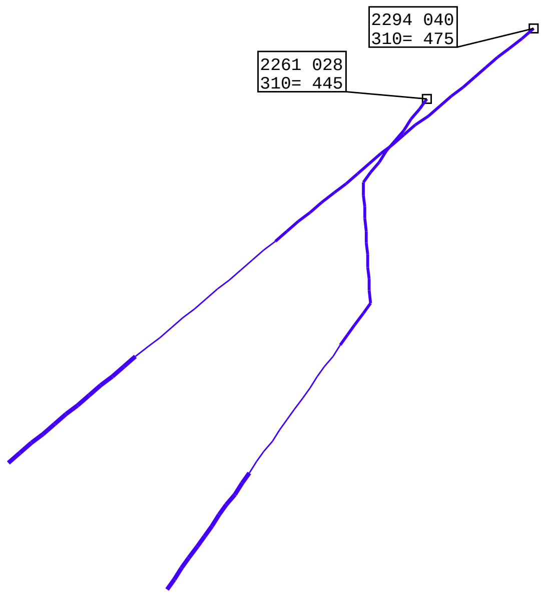

Figure 9: Conflict resolution at time 09:36:00 UT

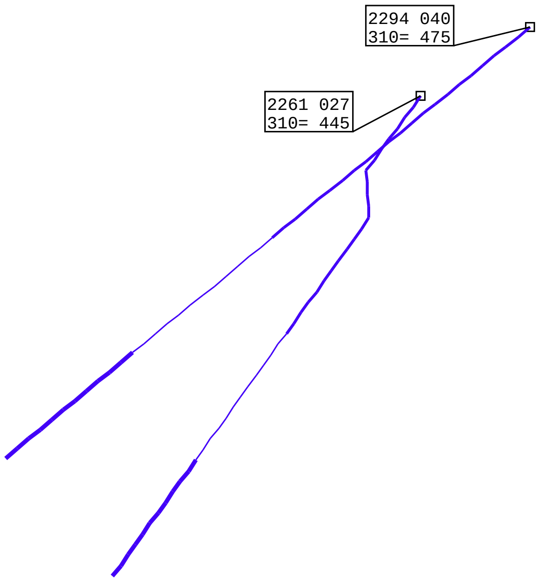

Figure 10: Conflict resolution at time 09:39:00 UT

minus 5% (its real speed is 427 kts). The solver calculates the optimal solution to solve the conflict in the horizontal plane: aircraft 2261 should be moved 30 degrees left at 09:43:00 UT during 2 minutes and 15 seconds. As only maneuver orders starting before 09:42:00 UT are definitive, no order is given to aircraft 2261.

At 09:39:00 UT (figure 10), the conflict is still detected, however, the solver suggests to move aircraft 2261 of 30 degrees left at 09:47:45 UT during 45 seconds only. As only maneuver orders starting before 09:45:00 UT are definitive, no order is given to aircraft number 2261.

At 09:42:00 UT, no conflict is detected between these two aircraft.

Because of uncertainties, the initial optimized trajectory requires a fairly large deviation from the intended path. As times goes on, aircraft are closer to the conflict point, uncertainty decreases, and the optimized trajectories give smaller deviations. Finally, at 09:42:00 UT, the conflict disappears.

4.2 Complex conflict involving 5 aircraft

In this example (figure12), at 10:33:00 UT, 5 aircraft are cruising at FL-350. 4 1-to-1 conflicts are detected (see table 1).

Only 2 aircraft are moved as follows: aircraft 2509 is first moved vertically at 10:39:15 UT to FL-340 till the end of the time window (10:45:00 UT), which resolves conflicts with aircraft 2539 and 2324. Aircraft 2485 is moved vertically at 10:41:15 UT to FL-340 till the end of the time

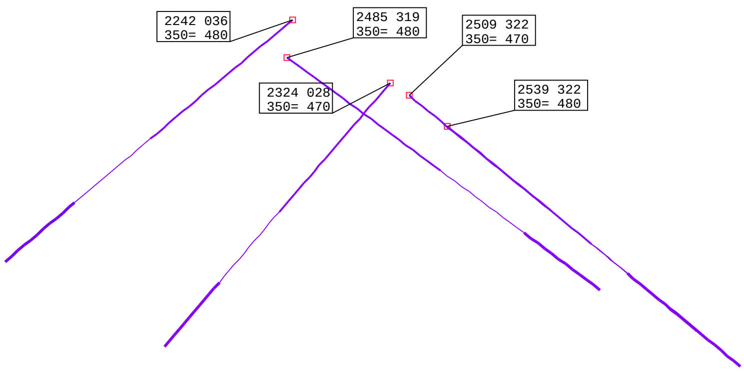

Figure 11: 10:33 UT - 5 aircraft conflict before resolution

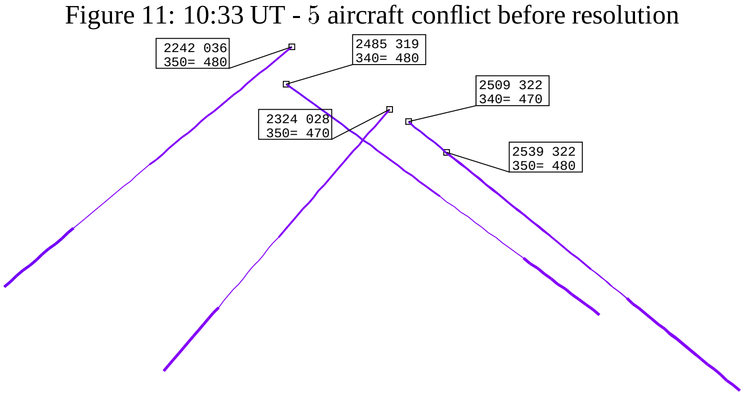

Figure 12: 10:33 UT - 5 aircraft conflict after resolution

aircraft	beginning	ending
$2509 - 2539$	$10:40:00$	$10:45:00$
$2324 - 2485$	$10:42:00$	$10:43:00$
$2324 - 2509$	$10:44:00$	$10:45:00$
$2242 - 2485$	$10:44:00$	$10:45:00$

Table 1: 10:33 UT - Conflict beginning and ending

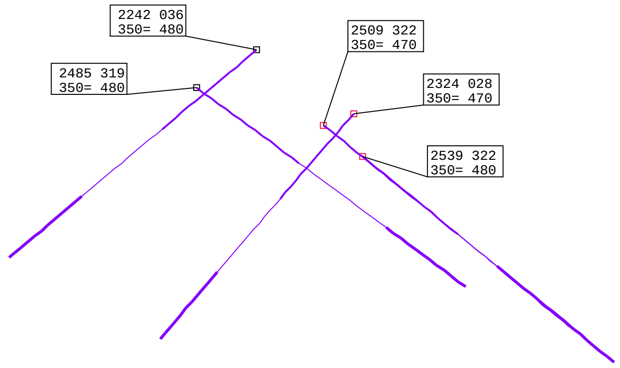

Figure 13: 10:36 UT - 5 aircraft conflict before resolution

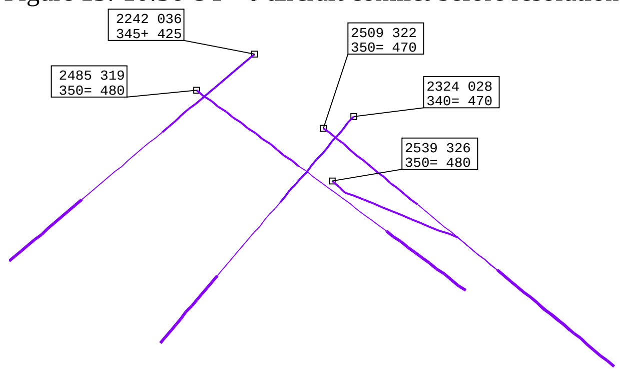

Figure 14: 10:36 UT - 5 aircraft conflict after resolution

window, which resolves conflicts with aircraft 2242 and 2324.

No aircraft is given an order at this step (no maneuver is supposed to start before 10:39:00 UT).

At 10:36:00 UT (figure12), 5 conflicts are detected (table 2).

The previous solution is not conflict free anymore because of the new 1-to-1 conflict that has appeared at time 10:45:15 UT between aircraft 2324 and 2539. The solver finds another solution. Only 3 aircraft are moved as follows: aircraft 2324 is first moved vertically at 10:41:30 UT to FL-340 during $4mn30s$, which resolves conflicts with aircraft 2485, 2509 and 2539. Aircraft 2539 is moved 20 degrees left at 10:42:00 UT during $3mn$, which resolves conflict with aircraft 2509. Finally, aircraft 2242

aircraft	beginning	ending
$2324 - 2485$	$10:42:15$	$10:43:00$
$2509 - 2539$	$10:42:30$	$10:48:00$
$2324 - 2509$	$10:44:00$	$10:46:15$
$2242 - 2485$	$10:44:15$	$10:45:00$
$2324 - 2539$	$10:45:15$	$10:46:15$

Table 2: 10:36 UT - Conflict beginning and ending

is moved vertically at 10:43:30 UT to FL-340 during $1mn30s$, which resolves the conflict with aircraft 2485.

Only aircraft 2324 will be given a maneuver order at this step because its maneuver will be definitive at the next iteration. Its maneuver ending and the other maneuvers will be reconsidered at time 10:39:00 UT.

At time 10:39:00 UT, 2 unconnected clusters (2324, 2509, 2539) and (2485, 2242) are found.

Aircraft 2324 finally ends its maneuver at 10:45:30 UT. Aircraft 2539 is moved 10^0 left at 10:43:30 UT during $7mn$. Aircraft 2485 is moved vertically to FL-340 at 10:44:00 UT during $45s$, which definitely resolves conflict with aircraft 2242.

4.3 Example with large numbers of Conflicting Aircraft

Figure 15 gives an example of a 27 aircraft cluster. It is here useless to try to understand what happens, but every conflict is resolved.

4.4 A complete test

Testing of the problem solver is still in progress, but some tests have already been completed [Cha95]. A complete experiment done with unregulated flight plans of the 21th of June 1996 is described here. It involves 6388 aircraft over France. Uncertainties on climbing rate and ground speed are respectively set to 20% and 5%, and standard separations are set to 6 nm and 1000 feet. The experiment is run under the Direct Route hypothesis (aircraft are allowed to go directly to their destination). 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.

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, 1649 conflicts are detected.

When running the complete simulation with detection and resolution, fixing $\delta = 3$ minutes and $T_w = 12$ minutes,

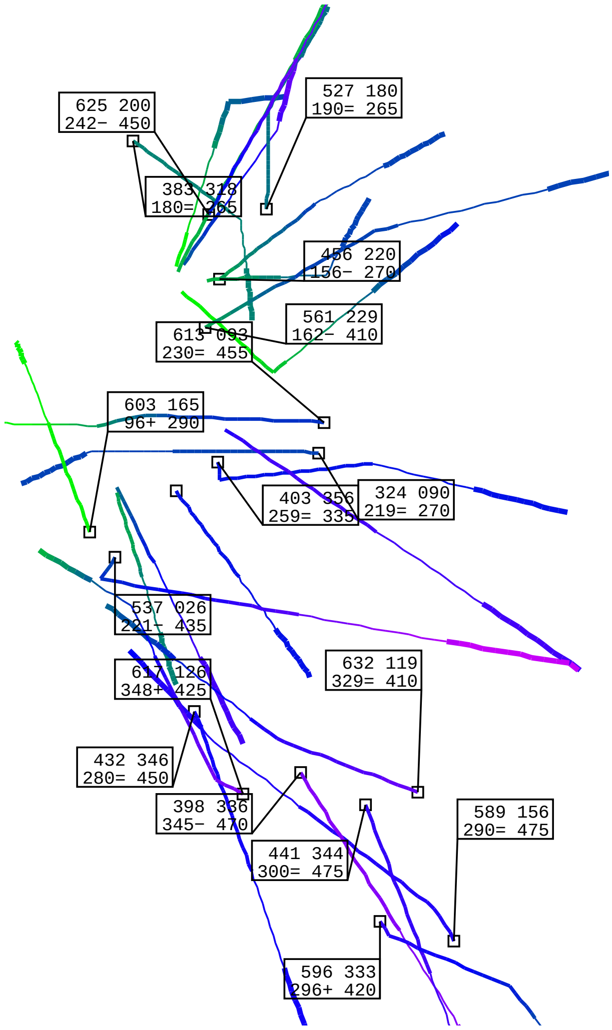

Figure 15: 27 aircraft cluster

the P2 process detects 5064 1-to-1 different conflicts (This means that a detected conflict has $\frac{1649}{5064} = 32.6\%$ chance to really occur. The problem solver resolves 9130 clusters of different sizes (table 3). There are 4155 clusters including different sets of aircraft. There is no unsolved cluster and consequently no conflict remains.

Only 1687 aircraft are given 2200 maneuvers which represents 1.3 maneuver per aircraft. The mean duration of a maneuver is $2mn23s$. Details on maneuvers are given in table 4.

1337 aircraft are delayed before taking off or entering the French airspace. For these aircraft, the mean take off or entering delay is $2mn56s$ (maximum $6mn$). The global mean take off or entering delay is $37s$.

The mean maneuver duration expectation per aircraft is $49s$ which represents 1.79% of the flight duration.

The mean flight duration is $45mn54s$ before resolution

clus size		clus size		clus size	
2	6448	10	23	18	1
3	1539	11	23	26	1
4	570	12	17	27	1
5	210	13	7	28	1
6	126	14	7	31	1
7	67	15	5	32	1
8	48	16	5		
9	28	17	1	total	9130

Table 3: Sizes of solved clusters.

type	number	mean duration	max duration
vert	1256	2mn 22s	15mn 45s
10^0	218	2mn 14s	8mn 45s
20^0	452	2mn 14s	12mn 30s
30^0	274	2mn 38s	8mn 45s

Table 4: Maneuvers repartition.

and $45mn58s$ after resolution. The mean delay caused by maneuvers is $4s$. Only 934 aircraft are delayed because of maneuvers (most of the aircraft moved in the vertical plane are not delayed). The maximum delay is $4mn$ and the mean delay (for aircraft delayed) is $29s$.

The same simulation, with the same parameters was performed without giving maneuvers in the vertical plane. 5112 different 1-to-1 conflicts are detected. The problem solver resolves 8869 clusters of different sizes (table 5). There are 4115 clusters including different sets of aircraft. There are 2 unsolved clusters involving 52 and 64 aircraft, but remaining conflicts are resolved at the next step. At last, no conflict remains.

Only 1779 aircraft are given 2344 maneuvers which represents 1.32 maneuver per aircraft. The mean duration of a maneuver is $2mn35s$. Details on maneuvers are given in table 6.

1300 aircraft are delayed before taking off or entering the French airspace. For these aircraft, the mean take off or entering delay is $2mn55s$ (maximum $6mn$). The global mean take off or entering delay is $35s$.

The mean maneuver duration expectation per aircraft is $57s$ which represents 2.07% of the flight duration.

The mean flight duration is $45mn54s$ before resolution and $46mn3s$ after resolution. The mean delay caused by maneuvers is $9s$. Some maneuvered aircraft are not delayed (a 10^0 maneuver during $1mn$ induces $1s$ of delay). The maximum delay is $13mn45s$ and the mean delay (for the 1269 aircraft delayed) is $41s$.

clus size		clus size		clus size	
2	6269	12	14	24	1
3	1440	13	4	25	1
4	548	14	4	46	1
5	244	15	7	48	2
6	133	16	5	50	1
7	79	17	3	52	1
8	46	18	3	55	1
9	26	19	2	59	1
10	17	20	1	63	1
11	12	22	1	64	1
				total	8869

Table 5: Sizes of solved clusters (horizontal maneuvers).

type	number	mean duration	max duration
10^0	518	2mn 11s	17mn 45s
20^0	934	2mn 22s	21mn 30s
30^0	892	3mn 1s	25mn 30s

Table 6: Maneuver repartition.

4.5 Limitations and improvements

The solver has different limitations. First of all, it is designed to handle En-Route control problems, with a large number of aircraft and a time window larger than 10 minutes. Even if it could perform resolution for a smaller number of aircraft and a shorter time window, we are currently investigating other algorithms, based on the A^* family and on interval programming that are probably much more fitted when considering problems linked to, for example, ASAS. The conflict detection system, that relies on a simulation of trajectories for the next T_w minutes, and thus prevents using combinatorial linear programming, could also be simplified for shorter time windows. Approximating an aircraft trajectory by linear segments is useless on 12 minutes, but could be considered for less than 5 minutes.

One of the main problems that remains to be addressed is certainly trajectory forecast. The system is highly sensitive to errors on aircraft speed. Indeed, the cluster size increases when T_w increases and when the uncertainty on speed increases. For example, with the uncertainty on speed estimation used in the above examples and $T_w = 15$ mn, the biggest cluster deals with 56 aircraft; with $T_w = 16$ mn, it reaches 71 aircraft. The solver could then quickly saturate. Trajectory forecast is definitely a serious issue for all systems doing either automatic resolution or controller assistance: no controller would accept an operational system which detects conflicts that never occur, or fails to detect conflicts that will occur. Work is in progress to improve dynamically aircraft trajectory forecast using the “standard” aircraft model and its past positions, based on neuro-mimetics technics and mathematical regressions.

To prevent clusters to become too large, another possibility is to forbid cluster merging after resolution by making new resolution with aircraft in one cluster being constraints for aircraft in the other cluster. Global optimality would be lost, but it would allow to increase the detection time window or uncertainty.

Other improvements are currently under development: introduction of time maneuver execution uncertainty, military zones, indirect routes, etc. We also plan to run many tests and statistics with different parameters (standard separation) and other traffic data, especially with projection for the future.

5 Conclusion

The conflict solver introduced in this paper is a step toward automatic resolution of en route conflicts. The goal of this work was to show that a scientific, mathematical approach along with a serious algorithmic design could build a complete system for conflict detection and resolution, that would still remain small in size (the whole system including the traffic simulator, conflict detection, clustering and problem solver is less than 4500 lines of code). Even if many improvements have to be done, the results of the simulation are good. Mean delays induced by maneuvers are very short (4 s), maximum en route delays remains also short (4 mn) and all these results were obtained with unregulated flight plans.

Trying this system on real traffic, to develop resolution tools, or for night control, would be an interesting challenge, but is more a political than a technical problem.

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.

[Cha95] Olivier Chansou. Résolution automatisée de conflits en route. Master’s thesis, Ecole Nationale de l’Aviation Civile (ENAC), 1995.

[DAAS94] Nicolas Durand, Nicolas Alech, J. M. Alliot, and Marc Schoenauer. Genetic algorithms for optimal air traffic conflict resolution. In *Proceedings of the Second Singapore Conference on Intelligent Systems*. SPICIS, 1994.

[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.

[DASF94a] Daniel Delahaye, J. M. Alliot, Marc Schoenauer, and Jean-Loup Farges. Genetic algorithms for air traffic. In *Proceedings of the Conference on Artificial Intelligence Application*. CAIA, 1994.

[DASF94b] 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.

[E.K82] E.Kreindler. Additional necessary conditions for optimal control with state-variable inequality constaints. *Journal of Optimization theory and applications*, 38(2):241–250, october 1982.

[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.

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

[MDA94] F. Medioni, Nicolas Durand, and J.M. Alliot. Algorithmes génétiques et programmation linéaire appliqués a la résolution de conflits aériens. In *Proceedings of the Journees Evolution Artificielle Francophones*. EAF, 1994.

[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.

[vKHHK95] C.H.M. van Kemenade, C.F.W. Hendriks, H.H. Hesselink, and J.N. Kok. Evolutionnary computation in air traffic control planning. In *Proceedings of the Sixth International Conference on Genetic Algorithm*. ICGA, 1995.

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