Automatic aircraft conflict resolution using Genetic Algorithms

Nicolas Durand
CENA[*]
7, av E. Belin
31055 Toulouse Cedex
durand@cena.dgac.fr

Jean-Marc Alliot
ENAC[†]
7, av E. Belin
31055 Toulouse Cedex
alliot@dgac.fr

Joseph Noailles
ENSEEIHT[‡]
3 rue Camichel
31000 Toulouse
noailles@enseeiht.fr

Keywords : Genetic Algorithms, Conflict Resolution

Abstract

In this paper, we show how genetic algorithms can be used to solve en-route aircraft conflict automatically to increase Air Traffic Control capacity in high density areas. The ATC[1] background and the model are presented. The complexity of the problem is then discussed. The author then justifies the choice of GAs. After a brief description of genetic algorithms, the author describes the improvements that were used for solving the conflict resolution problem. Several numerical applications are then given justifying the choices that were made and illustrating the interest of the research. Next steps of this work are discussed in the conclusion.

Introduction

Studies on the use of genetic algorithms for conflict resolution and air space sectoring have given promising results [2, 6, 7, 10, 19]. This paper presents new results for solving conflict[2] between aircraft for En Route ATC in real time situation.

The first part of the paper introduces the problem, its constraints and goals. The choice of the model is discussed in the second part. In the third part, we discuss the complexity of our problem and explain why classical optimization tools such as gradient methods are not adapted to solve it. Part four describes briefly genetic algorithms and the improvements that were used. Part five presents the coding and part six give several numerical applications.

1 Problem description

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and, whereas aircraft are highly automated and optimized systems, tools provided for ATC control are very basic, even out of date. If we compare 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.

However, conflict resolution is a very complex trajectory optimization problem under constraints. There have been many attempts to try to reach one of the two objectives, automation and optimization. However, most of the time these two objectives were confused (see [18, 21, 12]).

The solver we introduce has to handle the following constraints:

- Conflict free trajectories must respect both aircraft and pilot performances. Considering the evolution of ATC toward automation [8], 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[3].

- Maneuvers orders must be given with an advance notice to the pilot. When a maneuver has begun, it must

---

[1] Air Traffic Control

[2] 2 aircraft are said to be in conflict if their altitude difference is less than 1000 feet (305 meters) and the horizontal distance between them is less than 8 nautical miles (14800 meters). These two distances are respectively called vertical and horizontal standard separation.

---

[3] Aircraft ground speeds can not be forecasted precisely because of winds, and radar precision. Moreover, models to forecast aircraft ground speeds are not reliable enough. Consequently, uncertainty on speeds have to be introduced in the model.

not be called into question. A maximum of one maneuver per aircraft should be forecasted for the next twenty minutes.

- If possible, conflicts must be solved horizontally for comfort and economical reasons, especially when aircraft are leveled.

We want to achieve the following goals:

- find conflict free and optimal trajectories

- compute these trajectories in real time

- find many different optimal or nearly-optimal solutions, so that in a transition phase to automation, this tool could be used to assist a human operator

2 Modeling

In this paper we will restrict our problem to aircraft flying at constant level. It could and will be extended to $3D$ conflict resolution. A simple way to do this is to use the horizontal projection of the climbing and descending aircraft and to treat them as leveled aircraft with different speeds. Consequently, our $2D$ modeling can be extended to $3D$ immediately. Furthermore, the complexity of the problem does not increase when extended to $3D$.

2.1 Theoretical results

Durand, Alech, Alliot and Schœnauer showed in [10] that for a conflict resolution involving two aircraft : at the optimum, as long as the standard separation constraint is not saturated, aircraft fly in straight line. When saturating, aircraft start turning, and as soon as the separation constraint is freed 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 that trajectories are regular, they do not include discontinuous points. It can be proved [9] 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 getting closer to the conflict point.

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
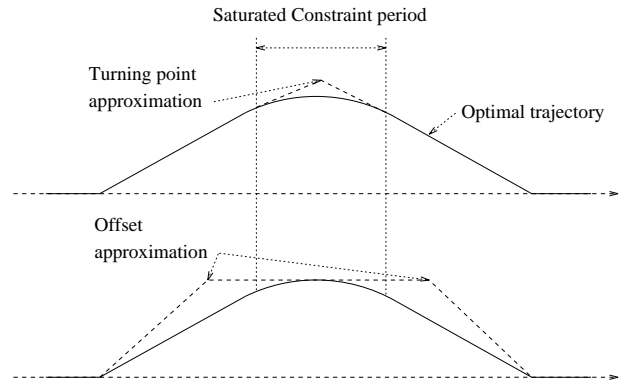

Figure 1: Turning point and offset approximation.

sets, one of the aircraft always lets the other one on its right side, whereas in the other set, it lets it on its left side.

The previous mathematical study leads naturally to approximate the conflict free trajectories by a turning point trajectory (figure 1).

It was shown [9] that the turning point approximation lengthens the optimal trajectory by less than 1% if distance between the aircraft and the conflict point is greater than two standard separations and the angle of incidence between trajectories greater than 30 degrees. It can also be proved that the offset modeling (figure 1), which moves an aircraft to put it on a parallel route, is worse, except for conflicts between overtaking aircraft. It has the advantage to linearize the separation constraints. The offset is thus very easy to compute, but separation constraints must be checked during maneuvers and the complexity of the problem remains (see [15]).

2.1.1 Speed uncertainties

In this paper, we will assume that there is an error about the aircraft future location because of speed uncertainties[4] The conflict free trajectory must be robust regarding these uncertainties. When climbing or descending, aircraft follow climbing rates and descending rates that are fixed by technical constraints and on which uncertainties are more important. To take the uncertainty problem into account, and to be able to guarantee to an aircraft a conflict free trajectory for the next 20 minutes, we will represent the aircraft by a point at the initial time. The point will become a segment of a line in the uncertainty direction, the speed direction in our case. To check the standard separation at time t, we calculate the distance between the two segments modeling the aircraft positions and compare it to

---

[4] Aircraft are able to follow a given route at a given altitude, but their speed can not be precisely predicted. Consequently, controllers prefer waiting to be sure two aircraft are involved in a conflict before solving it.
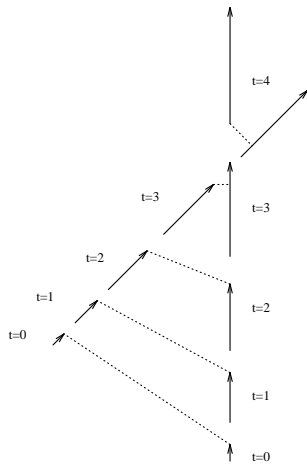
Figure 2: Modeling of speed uncertainties.

Figure 3: The Model.

the standard separation (see figure 2).

Because of uncertainties, a conflict may be detected twenty minutes before it should occur and finally may not happen. Consequently, deciding to move an aircraft in that case would be useless, and could even generate other conflicts that would not occur if no maneuver had been decided. Consequently, it is very important that the solver makes a decision as late as possible without penalizing the resolution.

2.2 Choosing the model

Both offset modeling and turning point modeling must be kept : for overtaking aircraft, offsets are more efficient whereas for other conflicts, turning points are more efficient. However, because of uncertainties, maneuvers must be started as late as possible with respect to the aircraft constraints. This explains the choice of a constant turning point angle (30 degrees, for example allows large deviations in very few time). The previous elements leads us to choose the following model (figure 3). A maneuver will be determined by :

- the maneuver starting time T.

- the duration of the maneuver and its direction D (right or left).

- the duration of the offset O (if no offset, this value will be 0).

T and O are positive values whereas D will be positive if the aircraft turns on its right and negative if it turns on its left. This model has the great advantage to reduce the size of the problem. For a conflict involving n aircraft, the dimension of the research space is $3n$. This will allow us to
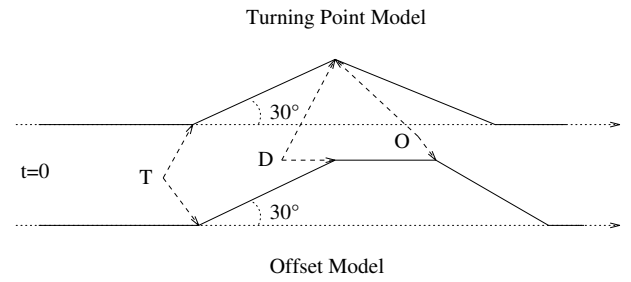
solve very difficult conflicts with many aircraft without investigating a too large space. The lost of optimality on delays is not important if speed uncertainties are large. Even if a 30 degrees deviation is 3 times more penalizing than a 10 degrees turning point, the total duration of a 10 degrees deviation is more than 3 times the duration of a 30 degrees deviation. Thus, when a 10 degrees deviation is started, it will last 3 times the 30 degrees equivalent deviation and it will have to be started earlier, when the probability of conflict is lower. The optimal deviation angle is thus directly depending on the aircraft speed errors and could be reduced in the future.

2.3 Real time optimization

The model introduced above is simple enough to be used in a real time optimization program[5]. Let's consider a conflict involving n aircraft and let's choose a time step (δ minutes for example). Let's imagine we want to recompute all trajectories every δ minutes. During the optimization time, aircraft are flying and must know if they must change their route or not. Consequently, for each aircraft, at the beginning of the current optimization, the next δ minutes of its flight may not be changed and is fixed by the previous optimization. Because of the constraints described above when a maneuver is started for an aircraft, it can not be changed, but at least the part of the maneuver that is not included in the first δ minutes may be changed with respect to the turning point or offset model. On example 4, at $t = 0$ on the first line, the aircraft trajectory can not 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 occured between $t = \delta$ and $t = 2\delta$ is kept as a constraint for the second optimization run (on the example, no maneuver is decided)...

On the example given, we can see that the maneuver described on line 2 (resulting of an optimization at $t = \delta$)

---

[5] Simulating 20 minutes of aircraft flight and checking standard separations is very long. Consequently, a simple modeling is necessary to achieve a real time optimization.
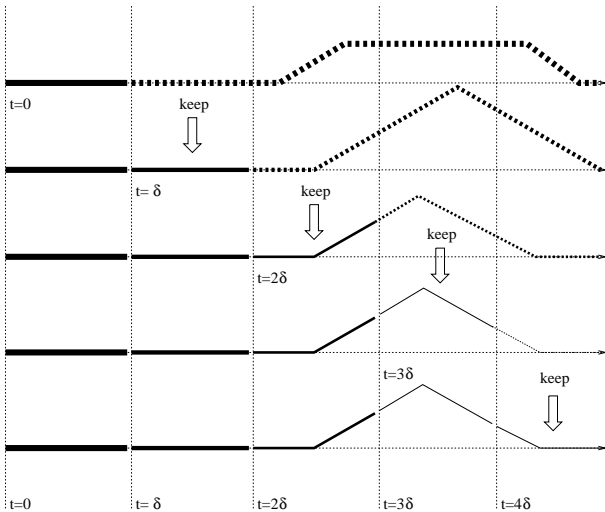
Figure 4: The model and real time optimization.

is more penalizing than the maneuver described on line 3 (resulting of 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 real time situation.

3 A combinatorial problem

In this part, we explain why our problem can not be solved by classical methods.

3.1 Complexity of the problem

The complexity of the problem is exposed by Medioni, Durand and Alliot in [16]. Let's consider a conflict between two aircraft. If we can easily prove that the minimized function is convex, 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^n connected components in the free trajectory space which strongly suggests that any method which requires exploring every connected component is NP. It is important to note that this complexity is independent of the modeling chosen. The offset modeling seems to be very attractive, because it linearizes constraints. Nevertheless, each constraint multiplies by two the number of linear programs to solve. Our problem involves $\frac{n(n-1)}{2}$ constraints. Moreover, linearizing the minimized function, multiplies by 2^n the number of linear program to solve (we minimize the sum of each aircraft offset which may be positive or negative). Finally, we will have to solve $2^{\frac{n(n+1)}{2}}$ linear pro-

grams, each one involving $\frac{n(n+1)}{2}$ linear constraints. For $n = 5$, 32768 linear programs have to be solved with 15 constraints in each program.

3.2 A global optimization problem

Using classical methods, such as gradient methods for example, becomes useless for our problem, because of the arbitrary choice of the starting point required by these methods. 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 can not lead by a classical method to an optimum in another component. We can thus expect only a local optimum. Practical attempts done on LANCELOT *Large And Nonlinear Constrained Extended Lagrangian Optimization Techniques* [5] have confirmed this problem. Furthermore, this approach is not efficient for a real time trajectory planning. The offset modeling linearizes the constraints but the problem remains so combinatorial that we can not expect to find the global optimum efficiently. Moreover, the separation during maneuvers must be checked afterwards. This experiment strongly suggests that classical methods are not well adapted to our problem.

4 Genetic Algorithms

4.1 Principles

We are using classical Genetic Algorithms and Evolutionary Computation principles such as described in the literature [13, 17]; Figure 5 describes the main steps of GAs.

First a population of points in the state space is randomly generated. Then, we compute for each population element the value of the function to optimize, which is called *fitness*. In a second step we select[6] the best individuals in the population according to their fitness. Afterwards, we randomly apply classical operators of crossover and mutation to diversify the population (they are applied with respective probabilities P_c and P_m). At this step a new population has been created and we apply the process again in an iterative way.

4.2 Improvements

4.2.1 Simulated Annealing Tournament

GA can be improved by including a Simulated Annealing process after applying the operators [14]. For example, af-

---

[6] Selection aims at reproducing better individuals according to their fitness. We tried two kinds of selection process, "Roulette Wheel Selection" and "Stochastic Remainder Without Replacement Selection", the last one always works out better.
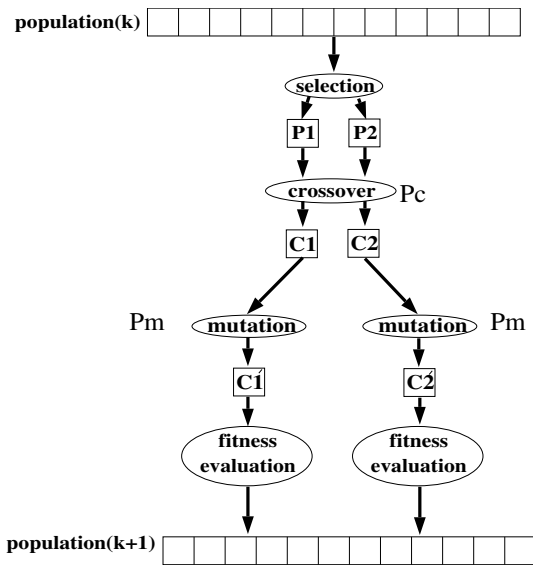
Figure 5: GA principle

Figure 6: GA and SA mixed up

ter applying the crossover operator, we have four individuals (two parents $P1$,$P2$ and two children $C1$,$C2$) with their respective fitness. Afterward, those four individuals compete in a tournament. The two winners are then inserted in the next generation. The selection process of the winners is the following : if $C1$ is better than $P1$ then $C1$ is selected. Else $C1$ will be selected according to a probability which decreases with the generation number. At the beginning of the simulation, $C1$ has a probability of 0.5 to be selected even if its fitness is worse than the fitness of $P1$ and this probability decreases to 0.01 at the end of the process. A description of this algorithm is given on figure 6.

Tournament selection brings some convergence theorems from the Simulated Annealing theory. On the other hand, as for Simulated Annealing, the (stochastic) convergence is ensured only when the fitness probability distribution law is stationary in each state point [1].

4.2.2 Sharing

Our problem is very combinatorial and may have many different optimal solutions. In order to find most of these solutions[7] and to avoid local optima, the sharing process introduced by Yin and Germay [20] is used. This sharing process has the great advantage to grow in $n \log(n)$ (instead of n^2 for classical sharing) if n is the size of the population.

A sharing process requires introducing a distance between two chromosomes. Defining a distance between two

---

[7] Finding several solutions is very interesting because the controller can make a choice and negotiate it with pilots.
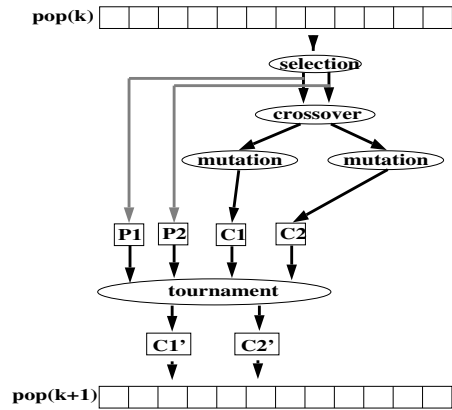
trajectories is not very simple. In this study, we define five trajectory types :

- the aircraft is not deviated
- the aircraft is deviated on the right for a turning point
- the aircraft is deviated on the left for a turning point
- the aircraft is deviated on the right for an offset
- the aircraft is deviated on the left for an offset

A discrete distance is defined as follow : two chromosomes belong to the same class if each of their aircraft follow the same trajectory types. If not, they belong to different classes. The fitness value of each chromosome is then divided by the number of elements in its class. Results show that sharing was very useful for combinatorial problems.

Simulated annealing and sharing process have really improved convergence of GAs and were definitely adopted for the three following applications.

4.2.3 Adapted Crossover and Mutation

GAs seem very powerful because they do not require much information on the fitness function. However, classical operators used by Gruber, Alliot and Schoenauer in [2] did not give good results. Moreover, we know much about the fitness function and it can be very useful to use this information to create adapted crossover and mutation operators. Durand, Alliot and Noailles describe these operators in [11]. Instead of considering a global fitness value that takes into account the different lengthening of the trajectories and the conflicts between the aircraft, we keep in a n^2 sized matrix F (where n is the number of aircraft) these values. If $i \neq j$, $F_{i,j}$ measures the conflict between
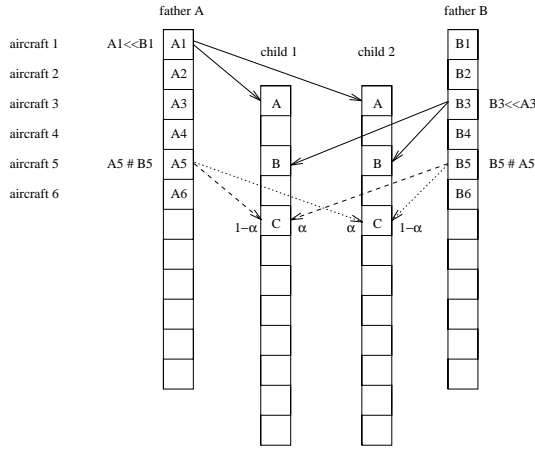
Figure 7: An adapted crossover operator

aircraft i and j. It is set to 0 if no conflict occurs and increases with the seriousness of the conflict. $F_{i,i}$ measures the lengthening of aircraft i trajectory. This fitness matrix contains much more information than the previous scalar global fitness and this will allow us to define more deterministic crossover and mutation operators. We can define for each aircraft :

$$F_i = \sum_{j=1}^{n} (F_{i,j})$$

The crossover operator is described on figure 7. After choosing 2 parents A and B, we compare the local fitnesses of their aircraft (These fitnesses are named A_i and B_i on figure 7). For aircraft i, if $A_i \leq B_i - \Delta$ (Δ is a value that modulates the operator's determinism) the 2 children inherit aircraft i of father A. If $B_i \leq A_i - \Delta$, the 2 children inherit aircraft i of father B. If none of these two conditions are checked, aircraft i of children 1 and 2 are two random combination of aircraft i of the two parents.

The mutation operator is described on figure 8. After choosing a chromosome, an aircraft is mutated (on figure 8 aircraft 4 is chosen). Aircraft having a high fitness (more than ϵ where ϵ modulates the operator's determinism) can be chosen only if every aircraft has a high fitness.

These operators have the great advantage to be rather deterministic at the beginning of the optimization so that a solution without conflict can be very quickly found. When such solutions become sufficiently numerous, these operators are less deterministic and other parts of the research space can be explored.

Two parameters (Δ and ϵ) where introduced and influence the determinism of the optimization. Because of the determinism of these operators, using sharing (described above) becomes essential.
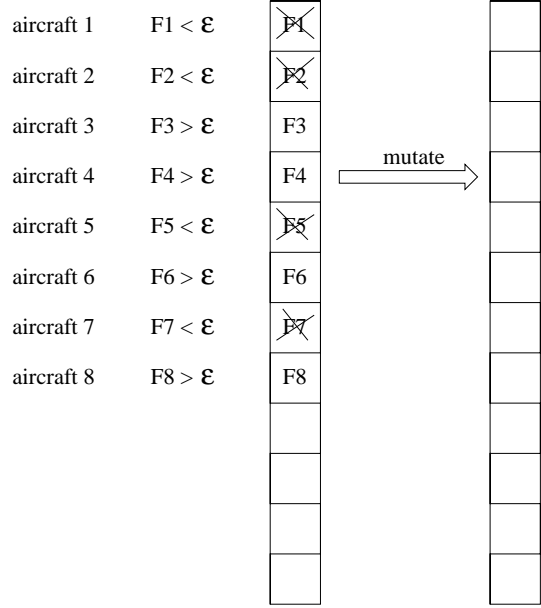

Figure 8: An adapted mutation operator

4.2.4 Improving the result

Genetic algorithms are very efficient to solve global combinatorial optimization problem but are not very efficient to solve local search with a good precision. Consequently, it is very efficient, at the last generation of the genetic algorithm, to use a local optimization method to improve the best solution of each chromosome class defined above. The local method adopted in this study is very simple : we apply a hill-climbing algorithm to the best chromosomes at the end of the GA run..

5 Coding our problem

5.1 The data

An example of chromosome is given in figure 9. Each value represents a time and is coded by a positive integer. For the deviation duration, this integer can be negative.

The duration of the optimization, the anticipation time, the time step used during the simulation, the horizontal separation must be define for every optimization.

Some data must be actualized at each optimization : the position, heading, speed and speed uncertainty[8] of each aircraft.

Other global data are required : the number of generations, the number of elements, crossover and mutation per-

---

[8] It can depend on its characteristics, its equipment and on the weather conditions.
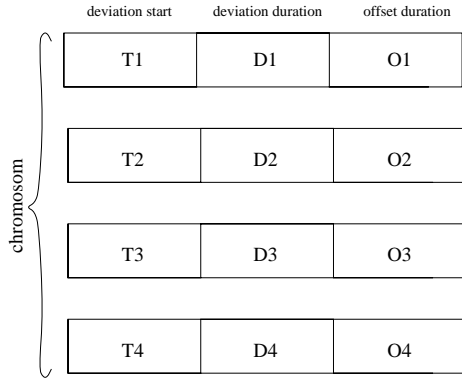
Figure 9: Structure of the chromosome

centage.

5.2 The fitness

One of the main issues is to know how to compute the *fitness* of a chromosome. We have a poly criteria problem to solve, in fact the following criteria have to be matched together to give us a single fitness function :

- The delay due to a deviation for each aircraft must be as small as possible.

- However, the number of aircraft deviated and the total number of maneuvers must be as low as possible (orders are given by the controllers to pilots on the VHF). Thus, instead of sharing the global delay on all the aircraft, some aircraft will support a part of the delay and others not. The sharing process described above is very important to give different equivalent solutions.

- The maneuver duration for an aircraft must be as short as possible so that the aircraft is freed as soon as possible for another maneuver.

- The trajectories must handle the separation constraints.

The fitness matrix previously described is calculated as follow :

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

To take into account both the delay and the maneuver duration, we decided to measure the maneuver occupancy surface S (figure 10). This value increases with both the delay and the maneuver duration. It is interesting to note
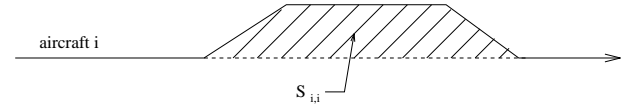

Figure 10: Maneuver Occupancy Surface

that controllers currently use this notion of maneuver occupancy surface. To minimize the number of maneuvers, add to S the number of maneuvers N multiplied by a coefficient k ($N = 0$ if no maneuver, 2 if it is an turning point and 3 if it is an offset).

At each time step t, we calculate $C_{t,i,j}$ as the difference of the standard separation and the distance between the segments i and j describing aircraft i and j position at time t. These values are added in $F_{i,j}$ and give a measure of the conflict between i and j.

It is obvious that the fitness matrix is symmetric. A triangular matrix could as well be used.

We can now define a global scalar fitness as follow :

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

This fitness function guarantees that if a chromosome value is larger than $\frac{1}{2}$, no conflict occurs. If a conflict remains the fitness does not take into account the delays induced by maneuvers.

6 Numerical Applications

These applications have been done to justify the model and illustrate the performance of the algorithm.

6.1 Model Justification

In this first application, we consider the conflict between two aircraft described on figure 11. The anticipation time is set to 3 minutes and aircraft speed (400 knots) are known with an error of 5 per cent. In this case, if aircraft 0 real speed is 380 knots and aircraft 1 real speed is 420 knots, because of uncertainty, at last no conflict occurs (the standard separation is 4 nm). However, solutions must be computed "in case". Figures 12 show the result of the genetic algorithm at time 4 minutes, 7 minutes, 10 minutes and 13 minutes. The continuous line describes the next 3 minutes trajectory that can not be changed, the dash line represents the optimal trajectory computed by the genetic algorithm. The parameters of the genetic algorithm were :
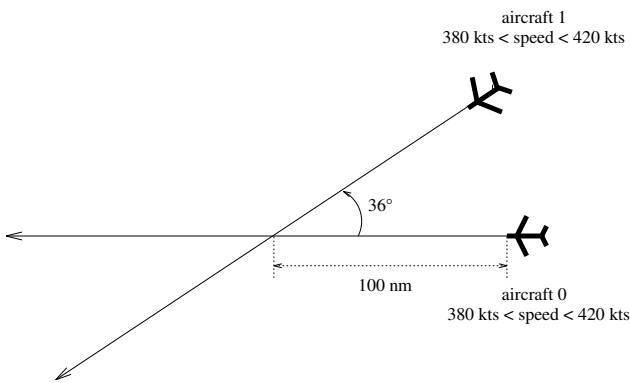
Figure 11: 2 aircraft conflict

number of generations : 20
population elements : 50
percentage of crossover : 60
percentage of mutation : 15
simulated annealing for crossover : yes
sharing : yes
hill-climbing method : yes

Because of uncertainties, the initial optimized trajectory is not very good. As times goes on, aircraft are closer to the conflict point and uncertainty decreases so the optimized trajectories give smaller deviations. Finally, at time $t = 13$, the conflict disappears. This example demonstrates the interest of the model because it allows uncertainties and delays maneuvers as long as possible. Thus, the genetic algorithm gives an optimal command for the next 3 minutes and the optimal trajectory associated.

6.2 Long lasting conflicts

In this test, 5 aircraft are coming from the west and going to the east with different speeds (the furthest has the greatest speed and the nearest the lowest speed). The aircraft are not exactly following the same routes. Figure 13 gives the trajectories of the 5 aircraft with 5 per cent error on speed and 5 minutes of anticipation time. The size of the population of the genetic algorithm was doubled.

This example was done to show that our model allowed long lasting deviations when they can not be avoided. Here, aircraft 5 is deviated for more than 30 minutes. This example shows the importance of the offset model to solve long lasting conflicts. Solving this problem with turning points would have induced much larger delays and very big deviations (in the example, aircraft 5 would have been deviated several hundreds of nautical miles away from its trajectory).
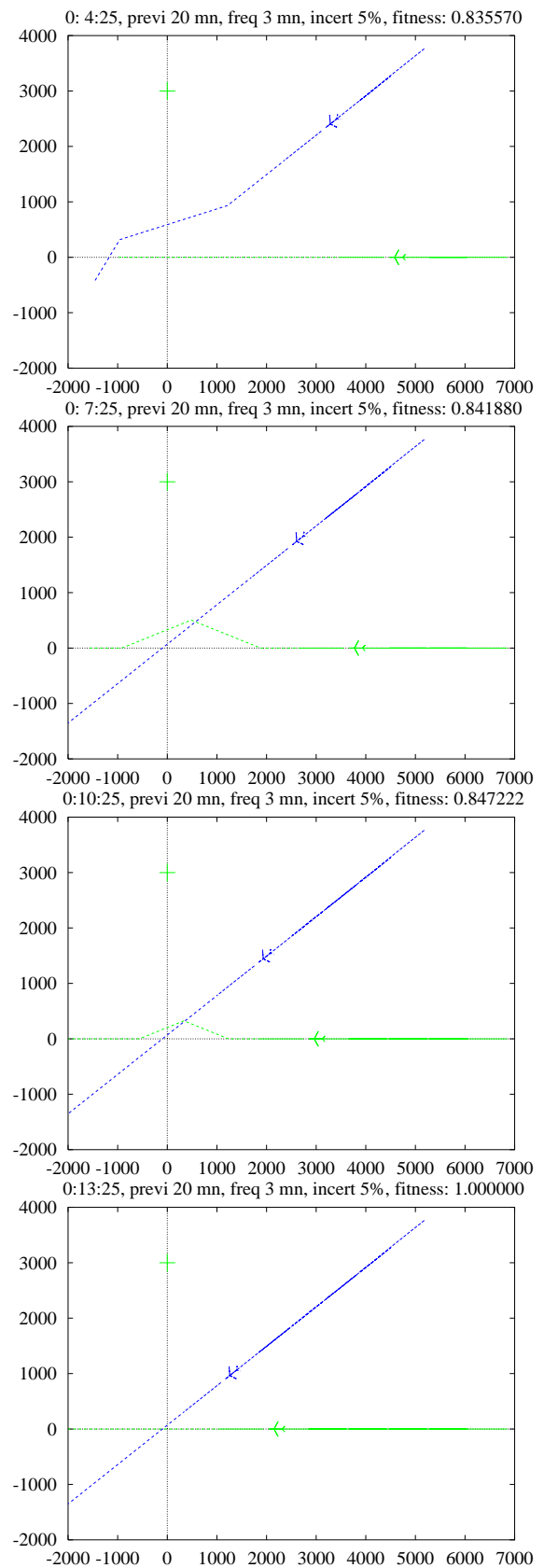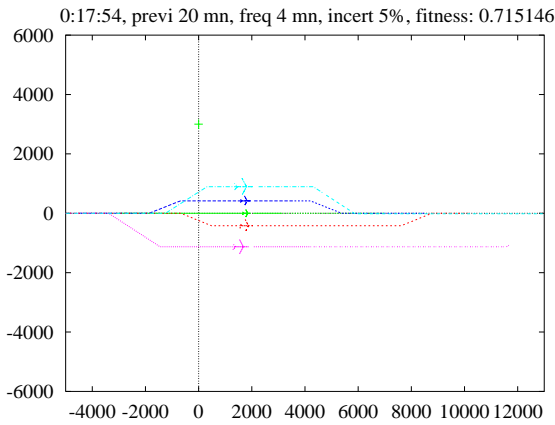

Figure 12: t=4, 7, 10, 13 minutes

0:17:54, previ 20 mn, freq 4 mn, incert 5%, fitness: 0.715146

Figure 13: 5 aircraft conflict (aircraft come from the left)

6.3 Adapted operators and sharing justification

To show the importance of the sharing process and of the adapted operators introduced in the Genetic Algorithm, a conflict involving 20 aircraft on a circle (figure 14) was tested. It has very few solutions because all the aircraft are conflicting with each other. Only solutions where all the aircraft (except perhaps one) turn in the same direction are conflict free. Figure 15 gives the result of the optimization with a speed error of 3 per cent and 5 minutes anticipation time. The parameters of the genetic algorithm were :

number of generations : 20
population elements : 100
percentage of crossover : 60
percentage of mutation : 15
simulated annealing for crossover : yes
sharing : yes
hill-climbing method : yes

The minimum distance between aircraft is 5.6 nm (the standard separation was fixed at 4 nm).

In order to validate the use of a sharing process combined to adapted operators, 4 different tests were done on this two examples using :

- no sharing process and adapted operators (described previously).

- a sharing process and adapted operators.

- no sharing process and classical operators.

- a sharing process and classical operators.

For these 4 tests, the value of the best fitness was measured (figure 16) on the 20 generations.
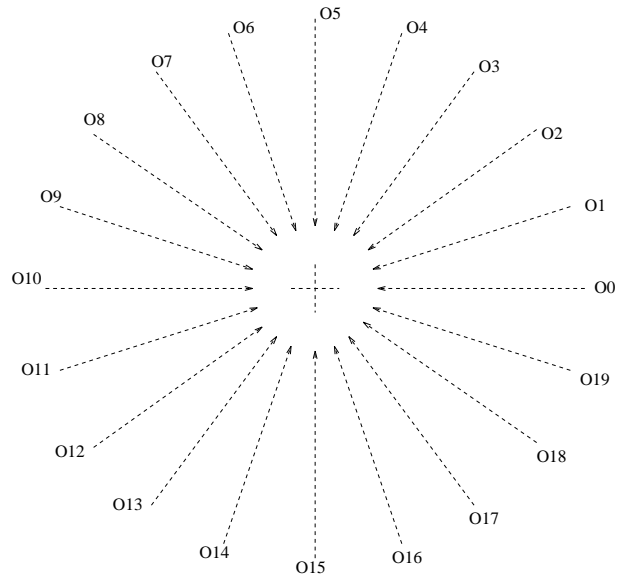

Figure 14: 20 aircraft conflict (aircraft on a circle)

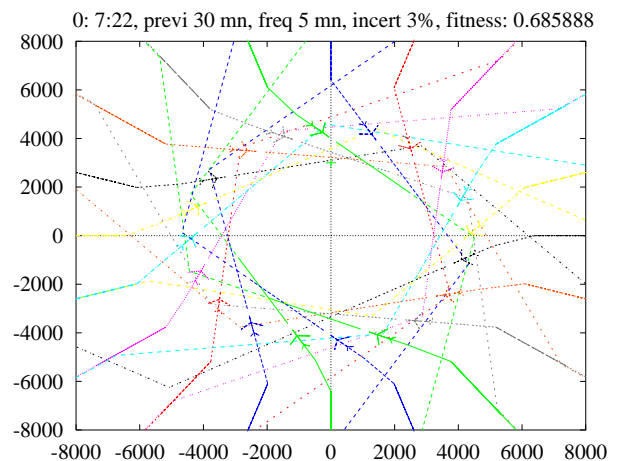0: 7:22, previ 30 mn, freq 5 mn, incert 3%, fitness: 0.685888
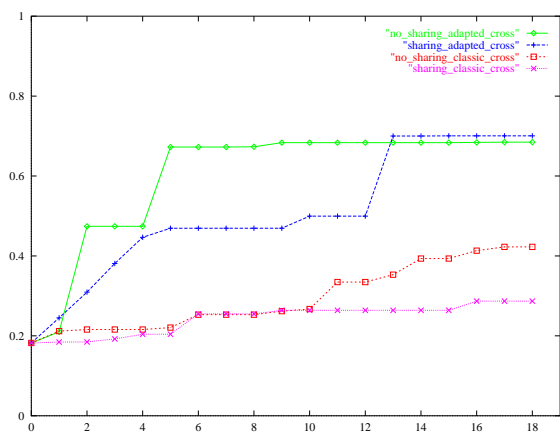
Figure 15: 20 aircraft conflict free

Figure 16: Best fitness (aircraft on a circle)

First, we can check that adapted operators are very efficient because with classical operators, no conflict free solution is found before generation 20 (tests have shown that the first conflict free solution is found after generation 200) whereas with adapted operators a conflict free trajectory is always found before generation 10 (A solution is conflict free if its fitness is more than 0.5).

Figure 16 show that the sharing process lengthens the growth of the best fitness. However, the final best fitness is as good with a sharing process as it is without. It is even better with a sharing process when the conflict is very difficult to solve. The sharing process has the great advantage to help the genetic algorithm to avoid local minima and to give different conflict free solutions.

These tests were repeated on different examples involving 20 aircraft[9] giving the same results and justifying the use of adapted operators with a sharing process for conflict resolution.

Conclusion

Genetic Algorithm have shown their efficiency to solve aircraft conflicts in a real time situation. However, we have shown that improvements are very useful, especially when the dimension of the problem grows. This combinatorial problem has shown that combining local and global optimization methods could help solving real variable combinatorial problems.

The model introduced in this paper is a first step in the resolution of real conflicts with real aircraft. A full simulation on a one day real traffic has been done by Chansou [4]. Results are very promising.

---

[9]Instead of choosing 20 aircraft on a circle 10 independant conflicts involving 2 aircraft were used

The next step consists in developing a graphical user interface to compare human being and computer resolutions. The trajectory forecast model should also be improved.

References

[1] Emile Aarts and Jan Korst. *Simulated annealing and Boltzmann machines*. Wiley and sons, 1989. ISBN: 0-471-92146-7.

[2] 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.

[3] J. M. Alliot and Thomas Schiex. *Intelligence Artificielle et Informatique Théorique*. Cepadues, 1992. ISBN: 2-85428-324-4.

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

[5] A.R. Conn, Nick Gould, and Ph. L. Toint. A comprehensive description of LANCELOT. Technical report, IBM T.J. Watson research center, 1992. Report 91/10.

[6] 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.

[7] 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.

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

[9] Nicolas Durand. Conflict free trajectory modelling for en route control. Technical report, ENAC/CENA, January 1994.

[10] 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.

[11] Nicolas Durand, J. M. Alliot, and Joseph Noailles. Algorithmes genetiques : un croisement pour les problemes partiellement separables. In *Proceedings of the Journees Evolution Artificielle Francophones*. EAF, 1994.

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

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

[14] Samir W. Mahfoud and David E. Goldberg. Parallel recombinative simulated annealing: a genetic algorithm. IlliGAL Report 92002, University of Illinois at Urbana-Champaign, 104 South Mathews Avenue Urbana IL 61801, April 1992.

[15] 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.

[16] Frédéric Médioni. Algorithmes génétiques et programmation linéaire appliqués a la résolution de conflits aériens. Master's thesis, Ecole Nationale de l'Aviation Civile (ENAC), 1994.

[17] Zbigniew Michalewicz. *Genetic algorithms+data structures=evolution programs*. Springer-Verlag, 1992. ISBN: 0-387-55387-.

[18] 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.

[19] 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.

[20] Xiaodong Yin and Noel Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In C.R. Reeves R.F.Albrecht and N.C. Steele, editors, *In proceedings of the Artificial Neural Nets and Genetic Algorithm International Conference, Insbruck Austria*. Springer-Verlag, 1993.

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