Genetic algorithms for air traffic assignment

Daniel Delahaye, Jean-Marc Alliot, Marc Schoenauer, Jean-Loup Farges

Genetic algorithms for air traffic assignment

Daniel Delahaye[1] and **Jean-Marc Alliot**[2] and **Marc Schoenauer**[3] and **Jean-Loup Farges**[4]

Abstract. In this paper, we show how genetic algorithms can be used to compute automatically a traffic assignment of aircraft on the air network to increase Air Traffic Control capacity in high density areas.

1 Introduction

The CENA is the organism in charge of studies and research for improving the French ATC systems. Studies on the use of genetic algorithms for conflict resolution and air space sectoring have given encouraging results [2, 5], and a new study has been funded to solve the traffic assignment problem for Air Traffic Control. This paper summarizes the results of this study.

When joining two airports, an aircraft must follow routes and beacons; these beacons are necessary for pilots to know their position during navigation and help controllers to visualize the traffic. Beacons can either be real radio-navigation means, such as VOR or ADB, or only the crossing points of two or more different airways.

Two aircrafts passing the same beacon can collide if their trajectories become too close; we say that there is a *conflict* if aircrafts are closer than (usually) 5 Nautic Miles.

At the dawn of civil aviation, pilots resolved conflicts themselves because they always flew in good weather conditions (good visibility) with low speed aircrafts. On the other hand, modern jet aircrafts do not enable pilots to resolve conflicts because of their high speed and their ability to fly with bad visibility. Therefore, pilots must be helped by an air traffic controller on the ground who has a global view of the current traffic distribution in the airspace and can give orders to the pilots to avoid collisions.

As there are a lot of planes simultaneously present in the sky, a single controller is not able to manage all of them. So, airspace is partitioned into different sectors, each of them being assigned to a controller. In a few words (we will discuss this in more details later), workload for a controller mainly depends on three terms:

- the number of planes in the sector; the controller has to monitor the trajectories: it is called *monitoring workload.*
- the number of planes crossing the sector boundaries; when an aircraft crosses a sector boundary, controllers in charge of the two sectors have to exchange informations about the flight: this is called *coordination workload.*
- the number of plane crossings over beacons. Each crossing induces a possibility of conflict, and the controller has to solve these conflicts: it is called *conflict workload.*

Nowadays, in Europe, traffic assignment can be, and need to be, seriously improved; when an aircraft goes from airport A to airport B, the pilot chooses himself a route and informs the ATC organism with a flight plan. The problem is that many planes choose to follow the same routes, and, moreover, never consider the global optimum of the system, but only their own optimality function. This way of choosing routes sometimes generates many conflicts on the same beacon inducing overloaded sectors. Traffic assignment aims at changing aircraft routes to reduce sector congestions, conflicts and coordinations. Of course, those changes must be done in a way that does not penalize too much aircraft routes. So a compromise between the aircraft objectives and the ATC objectives must be found.

There is however a serious constraint for traffic assignment: all planes going from airport A to airport B must follow the same route (we call this constraint *non segmented Origin-Destination flow*). The reason for this constraint is easy to understand: if two airlines are linking A to B, the airline being given the longest route would find this clearly unfair...

The goal of this study is to find an automatic way to optimize aircraft routing, given aircrafts, airlines and ATC constraints. In this paper, we show how well Genetic Algorithms deal with this problem (after some simplifications and modeling). In the first part we describe more precisely our problem and make some relevant simplifications, in the second one we present the principle of resolution and we finally give different results on test networks used to validate the algorithm.

2 A simplified model

2.1 Introduction

Our 3 dimensional transportation network (in the air space) will be modeled by a classical plannar 2 dimensional network.

This network is partitioned into K sectors, each one being assigned to a controller who manages all the traffic in it. An example of sectorised network is given on figure 1 (black points represent airports).

The controller workload has several origins that can be divided into two categories [11]:

1. there are quantitative factors which include the number of flights, the number of conflicts etc, which can be precisely modeled in a mathematical way and handled by an optimization algorithm;

[1] Centre d'Etudes de la Navigation Aérienne
[2] Ecole Nationale de l'Aviation Civile
[3] Centre de Mathématiques appliquées de l'Ecole Polytechnique
[4] Centre d'Etudes et de Recherche de Toulouse

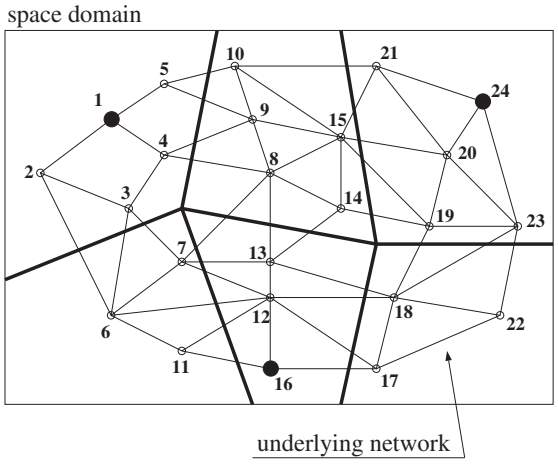

Figure 1. Sectorised network

2. there are psychological factors such as stress, concentration etc... which have no evident mathematical formulation but are in direct relationship with the previous ones according to the controllers themselves.

Having now a model, we can define more precisely our goals:

we consider an air traffic transportation network in a 2 dimensional space sectorised into K sectors for which we want to assign traffic between OD pairs in a way that minimizes extra route distance and reduces sector workloads.

This traffic assignment must take the following constraint into account :

1. the flow between each OD pair must not be partitioned.

This constraint is mandatory, as we want to be sure that planes of different airlines on the same Origin-Destination will follow the same route: this is the *equity constraint.*

2.2 Definition and goals of traffic assignment

A transportation system basically consists of two elements [10]: transportation supply and travel demand. The transportation supply is the set of facilities and means available to the users of the transportation network. The travel demand is expressed by the number of users using the network. The interaction between transportation supply and travel demand produces a flow pattern on network links. The goal of traffic assignment is to optimize this interaction to reach a predefine objective dependent of the problem. In our problem the main difficulty relies in the underlying dependence of the link costs coming from the sectoring. As a mater of fact the cost on one link depends directly on the flow on this link but is also in relationship (indirectly with the global objective function) with the flows of all the links in the same sector (we have *non-separable asymmetric link costs*). This last point has a serious influence on the principle of resolution. We are going to discuss this in the next section.

3 Principle of resolution

3.1 Traffic assignment algorithms

The equilibrium principle of route choice propounded by Wardrop in 1952 forms the basis of all those algorithms. This principle says:

"the journey times on all routes actually used are equal, and less than those which would be experienced by a single aircraft on any unused route."

There is a key point in equilibrium assignment: when the level of demand is sufficiently high, several routes can be used between the same Origin-Destination pair. This dispersion of demand over more than one route is entirely realistic: it is achieved in equilibrium assignment when the assignment of all the demand to any single route would make the cost on that route exceed the cost on some other routes. In these circumstances, the least cost route is not unique, but rather a number of routes can be used at identical costs. The process of calculating an equilibrium assignment of traffic for a network can normally be achieved by an iterative processus and many traffic assignment algorithms are based on this principle: they put quanta of traffic on different routes till the equilibrium is reached. Our problem being a traffic assignment problem on a network with non separable asymmetric link costs "only" Dafermos [4] algorithms (or algorithms based on the same principle) manage this kind of problem. Why can not we use those algorithms: because none of the Dafermos family algorithms handles the equity constraint that forces all aircrafts joining two airports to use the same route.

It is clear that handling this constraint will induce worse solution as it reduces the solution subspace in the state space. But the equity constraint can not be discarded for political reasons. So we have to try to find an algorithm which puts all the flows on the same route between each OD.

If our link costs were separable a simple Dijkstra algorithm [?] would have solved the problem because traffic assignment would not have relied on the choosing order of Origin-Destination pairs when assigning flows. But with non separability, traffic assignment depends on the order: as soon as a path is assigned for an OD pair, all previous assignments must be reconsidered as the new assignment changes the cost function by changing the link costs for all links in its sector.

So our problem induces a high combinatory complexity for which we must try to find a solution in a discrete space with $n!$ points where n is the number of Origin-Destination pair, a problem known to be NP_HARD.

According to the number of Origin-Destination pair we have to handle (several hundred), classical combinatorial optimization is not relevant and stochastic optimization seems to be more suitable. Moreover this kind of problem may have several optimal (or near optimal) solutions. As our goal is not to build the ultimate traffic assignment system, but a tool to help human experts assigning flows, we are definitely interested in all optimal or nearly optimal solutions the algorithm might find. This made us reject classical simulated reannealing optimization which updates only one state variable, even if it might give better results in some cases [7].

On the other hand, Genetic Algorithms (GAs) maintain and improve a numerous population of state variables according to their fitness and will be able to find several optimal (or near optimal) solutions. Then, GAs seem to be relevant to

solve our traffic assignment problem.

4 Genetic algorithms

4.1 Principles

We are using classical Genetic Algorithms and Evolutionary Computation principles such as described in the litterature [6, 9]; Figure 2 describe the main steps of GAs.

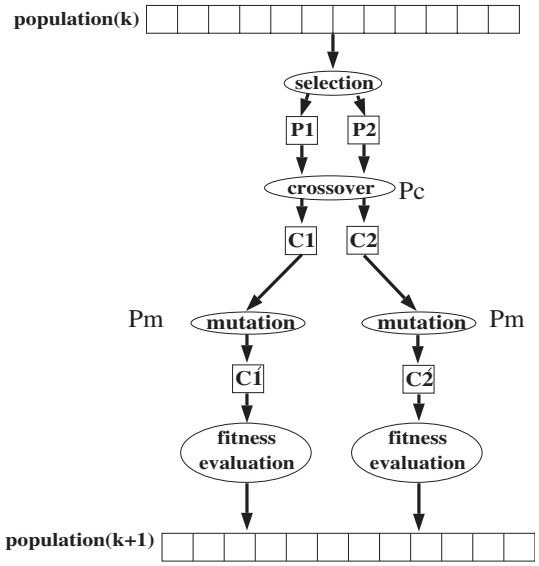

Figure 2. GA principle

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 the *fitness*. In a second step we select[5] the best individuals in the population according to their fitness. Afterward, 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.

This GA can be improved by including a Simulated Annealing process after applying the operators [8]. For example, after 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[6] is given on figure 3.

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].

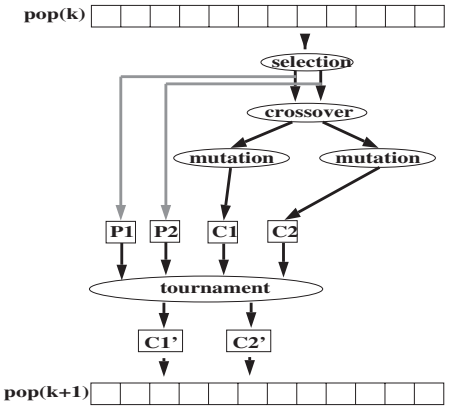

Figure 3. GA and SA mixed up

4.2 Coding our problem

To code our problem, we did not use binary chromosomes. The problem is not well suited for binary coding, and, as it has been advocated already by different experts, a specific coding with specific operators is usually more efficient.

An example of the coding of a chromosome is given in figure 4. The chromosome is a list of cells; each cell is the coding of the path for an OD flow. On the example, we see that all planes going from airport 1 to airport 16 will follow the path: airport 1, beacons 4,3,7,12 and airport 16. Planes from 16 to 1 will follow the path 16,11,6,3,4,1 and etc. So, all information necessary is encoded in each chromosome. It enables us to compute for each chromosome the traffic assignment cost giving the GA fitness.

One difficult point is the initialization of the population: to create one chromosome, we take into account distance costs only and we increase them by a random extra cost. Then, we apply a Dijkstra algorithm to find min cost paths for all the Origin-Destination pairs. This generates a list of OD paths which is our chromosome. We repeat these operations till the population size is reached. According to the deviation of the blank noise added as a cost, paths more or less different from the optimal ones are generated (optimal in the sense of the distance criterium only of course). This initialization method avoids the creation of purely random paths and ensures, for instance, that an aircraft coming from Madrid and going to London will not be routed via Moscow.

We had then to create operators for crossover and mutation. The efficiency of the algorithm depends of the ability of these operators to create new individuals that respect the constraints of our problem and that generate paths not too far from the optimal ones.

[5] Selection aims at reproducing better individual according to their fitness. We tried two kinds of selection process, Roulette Wheel Selection" and "Stochastic Remainder Without Replacement Selection", the last one always gave better results.

[6] We are using our own GA simulator, which includes some goodies usually not available on public domain GA, such as Simulated Annealing, very simple parallelism, etc.

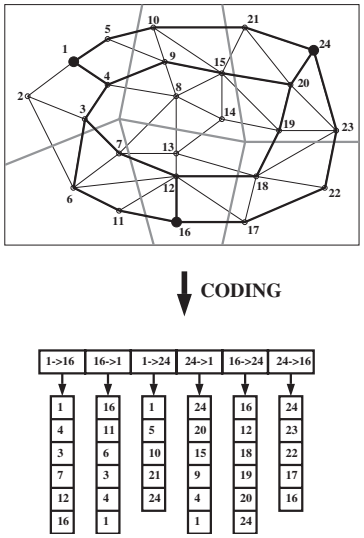

Figure 4. Structure of the chromosome

The crossover is implemented as a slicing crossover: after selecting two parents in the current population, we randomly chose an allele position creating twice two paths subsets. Then, we just exchange the two last subsets to create two children. An example of crossover is given on figure 5.

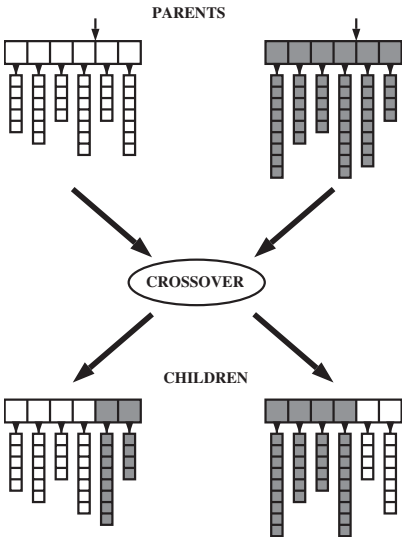

Figure 5. Crossover operator

To mutate a chromosome, we randomly select an allele position and generate a new path for the Origin-Destination pair selected by the same process as for generating the initial population. An example, of mutation is given on figure 6.

4.3 Results

To validate our algorithm, we used a toy network for which we knew a trivial traffic assignment solution (this network is drawn on figure 7). All the nodes on the first diagonal (upper-left to lower-right) are airports, all nodes on the other diagonal are beacons.

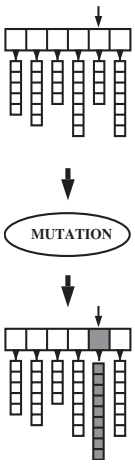

Figure 6. Mutation operator

All airports in the upper left corner generate a traffic flow which must be routed to the symmetrical airport (relative to the center of the web) in the lower right corner. Respectively, each airport in this corner generates a flow that must be routed to the symmetrical airport in the upper left corner. Face to face flows on the same link are forbidden and link capacity is very limited, in order to prevent two flows from being routed on the same link.

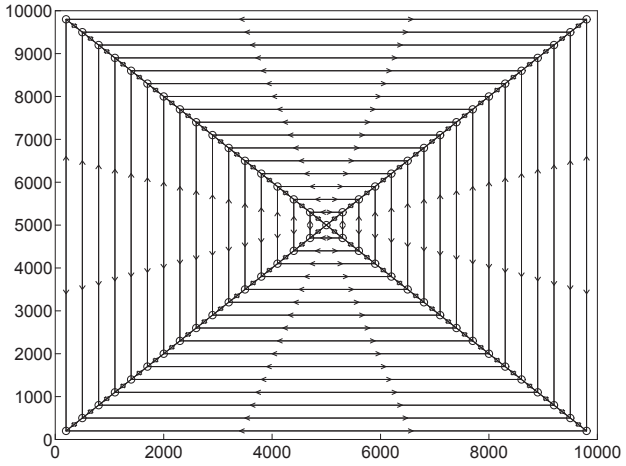

Figure 7. Test network

The parameters for the simulation were:
Population size: 400
Number of generations: 300
Probability of crossover: 0.6
Probability of mutation: 0.1
The evolution of best-ever chromosome fitness and aver-

age chromosome fitness is displayed in figure 8: an optimal solution is found at generation 180.

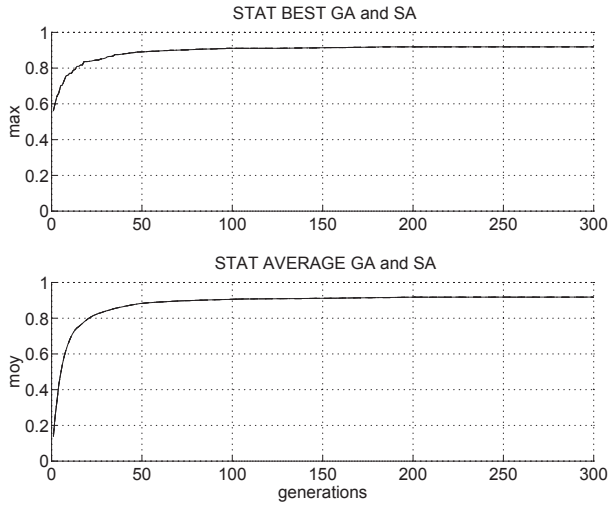

Figure 8. Fitness evolution

The solution is displayed in figure 9. It is clearly a correct solution (there were many other solutions with the same fitness: the direction of planes on each link can be either clockwise or counter clockwise). It must be noted that, even if this solution is trivial to find for a human being because of the symmetries of the problem, it remains as difficult as any other problem for our algorithm.

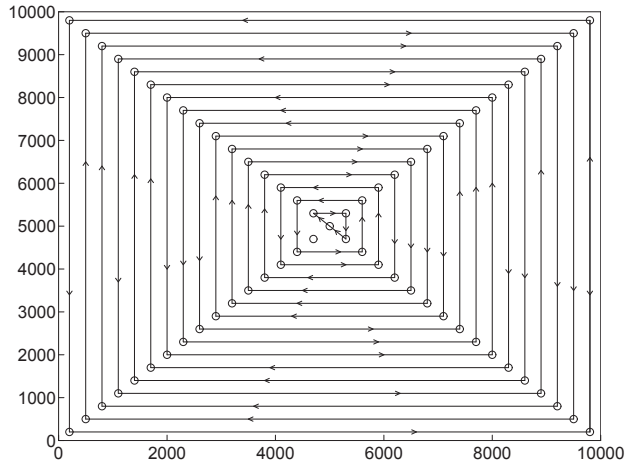

Figure 9. Traffic assignment result

The algorithm was then tested on more realistic networks (too large to be presented here) and gave also good results; moreover, we were not able to find a better traffic assignment by hand, which is a good presomption of a correct behavior of the algorithm.

5 Conclusion

This study shows how Genetic Algorithms are suitable to solve the traffic assignment problem with no segmented flows. We now consider this part of our work as completed. It is already an interesting result that can be used to build tools to help viewing and assigning flows.

But we are focusing now on mixing our two algorithms: the one described in [5] which does a sectoring of space given assigned paths for each OD flow, and this one which, given a sectoring, assigns paths.

Building a single algorithm able to do an optimal sectoring and flow assignment in one step, is a difficult, useful and interesting challenge.

REFERENCES

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

[2] Jean-Marc Alliot, Hervé Gruber, and Marc Schoenauer, 'Using genetic algorithms for solving ATC conflicts', in *Proceedings of the Ninth IEEE Conference on Artificial Intelligence Application*. IEEE, (1993).

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

[4] S. C. Dafermos, 'An extended traffic assignment model with application to two-way traffic', *Transportation Research*, (1970).

[5] Daniel Delahaye, Jean-Marc Alliot, Marc Schoenauer, and Jean-Loup Farges, 'Genetic algorithms for partitioning airspace', in *Proceedings of the Tenth IEEE Conference on Artificial Intelligence Application*. IEEE, (1994).

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

[7] Lester Ingber and Bruce Rosen, 'Genetic algorithm and very fast simulated reannealing: a comparison', *Mathematical and Computer Modeling*, **16**(1), 87–100, (1992).

[8] 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).

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

[10] M. Papageorgiou, *Concise encyclopedia of traffic and transportation systems*, Pergamon Press, 1991.

[11] P. L. Tuan, H. S. Procter, and G. J. Couluris, 'Advanced productivity analysis methods for air traffic control operations', FAA Report RD-76-164, Stanford Research Institute, Menlo Park CA 94025, (December 1976).