Flights Alternative Routes Generator by Genetic Algorithms

Sofiane Oussedik
Centre de Mathématiques
Appliquées (CMAP)
Ecole Polytechnique
91128 Palaiseau Cedex, France
oussedik@cmapx.polytechynique.fr

Daniel Delahaye
Centre d'Etudes de la Navigation
Aérienne / CMAP
7, Avenue Edouard Belin
31055 Toulouse, France
delahaye@cmapx.polytechnique.fr

Marc Schoenauer
Centre de Mathématiques
Appliquées (CMAP)
Ecole Polytechnique
91128 Palaiseau Cedex, France
marc@cmapx.polytechynique.fr

Abstract- This paper present a new Air Traffic routes generator based on Genetic Algorithms. Due to the traffic growth, direct (and near direct) routes are more and more congested and there is a real need for spreading the traffic on new alternative routes. Those routes have to be different from several operational criteria and must not generate too much extra-distance compared with the direct route.

To reach this goal, a GA has been implemented with an efficient sharing which automatically allows the emergence of different alternative routes.

This algorithm has been tried on the French airspace and gives realistic operational results.

Keywords : Routes Generator, Genetic Algorithms, Sharing, Air Traffic management.

1 Introduction

On June 1999, France reaches its most loaded day of traffic (for 99) with more than 8000 flights. This huge traffic induced a strong workload on air traffic sectors (which are managed by the air traffic controllers) and generated a lot of delays (aircraft are delayed on the ground to avoid the overload sectors – ground delay program [11]). Due to the traffic increase, this principle has reached its limit because Ground Delay Programs may produce large ground delay in order to adapt the demand to the actual capacity (for instance some aircraft may receive 2, 3 or more hours of delay which is too much for airlines).

In order to extend this principle, the demand may be spread in both time and space dimension. Different efforts have been investigated (see [12, 14]) to add more alternative choices, but all of these approaches need realistic alternative route sets. Depending on the operational objective, those routes must differ according to two different criteria that are :

- sectors crossings
- and, route geometrical structure

Furthermore, those routes must respect some operational constraints:

- the heading at each way-point must stay in a limited cone;
- the extra-distance of a route (compared with the direct route) is limited;
- some three-dimensional airspace zones (sectors or military zones) have to be avoided;

The present paper addresses this problem with the help of genetic algorithms for which an efficient sharing has been implemented.

The second section presents some previous works related to alternative routes generation. The problem modeling is given in the third part. The fourth part describes how the GA has been implemented and finally the fifth part presents some results on the French airspace.

2 Previous related works

Different methods have been developed to generate the minimum cost route between two nodes in a graph with link costs (for instance, see [4]). From a mono-path algorithm, it is possible to generate several random alternative routes by adding random noise on the link costs and by applying the method for each draw. This method has been used in a GA but was not able to manage heading constraints [3].

Extension of those algorithms have been proposed ([2]) to identify the K_minimun distance paths between two nodes in a graph with link costs but those methods have the same drawbacks as in the mono-path algorithms.

Another approach, consists in observing the different routes used by the traffic for one OD during a long time period. This method generates realistic routes but it is limited by the current traffic and then it is very poor in terms of the number of obtained routes, also it do not take into account the distance between routes.

All the previous methods do not match the operational needs and a description of our method is now given.

3 Problem modeling

For each chosen Origine-Destination pair, some alternative routes are generated. Among the French airspace beacons, only a sub-set is concerned with one OD. As a matter of fact, a route $O.B.D$ with only one beacon B may already generate a large extra-distance; it is then possible to remove (pre–processing) the beacon B from the set of possible beacons. A way to build this sub-set of possible beacons is to open an ellipse that is centered at the middle of the segment $\vec{OD}$ (with a given exentricity) and to keep the beacons which are inside

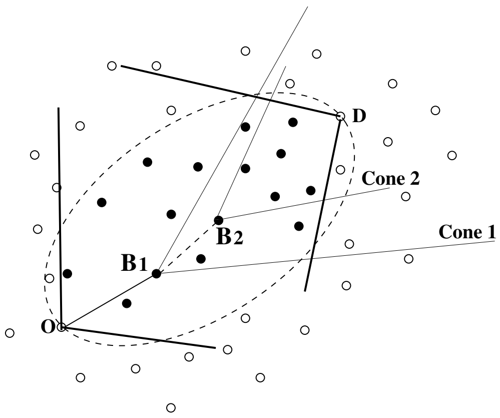

Figure 1: Construction of the beacons sub-set

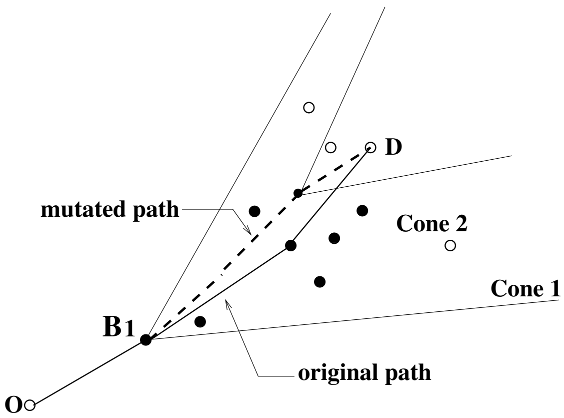

Figure 2: Mutation

this ellipse (see figure 1).

A greedy process is then used to construct some routes in this subset. From the origin O, a beacon is randomly selected from the beacon sub-set (beacon B_1 on figure1). A cone (cone 1) is then opened from this beacon (its axis being colinear to the vector $\vec{OB_1}$) and enables to build a new beacons sub-set. A new beacon is then draw from this subset (B_2) and the process is repeated until the destination is reached. The distance between B_2 and D must be lower than $B_1.D$ distance. It must be noticed that when the beacon subset is empty, the route is completed straightly to the destination D. This generation process produces routes that fit the heading constraint.

The other constraints (extra-distance, zone avoidance) are taken into account by penalty in the objective function:

$$f(R) = length(R) + \alpha(\delta_R^S)$$

where R is the route (list of beacons), δ_R^S is the Kronecker symbol which is equal to 1 if a part of the route R is in a forbidden zone S and α is a penalty constant.

Difference between routes is managed by the GA sharing which is described in the following part.

4 Genetic Algorithms

Genetic Algorithms (GAs) are probabilistic search algorithms. Given an optimization problem they try to find an optimal solution. GAs start by initializing a set (population) containing a selection of encoded points of the search space (individuals). By decoding the individual and determining its cost, the fitness of an individual can be determined, which is used to distinguish between better and worse individuals. A GA iteratively tries to improve the average fitness of a population by construction of new populations. A new population consists of individuals (children) built from the old population (parents) by the use of re-combination operators. Better (above average) individuals have higher probability to be selected for re-combination than other individuals (survival of the fittest). After some criterion is met, the algorithm returns the best individuals of the population.

Theoretical foundations of GA, applications, properties and convergence can be found in [7, 8, 6, 10, 9, 5, 13]. However, By contrast to the theoretical foundations, GAs have to deal with limited population sizes and a limited number of generations. This limitation can lead to premature convergence, which means that the algorithm gets stuck at local optima. A lot of research has been undertaken to overcome premature convergence. Also, experiments have shown that incorporation of problem specific knowledge generally improve GAs. In this paper, attention will be paid on how specific ATM information (heading, sectors crossing constraints, ... have been incorporated in GAs.

5 GA implementation

5.1 Coding

The coding of a route is represented by a list of beacons. The initial population is obtained by using the greedy route construction process described in the problem modeling part.

Example : $0\ B2\ B7\ B4\ D$ is a route from O to D overflying the beacons $B2$, $B7$ and $B4$.

5.2 Operators

The mutation and crossover are defined here :

Mutation : The mutation operator enables to change a sub-list of beacons that belongs to a route and processes the following way:

- a route is selected with a probability P_m ;
- a beacon that belongs to the route is randomly chosen (B_1 one figure 2);
- the beacons that follows the selected beacon are then all removed ;
- the regular construction process is then re-conducted to build a new route from B_1 to D.

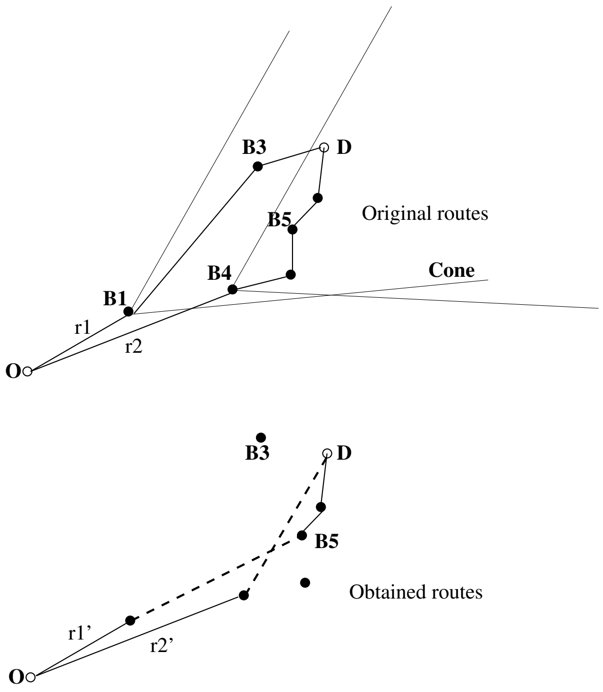

Figure 3: Crossover

Crossover : The crossover operator works the following way :

- two parent routes r_1 and r_2 are selected with a probability P_c.
- for each parent route, the following process is applied :
  - a beacon that belongs to the route is randomly chosen (beacon $B1$ on figure 3);
  - the beacons that follows the selected one are all removed;
  - the regular construction process is then reconducted to build a new route from B_1 to D with the difference that the subset of beacons do not come from all the possible beacons but only from the other parent route.

5.3 Sharing

The Sharing idea is that the GA perception of the fitness function is changed in such a way that when individuals tend to concentrate around a high peak, the fitness there is reduced by a factor proportional to the number of individuals in the region. This has the effect of diminishing the attractiveness of the peak and allowing parts of the population to concentrate on other regions.

This operator is the key success of our algorithm because it automatically generates different alternative routes (individuals) with regard to their geometric aspect and to the sectors they cross. In order to implement a sharing, a distance between two routes (individuals) has to be defined:

$D(r_1, r_2) = (1 + \alpha * \delta) * S(r_1, r_2)$.

where:

- $S(r1, r2)$ is the area between the routes r_1 and r_2 (see figure 4).
- δ is the number of sectors which are not crossed by both routes.
- α is a weighted factor that is used in order to make the algorithm run only on geometrical difference, only on sector difference or both.

The associated complexity of a regular sharing is given by N^2 where N is the population size. For our problem, we used an adaptive $N * \log(N)$ developed by Alliot [1] and based on the work of Yin and Germay [15]. It works on individual clusters for which the number of pools is adaptively tuned by the population performance.

This sharing is working on the following way :

A new cluster is created from an individual if:

- it is enough distant from the other cluster barycenters (average individuals positions in the cluster) else the individual is included in the closest cluster (see figure5).
- the number of cluster is smaller than the population size

On another side, a cluster may be removed if:

- its barycenter is too close to another barycenter (the two clusters are then merged);

It must be noticed that an extended elitism is applied in addition to this sharing by keeping the best individuals of all generated cluster that have their best local individuals at less

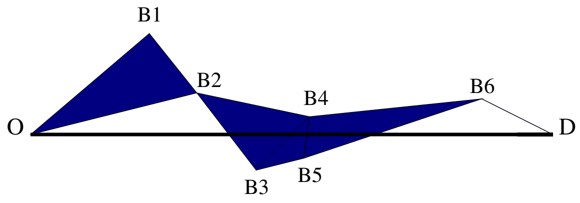

Figure 4: Area between two routes

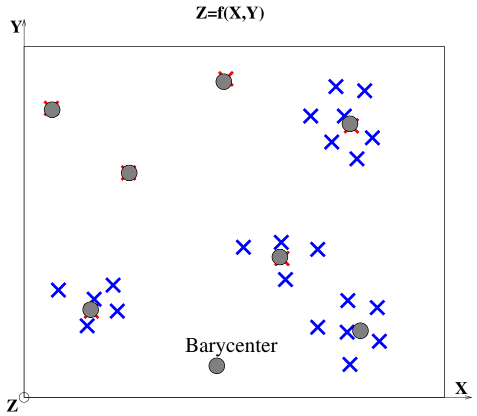

Figure 5: Clustering $O(N \star log(N))$

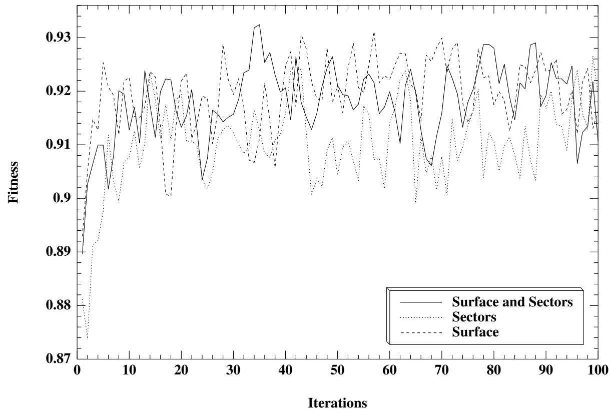

Figure 6: Average Fitness (area, (area and sectors) and sectors differentiation)

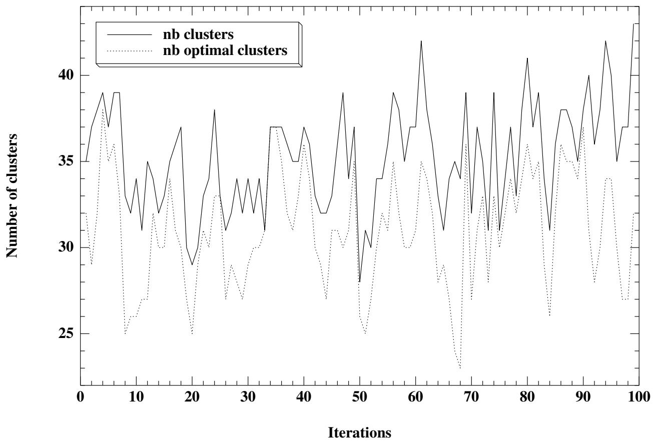

Figure 7: Number of clusters

at $\eta\%$ from the best fitness in the population.

To guarantee a good differentiation between the routes, a minimum distance between an individual and a cluster center is set. If the individual is too close to the cluster barycenter (the distance is smaller than d_{min}), it must be on the cluster. So, it's an adaptive clustering as explained before, but with a bound that lets a clear differentiation between the routes.

d_{min} is set to :

$$d_{min} = \alpha * d^2$$

where :

d^2 is the square of the distance between O and D.

α tunes the area differentiation we need.

In case of sectors differentiation the minimum distance required is represented by a one sector difference.

6 Results

The experiments were based on a data set that involves the French airspace beacons (1150 beacons) and sectors. They were performed on a Pentium Pro 200Mhz Computer.

Some examples of the elementary sectors names are $LFBDNL$, $LFBDC1$, $LFBDTG$ where LF stands for France ; BD for Bordeaux ; and NL, $C1$ TG are the identifiers of three elementary sectors. Each elementary sector being defined by a polygon and the upper and lower flight level bounds. Due to these bounds, and depending on the desired cruise flight level, the avoided sectors can or not be taken into account and ploted (see figures 9 and 10).

Several tests have been performed in order to estimate the influence of some parameters such as the population size (200, 100, 50) and the mutation and crossover probabilities (mutation: 0.0, 0.4, 0.8, 0.9; crossover: 0.0, 0.2, 0.9).

The figure 8 shows the evolution of the best fitness for 10 different GA strategies.

The Best solution that induce the smallest distance route is the strategy with 200 individuals, 200 iterations, mutation 0.8, crossover 0.1 and a differentiation that was only uses the area between two routes.

Also, This area sharing differentiation enables a natural appearance of shortest routes.

The strategy that combines the sectors and area differentiation is often the second best and the sectors differentiation one is at the third position (see figure 6) with regard to the average length of the routes in the population (longer route when the constraints are harder - sector differentiation).

The heading change constraints which gives more or less flexibility on the routes construction is very important as it can be seen on the lowest curve (0.74 on figure 8) where the cone angle is the smallest.

A larger population size (See figure 8) helps on getting good routes on a shorter number of iterations, the tests using 100 and 50 individuals being at a lower level than the other tests, they evolves slower and join the tests that uses 200 individuals after approximately 10 iterations (50 individuals) and 80 iterations for the test with 100 individuals.

The importance of the mutation operator is represented by the fact that the best route that was found by using crossover alone is only at 0.86. However the use of the mutation operator alone gives very good results (0.91) where the shortest routes are often equivalent to the ones found using a combination of crossover and mutation with a mutation probability between 0.5 and 0.8 and a crossover probability included between 0.1 and 0.4.

The quality of our results don't depends only on the shortest route we found but also on reaching our first goal (as previously specified) that is to obtain a set of different routes, that's why the number of clusters and especially optimal ones is very important. The figure 7 represents the evolution of the clusters number and the evolution of the optimal ones with a low optimality at 0.5. This means that at each GA iteration, the routes that are considered as optimal are each best cluster route with a length that is at most two times bigger than the best route length. The figure 13 shows the final alternative

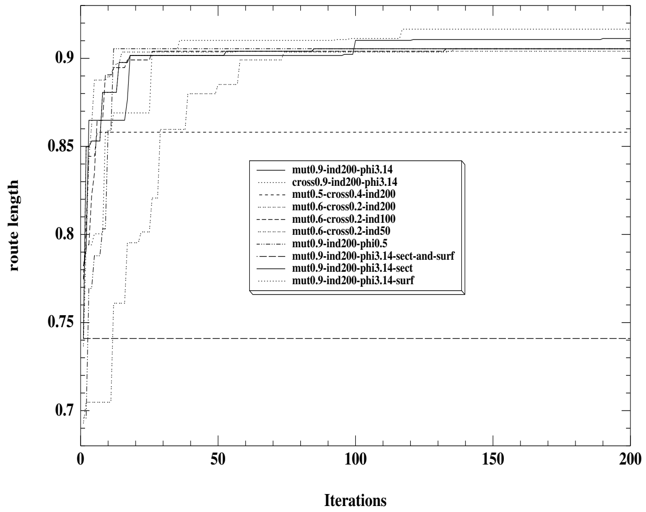

Figure 8: Best Fitness

routes set that was obtained from Toulouse to Paris (LFBO to LFPG) using a $d_{min} = 0.5 * d^2$ area differentiation.

We can also add a penalization term in the fitness (multiplying the fitness by the number of obtained beacons in the route (individual)) to reach a desired number of beacons. Then, the shape of the obtained routes is shown on the figures 11 and 12.

On all the tests, the computation times were of about 5 to 6 minutes depending of the chosen parameters, the GA ones and the others like taking an ellipse or not and the areas computing metric. A “good ” routes set that respect the constraints is fastly reached. A parallelized GA will surely increase the time performances, however this dynamic performance is not needed due to the aim of the algorithm that is designed to provide a database of routes while the network is not changing that fast on operations.

7 Conclusion

In this paper, a Genetic Algorithm formulation of the static alternative routes generation problem has been introduced. Our objective was to produce a realistic model to help airlines and ATC system on their routes choices. The resulting software generates a set of alternative routes that differ from several point of view (geometrical metrics, crossed sectors or both) with reasonable extra distance compared with the direct route (with the minimum distance). It also produces routes that avoid some congested sectors or restricted areas.

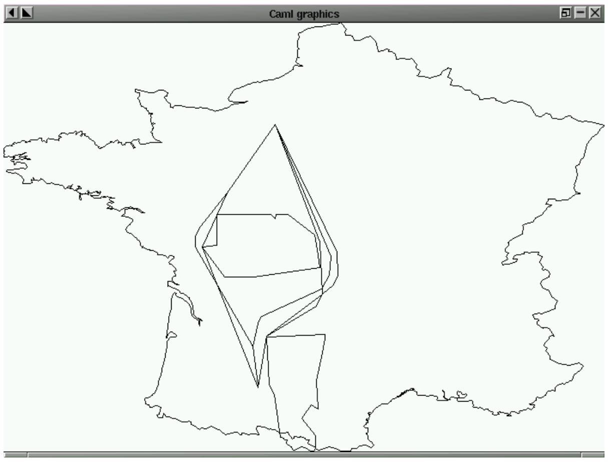

Figure 9: Avoid NL and TG

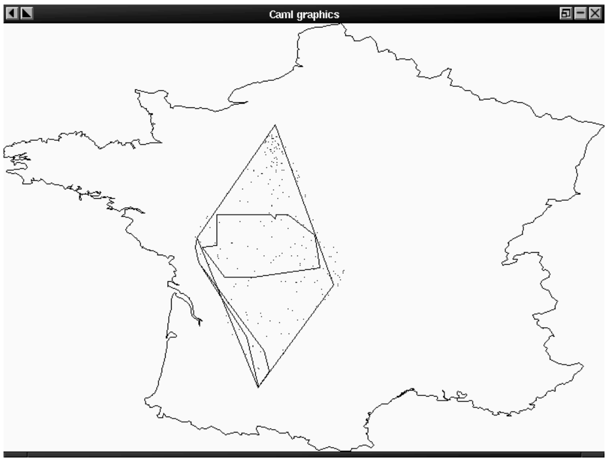

Figure 10: Avoiding C1

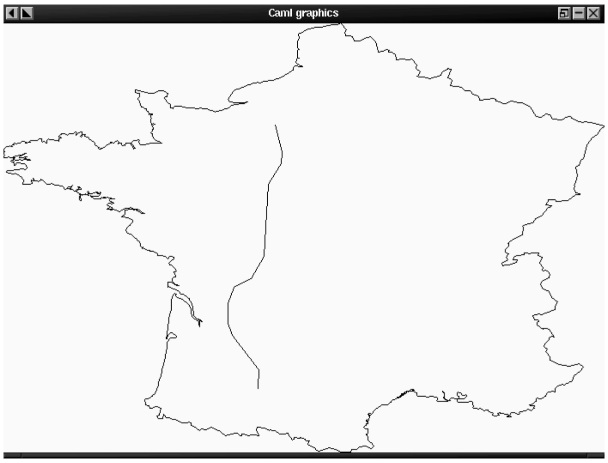

Figure 11: more beacons on a route

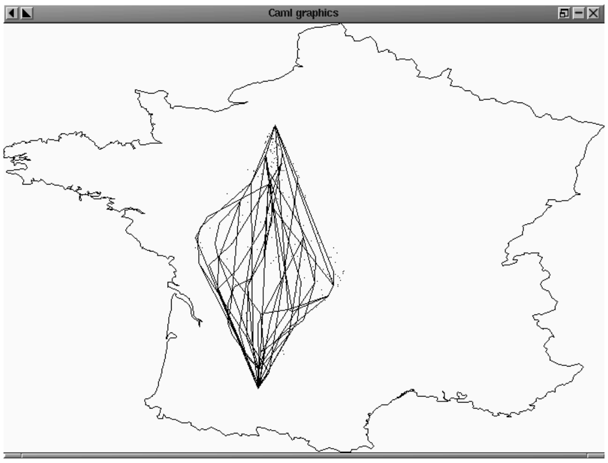

Figure 12: more beacons

The main gain of GA for this problem is the sharing operator which split the population into different clusters, each one (the best individual of the cluster) being associated to a potential alternative route.

The success key of this process is associated to the distance definition which must be closely related to our route differentiation objective.

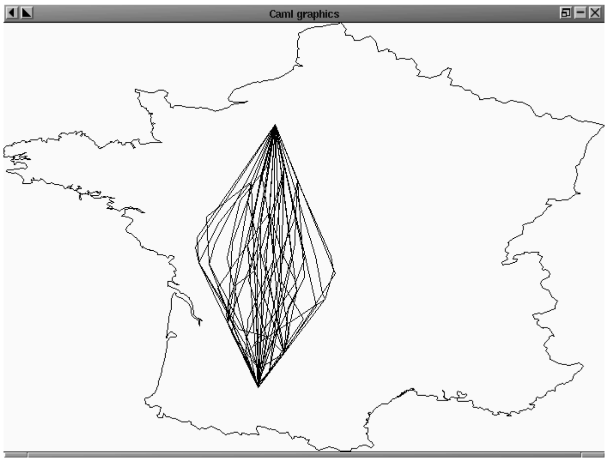

Figure 13: LFBO to LFPG

Throughout the paper, a realistic GA formulation has been presented which encompass a number of operational issues such as the route length, the sector crossing restrictions, the military zones (or others) crossing restrictions, the heading restrictions and the cruise flight level.

Real instances of the problem involving 1150 beacons have been presented and the given results are quite realistic from the Air Traffic Operations point of view.

Bibliography

[1] J.M Alliot. *Techniques d'optimisation stochastique appliquées aux problèmes du trafic aérien*. PhD thesis, 1996.

[2] J. A. Azevedo, J. J. E. R. Silvestre Madeira, E. Q. V. Martins, and F. M. A. Pires. A computational improvement for a shortest paths ranking algorithm. *European J. Operational Research*, 73:188–191, 1994.

[3] D Delahaye, J.M Alliot, M Schoenauer, and J.L Farges. Genetic algorithms for air traffic assignment. In *Proceedings of the European Conference on Artificial Intelligence*. ECAI, 1994.

[4] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numeriche Mathematics*, 1:269–271, 1959.

[5] D.B Fogel. *Evolutionary Computation. Toward a new Philosophy of Machine Intelligence*. IEEE press, 1994.

[6] L.J Fogel, A.J Owens, and M.J Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley and sons. NY, 1966.

[7] D.E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA Addison Wesley, 1989.

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

[9] J.R Koza. *Genetic Programming*. MIT press, 1992.

[10] Z Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992.

[11] A.R Odoni. The flow management problem in air traffic control. In A.R Odoni et al, editor, *Flow Control of Cogested Networks*, volume F38 of *ASI Series*, pages 269–288. NATO, 1987.

[12] S Oussedik and D Delahaye. Reducing air traffic congestion by genetic algorithms. In *Proceedings of the Fifth International Conference on Parallel Problems Solving from Nature*, 1998.

[13] H.P Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.

[14] V Tosic and al. A model to solve en route air traffic flow management problem : a temporal and spatial cases. In *Proceeding of the Air Traffic Management Seminar*. Eurocontrol - FAA, 1997.

[15] X Yin and N Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Proceedings of the Artificial Neural Nets and Genetic Algorithms*, 1993.