

# Airspace congestion smoothing by multi-objective genetic algorithm

Daniel Delahaye, Sofiane Oussedik, Stéphane Puechmorel

► **To cite this version:**

Daniel Delahaye, Sofiane Oussedik, Stéphane Puechmorel. Airspace congestion smoothing by multi-objective genetic algorithm. SAC 2005, 20th Annual ACM Symposium on Applied Computing, Mar 2005, Santa Fe, United States. pp 907-912, 10.1145/1066677.1066887 . hal-01004145

**HAL Id: hal-01004145**

**<https://hal-enac.archives-ouvertes.fr/hal-01004145>**

Submitted on 11 Jul 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Airspace Congestion Smoothing by Multi-objective Genetic Algorithm

Delahaye Daniel CENA 7, Ave Ed Belin 31055 Toulouse France (33)562 174 179 delahaye@tls.cena.fr	Sofiane Oussedik ILOG 3-5 Ave Galliéni 94257 Gentilly France (33)149 083 604 soussedik@ilog.fr	Puechmorel Stephane ENAC 7, Ave Ed Belin 31055 Toulouse France (33)562 174 152 puechmor@recherche.enac.fr
---	--	---

## ABSTRACT

Air traffic control systems become more and more congested due to the increase of demand. One way to reduce this congestion is to modify the flight plans (slot of departure and route of aircraft) in order to adapt the demand to the available capacity. This paper addresses the general time-route assignment problem which can be stated as follows : **one has to find an optimal time of departure and an optimal route for all the aircraft involved in the considered airspace, in order to minimize the associated congestion and the induced delay** . This problem is a multi-objective NP\_Hard problem. We perform our research on the application of multi-objective stochastic methods on real traffic data without using the flow network concept, but by simulating the flight of each aircraft. The first results show that our approach is able to reduce congestion of the French airspace by a factor 2.

## 1. INTRODUCTION

Airspace congestion is due to aircraft which are located in the same area during the same period of time. When a flight plan is deposit, aircraft has to follow a legal route and has to be assigned to a slot (time period of the day). Congestion being related with aircraft located at the same place during the same period of time, it can be reduced by moving aircraft in time (slot allocation) or in space (route allocation). This global assignment is called bi-allocation problem. This bi-allocation has a price which is related with the amount of delay produced by the slot allocation and the extra-distance (and delay) due to the route changes. This bi-allocation problem is then a multi-objective problem for which congestion has to be reduced with the minimum amount of delay. Historically, arrival slot allocation is the most current way to avoid congestion in the terminal area of airport. When

an airport is plan to undergo congestion due to an exceed of demand, the involved aircraft are delayed at the departure airports in order to avoid them to be delayed in the sky at the terminal area. This principle has been extended for the en-route sectors (sectors between terminal areas) and insures that sector demands are always under the capacity. Those operational concepts are based on human operator (for instance the slot allocation is given by controllers who are located at the CFMU (Central Flow Management Unit)) and could be enhanced by the use of optimization techniques. The first part of this paper describes the previous related works trying to use mathematical model in order to optimize the route and slot allocations. The second part gives the details of the associated mathematical modeling and explains why multi-objective genetic algorithms are needed to solve this problem. The third part describes the multi-objective genetic algorithm principle and the fourth part presents how they have been implemented to solve our problem. Finally the fifth part gives the result produced by our method for a real instance of the problem (one day of traffic over the French airspace).

## 2. PREVIOUS RELATED WORKS

Traffic assignment techniques have been developed in a way to reduce congestion in transportation networks by spreading the traffic demand in time and space. Historically, those techniques have been applied to road traffic assignment due to the strong level of congestion encountered in this domain. Those techniques try to find an optimal route, or an optimal time of departure, or both, for each individual in a way to reach a dynamic system-equilibrium. This equilibrium have been introduced by Wardrop en 1952[14] and is a target which represents the best assignment of the demand to a transportation network. The complexity induced by the dynamic traffic assignment is strong, especially when route and time of departure are simultaneously optimized. This problem is NP\_HARD [1] and may have several optima [8]. Due to this complexity, the problem is always partially solved for simplified instances :[3]. In the same time, specific approaches have been developed to solve this route-time allocation problem (space-time network : [15]; variational inequality : [11];optimal control : [9];simulation :[4];dynamic programming (Ground Holding Problem) :[2, 10]). All the previous approaches solve partially the bi-allocation prob-

lem and propose solutions for slot allocation only, route allocation only or both for very small instances of the problem. Because of the strong complexity associated to the discrete bi-allocation problem, new mathematical models have been developed in order to find approximate solutions. These models use stochastic optimization techniques and address large instances of the problem within a reasonable computer time [6, 12]. Those approaches are mono-objective and try to minimize the congestion in the sector. The present approach is the multi-objective extension of the work presented in the paper [12] for which the delay minimization has been included straightly in the objectives. The following part will then remind the model used in the paper [12] and will present how the delay minimization has been included to build a multi-objective genetic algorithm.

### 3. MATHEMATICAL MODEL

Congestion in the airspace is due to aircraft which have close positions in a four dimensional space (one time dimension and 3 space dimensions). It is then relevant to investigate ways to separate aircraft in this 4 dimensional space by changing their slots of departure (time separation) or by changing their routes (spatial separation) or both. Those changes must be done in a way that takes into account the objectives of the airlines :

- the moving of the slot of departure must be done in a limited domain (otherwise, for instance, some aircraft will be forced to take off at 2:00AM to reduce the congestion of the day but at this time there will be no passenger to carry);
- the new slot of departure must take into account the connections between flights (as a matter of fact some aircraft have to wait the arrival of some previous flights to take off (hub phenomenon));
- the possible routes must not generate too large additional distances.

So, for each flight, a new pair (slot of departure, route) will be chosen from two discrete and finite sets :

- a set of possible slots of departure (around the original slot of departure) ;
- a set of routes which do not increase too much the total path length and are approved by the airline company the flight belongs to.

After taking off, the aircraft will follow its flight path and will generate congestion in the different sectors encountered and also at the arrival airport when arriving. According to the controllers themselves, the workload induced in a control sector is a function of the three main following criteria :

- the conflict workload that results from the different actions of the controller to solve conflicts.
- the coordination workload related to the information exchanges between a controller and the controllers in charge of the bordering sectors or between a controller and the pilots when an aircraft crosses a sector boundary;
- the monitoring aims at checking the different trajectories of the aircraft in a sector and induces a workload.

As it has been noticed before, slot allocation has a price which can be measured by the amount of delay produced by the new flight planning. This objective can be measured by the number of minutes of delay given to the aircraft. When an aircraft is assigned to a new route, it has to fly an extra-distance which produces a delay at the destination airport. This delay has a higher price compared to the ground hold delay; one minute of such a delay is equivalent to three minutes of delay on the ground [7].

A pair of decision variable  $(\delta_i, r_i)$  is associated with each flight in which  $\delta_i$  is the advance or the delay from the original slot of departure and  $r_i$  is the new route. With this notation,  $(0, r_0)$  will be considered as the most preferred choice from the user point of view. Those two decision variables  $(\delta_i, r_i)$  will be chosen from two finite-discrete sets :  $\Delta$  for the slots and  $R$  for the routes. More precisely the structures of  $\Delta$  and  $R$  are the following :

$$\begin{aligned} \Delta &= -\delta_m, -\delta_m + 1, \dots, -1, 0, 1, \dots, \delta_p - 1, \delta_p \\ R &= r_0, r_1, r_2, \dots, r_{max} \end{aligned} \quad (1)$$

For which  $\delta_m, \delta_p$  are respectively the maximum advance and the maximum delay permitted for a flight. Those limits can be different for each flight. The routes are ordered according to the induced cost for the associated flight. So  $r_0$  is the best one and  $r_{max}$  the worst.

When a flight has to be connected to some arriving flights, its slot of departure must be later than the time of arrival of the previous flights and separated from them by a minimum amount of time ( $\tau$ ). So, when a slot of departure is changed, one must first check that this new schedule matches the connecting constraint. This will ensure that the random point generated in the state domain by the stochastic optimization process will always satisfy the connection constraint. The first objective of our problem consists in the **congestion minimization**. This objective is defined in the following way : “one must try to reduce congestion in the most overloaded sectors” ; this will spread the congestion over several sectors. So, we have :

$$y = \min \sum_{k=1}^{k=P} \left( \left( \sum_{t \in T} \widetilde{W}_{S_k}^t \right)^\phi \times \left( \max_{t \in T} \widetilde{W}_{S_k}^t \right)^\varphi \right) \quad (2)$$

where :

- $\sum_{t \in T} \widetilde{W}_{S_k}^t$  : is the sector  $S_k$  smoothed<sup>1</sup> congestion surface computed during the day.
- $\max_{t \in T} \widetilde{W}_{S_k}^t$  : is the maximum sector smoothed congestion reported during the day.
- $P$  is the number of elementary sectors.
- $\phi$  and  $\varphi$  are weight factors

The congestion reduction objective of each individual will be normalized into an objective maximization ( $y_1$ ) by defining the ratio of the congestion associated with the initial planning (*ref* : before the changes) and the distribution given

<sup>1</sup>This smoothing is done by the mean of a moving time window which averages the congestion over  $\frac{D}{2}$  slots in the past and  $\frac{D}{2}$  slots in the future

by the chromosome (*chrom* :after applying the changes of the flight plans) :

$$y_1(chrom) = \frac{y(ref)}{y(chrom)} \quad (3)$$

So, when  $y_1(chrom) > 1$ , it means that the induced congestion is lower than the reference one.

The second objective which has been considered is the **delay and extra-distance minimization**. When an aircraft  $i$  is supposed to take off during the slot  $t_k$  and receives a regulation and takes off during the slot  $t_n$  induce delay is given by  $\delta_s(i) = t_n - t_k$ . If the slot allocation is optimized for the long term schedule, it is possible to produce negative slot in order to better reduce the congestion, then the regular time of departure of the aircraft is modified and the induce "delay" could be given by  $\delta_s = |t_n - t_k|$ . After taking off, aircraft can use a longer route which induces a longer fly time. This extra flying time costs about three times more than a delay on the ground :  $\delta_r(i) = 3 * (T_r - T_0)$  where  $T_r$  is flying time of the actual route and  $T_0$  the flying time of the shortest route. So, the total delay cost for a flight  $i$  is given by  $\delta(i) = \delta_s(i) + \delta_r(i)$ . In order to insure the equity<sup>2</sup> between aircraft, the second objective is built by the mean of a quadratic summation of delays instead on a regular linear one :

$$y_2 = \sum_{i=1}^N \delta(i)^2 \quad (4)$$

This problem is a strong NP\_HARD multi-objective problem[12] with non-separable state variables. So, one must solve a multi-modal multi-objective combinatorial optimization problem in a huge space with non-separable decision variables. Currently, only multi-objective genetic algorithm stochastic optimization is well adapted to address this kind of problem.

#### 4. MULTI-OBJECTIVE GENETIC ALGORITHM

Multi-objective optimization can be defined as the problem of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions represent a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term "optimize" means finding such a solution which would give the values of all the objective functions acceptable to the designer. Genetic algorithms (GAs) [13] are problem solving systems based on principles of evolution and heredity. A GA maintains a population of individuals which represents potential solutions to the problem at hand and is implemented as some (possibly complex) data structures. Each solution  $x_i$  is evaluated to give some measures of fitness in order to apply the selection process. The genetic operators are applied to produce new solutions. There are unary transformations (mutation type), which create new individuals by a

<sup>2</sup>For the same among of delay, say 150, the linear objective  $\sum \delta(i)$  is independent of the distribution (150+0+0=50+100+0=50+50+50) but the quadratic form,  $\sum \delta(i)^2$  is minimum when the distribution of delays is balanced (150<sup>2</sup> = 22500 > 50<sup>2</sup> + 100<sup>2</sup> = 12500 > 50<sup>2</sup> + 50<sup>2</sup> + 50<sup>2</sup> = 7500)

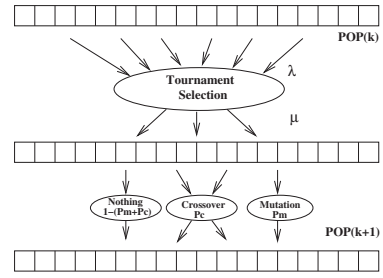


Figure 1: Classical Genetic Algorithm with Tournament Selection

small change of a single individual and higher order transformations (crossover type), which create new individuals by combining parts from several (two or more) individuals. The structure of our genetic algorithms is summarized on figure 1. A genetic algorithm works on a full ordered fitness in order to apply a selection operator. Having several objective, a map between those criteria and the scalar fitness has to be found. This mapping is given by the Pareto rank computation of the non dominated solution ranking algorithm [5]. This algorithm works on a set of points of the state domain for which the associated objectives are known. From this initial set, the algorithm extracts the solution which are not dominated<sup>3</sup> by any other. The first step consists in creating two subsets related with the domination property : dominated and non-dominated. To do so, each element of the original set is compared with the other ones until it is found one which dominates it, so this element is put into the dominated subset. If no so, it is put in the non-dominated subset. The first non-dominated subset is then call front 1. Afterwards, the process is repeated on the dominated subset and the front 2 subset is created. This process ends when the dominated subset is empty. The figure 2 describes the process and the figure 3 shows the induce classification in the objective space. Each element of the original set is then labelled with the associated Pareto front which produce a classification between individuals in the objective space. The fitness of an individual is then computed by using a decreasing mapping of the rank number. So, the individual from the first rank will have a higher fitness than those from the rank two and so on. The function that has been used in our experiment is the following :

$$Fitness(i) = \exp(1 - Rank(i)) \quad (5)$$

<sup>3</sup>Suppose two objective vectors  $\vec{A}$  and  $\vec{B}$  are given. A The solution  $\vec{A}$  is said to dominate the other solution  $\vec{B}$ , if both following conditions are true :

1. The solution  $\vec{A}$  is no worse than  $\vec{B}$  in all objectives ;
2. The solution  $\vec{A}$  is strictly better than  $\vec{B}$  in at least one objective.

A solution  $\vec{x}^*$  is said to be Pareto optimal if there exist no feasible vector  $\vec{x}$  which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Pareto optimum almost always gives multiple solutions called non-inferior or non-dominated solutions. The set of all such solutions which are non-dominated constitutes the Pareto front. These solutions are in the boundary of the design region, or in the locus of the tangent points of the objective functions.

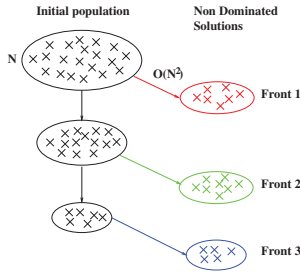


Figure 2: Domination ranking algorithm

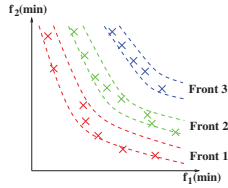


Figure 3: Front ranking

When selection is used straightly on such a fitness, it can be noticed that individuals converge toward the best Pareto front but they are not spread on it. In order to avoid this miss-spreading, a sharing in the objective space has been included. This sharing is computed for each front and must insure that the worst individual from the previous front (in the increasing order) will be always better than the current one from the shared fitness point of view. The sharing fitness is then given by :

$$fit_{sh}(k) = fit(k) - \left( \frac{m_i - 1}{N_k} \right) d_k \cdot 0.99 \quad (6)$$

where  $m_i = \sum_{j=1}^{N_k} Sh[d(\vec{y}_i, \vec{y}_j)]$  where  $N_k$  is the number of individuals of the Pareto front of rank  $k$  and  $m_i$  is the objective space sharing function of the individual  $i$  which measures the local level of aggregation around individuals with :

$$Sh(d) = \begin{cases} 1 - \left( \frac{d}{\sigma_{share}} \right)^\alpha & \text{if } d < \sigma_{share} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where  $d_k = fit(k) - fit(k+1)$ . For our experiments the value a delay being much larger than the values of the congestion reduction, a mathematical affinity has been included in the objective distance measure in order to have a final isotropic distance metric. This affinity produces a new distance metric given by :

$$d(\vec{y}_i, \vec{y}_j) = \sqrt{\frac{(y_{j1} - y_{i1})^2}{n_c^2} + \frac{(y_{j2} - y_{i2})^2}{n_d^2}} \quad (8)$$

where  $n_c$  and  $n_d$  are weight factors. Finally,  $\sigma_{share}$  has been adjusted at 5% of the maximum observed distance from the origin in the objective space (after correction with the affinity).

## 5. APPLICATION TO AIRSPACE CONGESTION

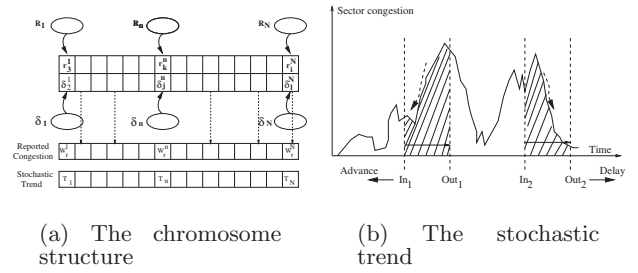


Figure 4: Special coding and stochastic problem specific knowledge

To make run this optimization process, a set of flights with their original slots of departure and their original routes has to be generated. Each flight is supposed to have also a set of alternative slots and a set of alternative routes depending of their own constraints. From those initial data, an air traffic simulator generates a sample of the trajectories for all the original routes and for all the alternative routes. Those simulations are done before the optimization process is started. Based on those simulation samples, it is possible to compute the induced congestion for any route/slot combination and the associated delays also.

### 5.1 Data Coding

A straight forward coding has been used in the sense that each chromosome is built as a matrix (see fig. 4-(a)) which gather together the new slot movements (for the times of departure) and the new route numbers (for the flight paths). The initial population is generated by random trials of such matrices. The coding has been improved by including problem knowledge in the chromosome. To be able to identify the aircraft involved in the biggest sector congestion, new information has been added into the chromosome which indicates for each gene, the maximum level of sector congestion encountered during a flight. When an aircraft is involved in a congestion peak, and when its time of departure is changed, the following conclusions can be given (see figure(see fig. 4-(b)) :

- if the aircraft enters a congestion zone when the associated congestion level is increasing, a congestion reduction can be expected by advancing the entering of such aircraft.
- if the aircraft enters a congestion zone when the associated congestion level is decreasing, a congestion reduction can be expected by delaying the entering of such aircraft.
- else pure random move of the time of departure is applied.

Based on those conclusions, it is possible to compute a stochastic trend for each flight knowing the congestion encountered during the flight. Those two indicators (the level of encountered congestion and the stochastic trend) have been added to the chromosome and will be used by the recombination operators in order to speed up the evolution process.

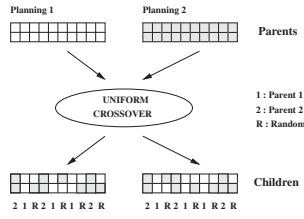


Figure 5: Crossover operator

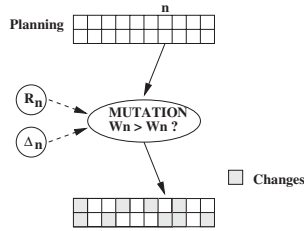


Figure 6: Mutation operator

## 5.2 Crossover and mutation

The successive steps of the crossover operator are the following (see figure 5) :

- two parents are first selected ;
- the summation of the sector congestion levels is computed for each flight in both parents. For a flight  $n$ , total congestion level in the parent  $p$  will be noted  $W_n^p$  ;
- an order relationship is then built with the total congestion level in the following way :
  - flight planning  $n$  in parent 1 is said to be “much better” than flight planning  $n$  in parent 2 if  $W_n^1 < \delta.W_n^2$ ; where  $\delta \in [0.7, 0.95]$ ;
  - flight planning  $n$  in parent 2 is said to be “much better” than flight planning  $n$  in parent 1 if  $W_n^2 < \delta.W_n^1$ ;
  - flight planning  $n$  in parent 1 and in parent 2 are said to be “equivalent” if none of the previous relations matches;
- if a flight planning “is much better” in the first parent than in the second then it is copied in the second ;
- if a flight planning “is much better” in the second parent than in the first then it is copied in the first ;
- if the two flight planning “are equivalent” they are randomly exchanged with a fixed probability (0.5) ;

In order to create new genes in the chromosomes, a mutation operator has to be developed (see figure 6). This operator affects more often the flights involved in the highest peaks of congestion, and also determines whether it is “more suitable” to delay or advance a flight (see fig.4-(b)). So to compute the *stochastic trend* over all the sectors, the signed indicator  $T_n \in [-1, 1]$  is computed; it is a kind of bias to advance or delay each flight.  $T_n$  is a signed pondered (by the encountered flight congestion) summation over sectors. The

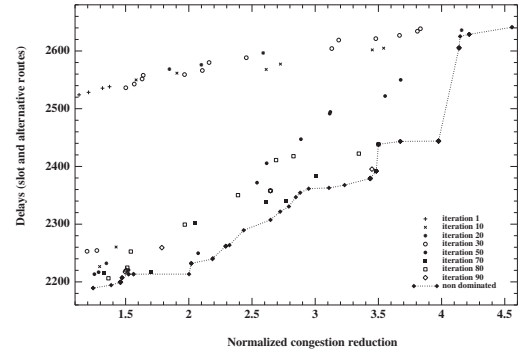


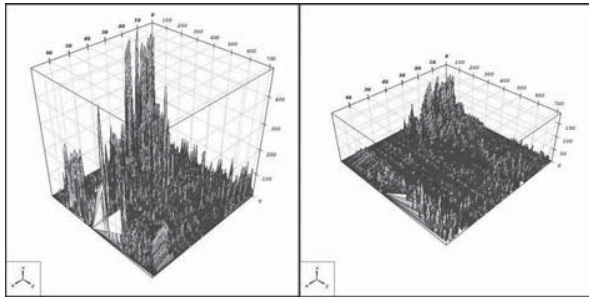
Figure 7: Pareto front of the 1th June 1996 ( $\sqrt{y_2} = f(y_1)$ . The vertical axis ( $y_2$ ) has been presented with a square root in order to have shorter numbers)

sign indicates the sector state during the entry of the flight (congestion increase or decrease). The mutation operator works in the following way :

- a threshold congestion level ( $W_h$ ) is randomly chosen in  $[0, W_{max}]$  (where  $W_{max}$  is the maximum level of congestion observed for the current generation) ; then for each flight  $n$  in the chromosome the following is applied :
- if ( $W_n > W_h$ ) then the associated flight plan is modified :
  - if  $T_n > \text{rand}(1)$  then a random future slot is assigned to the flight.
  - if  $T_n < -\text{rand}(1)$  then a random past slot is assigned to the flight.
  - otherwise we randomly affect the flight slot with no preference for the advance or the delay.
- else the flight planning is unchanged;

## 6. RESULTS ON A REAL DAY OF TRAFFIC

The computations were based on a whole day of real traffic data including 6381 flights that cross the French airspace. The number of elementary sectors was 89. We consider also that congestion of an elementary sector  $S_k$  at time period  $t$  is equal to the congestion of the sectors grouping  $R_{S_K}$  to whom it belongs ( $\bar{W}_{S_k}^t = \bar{W}_{R_{S_k}}^t$ ) during the same period (grouping is applied to address low traffic periods). By this, we take into account the changes in the critical capacities values during the day. Also, the critical capacity of the prohibited sectors (as military sectors) is set equal to 0. The tests are performed with the elitism principle and have been processed on a Pentium Pro 1Gmhz computer and need about 6 hours computing. The parameters for our algorithm are : the smoothing window  $D = 5min$  ; the population length  $poplength = 50$ ;  $dt = 1min$  so,  $T = 1440$  minutes for the day ;  $\phi = 0.9$  and  $\varphi = 0.1$ . The number of generations : 300 ; and the maximum slots moving in the future or in the past : 45 minutes. The result of the algorithms is presented on the figure 7. This figure shows the Pareto front evolution with generations. The first front



(a) Before GA

(b) After GA

**Figure 8: Before and after GA**

(generation 1) is located on the upper left corner and the last one is symbolized with dash lines. Each dot is a potential planning of our 6381 flights (slot and route allocation). At the end of the evolution, the algorithm produce several solutions with different features from the objective point of view. For instance the extreme solutions of the front are  $(y_1, \sqrt{y_2}) = (4.58, 2640)$  and  $(1.24, 2190)$  and represent respectively a congestion reduction of 4.58 and 1.24 (compared with the initial planning) and a total amount of delay of 210886 and 174939 minutes. This means that if congestion has to be divided by 3.58, it will cost 210886 minutes of delay. It can be also noticed that the sharing principle works well because the solutions are well spread on the fronts. A good compromise would be the solution located at  $:3.96, 2442$  which induce a real cumulated delay of 195069 minutes. The associated congestion reduction profile for this solution associated to the most congested sector is given on figure 8. The vertical axis gives the absolute congestion in the sectors, the second axis (0 – 700) is the time of the days (2 minutes step) and the last one gives the sector numbers (only the congested sectors). It can be noticed that the congestion has been strongly reduced.

## 7. CONCLUSION

This paper addresses the general problem of the reduction of the air traffic congestion by optimizing the time of departure and the route of aircraft. To that end, Multi-Objective GAs have been used with problem knowledge recombination operators which really improve the performances of the algorithm. The model takes into account the connection constraint and produces realistic solutions which could be implemented for real traffic. The multi-objective optimization gives several realistic solutions which could be analyzed by experts. Those solutions are targets which reduce the congestion in a significant way. The next step is to find a principle which “forces” the demand moving toward those targets by changing the time of departure and the route of aircraft. This means that a congestion pricing has to be developed in order to keep equity between flights which choose the shorter routes or the best slots and the ones which choose a longer route or a worse slot (they have to pay less so they will decide to do such a choice).

## 8. REFERENCES

- [1] M. Ben-Akiva, A. DePalma, and I. Kaysi. Dynamic network models and driver information systems. *Transportation Research*, 25A(5):251–266, 1991.
- [2] D. Bertsimas and S. Stock. The air traffic flow management problem with en-route capacities. Technical report, A.P Sloan School of Management. M.I.T, 1994.
- [3] M. Carey. Optimal time-varying flows on congested networks. *Operations Research*, 35(1):58–69, 1987.
- [4] E. Cascetta. Static and dynamic models of stochastic assignment to transportation networks. In *Flow Control of Congested Networks*, 1987.
- [5] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, New York, 2002.
- [6] D. Delahaye and A. Odoni. Airspace congestion smoothing by stochastic optimization. In *Proceedings of the Evolutionary Programming Conference*. Natural Selection inc., 1997.
- [7] S. Elefante. A strategic and tactical tool. In *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, 1998.
- [8] M. Ghali and M. Smith. A model for the dynamic system optimum traffic assignment. *Transportation Research*, 29B(3):155–170, 1995.
- [9] B. Janson and J. Robles. Dynamic traffic assignment with arrival time cost. In *Transportation and Traffic Theory*, 1993.
- [10] L. Maugis. Mathematical programming for the air traffic flow management problem with en-route capacities. In *Proceeding of the 1996 IFORS Conference*. IFOR, 1996.
- [11] A. Nagurney. *NETWORK ECONOMICS A Variational Inequality Approach*, volume 1 of *Advances in Computational Economics*. Kluwer Academic, 1993.
- [12] S. Oussedik and D. Delahaye. Reduction of air traffic congestion by genetic algorithms. In *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*, 1998.
- [13] H. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
- [14] J. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers. Part II*, pages 325–378, 1952.
- [15] S. Zenios. Network based models for air traffic control. *European Journal of Operational Research*, 50:166–178, 1991.