



# Airspace congestion smoothing by stochastic optimization

Daniel Delahaye, Amedeo Odoni

## ► To cite this version:

Daniel Delahaye, Amedeo Odoni. Airspace congestion smoothing by stochastic optimization. EP 97, 6th International Conference on Evolutionary Programming, Apr 1997, Indianapolis, United States. pp 163-176, 10.1007/BFb0014809 . hal-00937702

**HAL Id: hal-00937702**

**<https://enac.hal.science/hal-00937702>**

Submitted on 18 Apr 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 Stochastic Optimization

Daniel Delahaye      Amedeo Odoni  
C.E.N.A\*/M.I.T      M.I.T<sup>†</sup>

August 15 1996

## Abstract

*This paper addresses the general time-route assignment problem :*

*One considers an air transportation network in a 2 dimensional space with asymmetric non-separable link cost and a fleet of aircraft with their associated route and slot of departure. For each flight a set of alternative routes and a set of possible slots of departure are defined. One must find "optimal" route and slot allocation for each aircraft in a way that significantly reduces the peak of workload in the most congested sectors and in the most congested airports, during one day of traffic.*

*A state of the art of the existing methods shows that this problem is usually partially treated and the whole problem remains unsolved due to the complexity induced. Genetic algorithm are then presented and adapted to the problem. New recombination operators are described and really improve the quality of the given solutions. Results on a large instance of the problem are then given and show how well genetic algorithm can manage this kind of combinatorial optimization problem.*

**Key Words:** Genetic Algorithm, Airspace Congestion, Time of Departure, Route, Network

## 1 Introduction

When joining two airports, an aircraft must follow routes and beacons ; these beacons are necessary for pilots to know their position during navigation and because of the small number of beacons on the ground they often represent crossing points of different airways.

Crossing points may generate conflicts between aircraft when their trajectories converge on it at the same time and induce a risk of collision.

At the dawn of civil aviation, pilots solved conflicts themselves because they always flew in good weather conditions (good visibility) with low speed aircraft. On the other hand, modern jet aircraft do not enable pilots to solve 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 many aircraft simultaneously present in the sky, a single controller is not able to manage all of them. Then, airspace is partitioned into different sectors, each of them being assigned to a controller.

Before taking off, when a flight is decided, the pilot has to file a Flight Plan (FPL) which describe the main steps of the flight (slot of departure, flight path, speeds, altitudes etc...), this FPL is necessary for the air traffic control system to anticipate the flight monitoring. Usually, when there are few aircraft in the airspace the flight plan is well respected.

As any human being, a controller has working limits, and when the number of aircraft increases, some parts of the airspace reach this limit and become congested. Two kinds of congestion can be identified according to the part of airspace involved : terminal congestion (around airports) and en route congestion (between airports). In the past, the first way to reduce these congestions was to modify the structure of the airspace in a way that increases the capacity (increasing the number of runways, increasing the number of sectors by reducing their size). This has a limit due to the cost involved by new runways and the way to manage traffic in too small sectors (a controller needs a minimum amount of airspace to be able to solve conflicts). The other way to reduce congestion is to modify the flight plans in a way to adapt the demand to the available capacity. As a matter of fact, when we have a look at the congestion in the airspace one can remark that the peaks of congestion happen approximately at the same times during the day (one peak in the morning and one peak in the evening). The current time scheduling is usually

---

\*Centre d'Etudes de la Navigation Aérienne

<sup>†</sup>Massachusetts Institute of Technology

fixed by the airline companies without taking into account the ATC constraints. Then congestion is expected to be reduced by moving (in a limited domain) the time of departure of aircraft (in the past and in the future) and by changing the current flight paths (without too much extradistance).

This paper is focused on these flight plan modifications and shows how stochastic optimization is able to manage this kind of problem. In a first part, a description of the previous related works is given and in a second one, a simplified model is developed and a mathematical formulation of our problem is given. In the third part a description of stochastic optimization is given and particularly of a genetic algorithm. The application of the algorithm is given on the fourth part and finally an improvement is given in the fifth part.

## 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 in space. Historically, those techniques have been applied to road traffic assignment due to the strong level of congestion encountered in this domain. The problem of users of a congested transportation network seeking to determine travel paths of minimal cost from origins to their respective destinations is a classical network problem. An equilibrium occurs when the number of trips between an origin and a destination equals the associated travel demand. Wardrop [49] stated the traffic equilibrium conditions through two principles :

- **First Principle** : The journey times of all routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.
- **Second Principle** : The average journey time is minimal.

From those two principles, Dafermos and Sparrow [12] coined the terms *user-optimized* and *system-optimized* transportation networks to distinguish between two distinct situations in which users act unilaterally, in their own self-interest, in selecting their routes, and in which users select routes according to what is optimal from the societal point of view, in that the total costs in the system are minimized. More detailed descriptions of those equilibria can be found in the following references : [21, 19, 44].

Improved traffic assignment algorithms, take into account that demand can be influenced by the induced cost but still with a static demand over time [24, 25, 11, 37].

By definition all those techniques, are applied to static traffic demand and are mainly used to optimize traffic on a long time period and can only capture the macroscopic events.

When a more precise matching between traffic demand and capacity has to be found, microscopic events have to be taken into account, and dynamic traffic assignment techniques have to be used ([40] gives a good description of those techniques). 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 user-equilibrium or a dynamic system-equilibrium. 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 [16, 5] and may have several optima [26].

Even when only the route decision is investigated, this problem does not have an exact solution. Due to this complexity, the problem is always partially solved for simplified instances [3, 30, 33, 34, 35].

In the same time, specific approaches have been developed to solve this route-time allocation problem. The main ones are the following :

- Space-time network [51, 17, 29, 32] ;
- Variational Inequality [7, 22, 6] ;
- Optimal Control [39, 50, 23, 41, 8] ;
- Simulation [10, 47, 4, 9] ;
- Dynamic Programming [38, 45, 43, 2, 42, 48, 1, 46, 27]

From the traffic assignment point of view, the problem we have to solve could be summarized in the following way :

**a fleet of aircraft is considered for which one must find the optimal route and the optimal time of departure in a way to reach a “system-equilibrium”**

All the previous approaches are not able to manage the whole problem due to the complexity induced. In the following, a model is proposed and a method is developed in a way to give “very good” solutions for realistic instances of the whole problem.

## 3 A Simplified Model

### 3.1 Introduction

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 those aircraft in this 4 dimensional space by changing their slot of departure (time separation) or by changing their route (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 passengers to carry);
- the new slot of departure must take into account the connexions between flight (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 workload in the different sectors encountered and also a congestion 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 corresponds to the information exchanges between a controller and the controller in charge of the bordering sector 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.

From the airport point of view the congestion is defined by the number of aircraft unable to land due to the runways use (so it depends on the number of aircraft waiting in the sky). This airport congestion is dependent on the available capacity which changes over the day according to the weather. A static capacity has been attributed to each airport for our experiments, but the model is able to manage dynamic airport capacity, as well.

If one sector is considered, one can observe variations of the associated workload during the day according to the traffic in it. So, a typical profile of the sector control workload is very chaotic with different peaks correlated with the peaks of traffic in it (usually one peak in the morning and one peak in the evening).

It must be noticed that the peaks do not appear at the same time in the different sectors due to the flow propagations.

According to our definition of workload, only the information related with the route and the slot of departure are relevant from the flight plan. As a matter of fact, this information enables us to compute the time when an aircraft enters (or exits) a sector and when it is over a crossing of two airways (point of conflict generation). When examining the physical air traffic network, we notice that airways are superpositions of several flight routes which have the same projection on the ground but different altitudes according to their azimuth (semi circular rule<sup>1</sup>). So an airway can be modeled by a bidirectional link which combines several individual aircraft routes. Then, without loss of generality, the real 3-dimensional transportation network will be modeled by a classical 2-dimensional network on a horizontal plan.

We can now define our goals more precisely in the following way :

one considers an air traffic transportation network in a 2 dimensional space and a fleet of aircraft with their associated route and slot of departure. For each flight a set of alternative routes and a set of possible slots of departure are defined. One must find “optimal” route and slot allocation for each aircraft in a way that significantly reduces the peak of workload in the most congested sectors and in the most congested airports, during one day of traffic.

This “optimal” bi-allocation (route-time) must take into account the objectives of the airlines.

### 3.2 Mathematical formulation

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 :

$$\Delta = -\delta_m, -\delta_m + 1, \dots, -1, 0, 1, \dots, \delta_p - 1, \delta_p \quad R = r_0, r_1, r_2, \dots, r_{max}$$

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 cost induced to the associated flight. So  $r_0$  is the best one and  $r_{max}$  the worst.

To be able to compute the control workload (according to our definition), the route of an aircraft will be defined by the two following lists :

---

<sup>1</sup>Aircraft with heading between 0 and 180 have to fly at odd altitudes (in thousands of feet) and even altitude for headings between 180 and 360

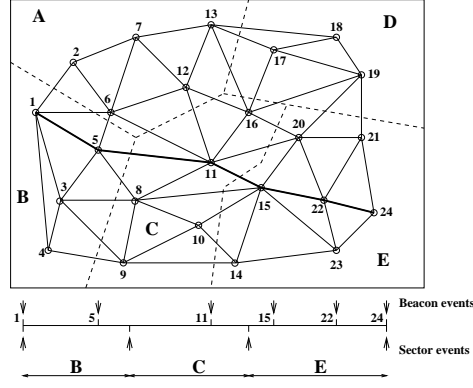


Figure 1: Simplified Flight Plan

- list of the overflown beacons in the air network with the associated passing time :

$$L_b = (b_1, t_1), (b_2, t_2), \dots, (b_i, t_i), \dots$$

- list of crossed sectors with the associated in-out times :

$$L_s = (S_1, Tin_1, Tout_1), (S_2, Tin_2, Tout_2), \dots, (S_k, Tin_k, Tout_k), \dots$$

An example of a simplified flight plan is given in figure 1. In this case an aircraft is joining airports 1 and 24. It overflies 5 beacons (5,11,15,22) and crosses 3 sectors.

Given this information, it is now possible to compute the three components of the control workload (monitoring, coordinations, conflicts) in each sector of the airspace. To reach this goal, one first begins to compute the exiting flow on each link of the network and the crossing flow over each sector. With the definition of our simplified flight plan, it is easy to compute the discrete event when an aircraft exits a link of the network or when an aircraft enters or exits a sector. The flows are then computed by using a moving window in which an average of the number of events is computed (the temporal size of the window is a parameter of the model). So the flow  $f_{ij}(t)$  exiting a link  $(i, j)$  at time  $t$  is computed in the following way :

$$f_{ij}(t) = \frac{1}{2.D + 1} \sum_{s=t-D}^{s=t+D} \sum_{n=1}^{n=N} \delta_{ijn}^t$$

where  $D$  is the half extension of the temporal window (in slot) and  $\delta_{ijn}(t)$  is the kroneker symbol which is equal to 1 if the flight  $n$  exits the link  $(i, j)$  during the slot  $t$  ( $N$  is the whole number of flight).

Given the flows on links, it is now possible to give a mathematical formulation of the control workload induced in a sector. As it has been previously said, workload in a sector  $S_k$  at time  $t$  can be roughly expressed by the summation of three terms :

$$W_{S_k}^t = Wmo_{S_k}(t) + Wco_{S_k}(t) + Wcf_{S_k}(t);$$

- $Wmo_{S_k}(t)$  Monitoring workload (related to the number of aircraft in a sector) ;
- $Wco_{S_k}(t)$  Coordination workload (related to the flows cut by the sector boundaries) ;
- $Wcf_{S_k}(t)$  Conflict workload (depends on the crossing flows at each node in the sector).

### 3.2.1 Formulation of the objective function

With the previous definition of the control workload , when a set of decision variables has been chosen, it is now straightforward to compute the dynamic workload profile over time for each sector and to memorize the associated maxima. We define our objective in the following way : “ one must try to reduce congestion in the most overloaded sectors but not necessarily simultaneously in all the sectors” ; this will spread the congestion over several sectors. So, we have :

$$obj = \min \sum_{k=1}^{k=K} \left( \max_{t \in T} W_{S_k}(t) \right)$$

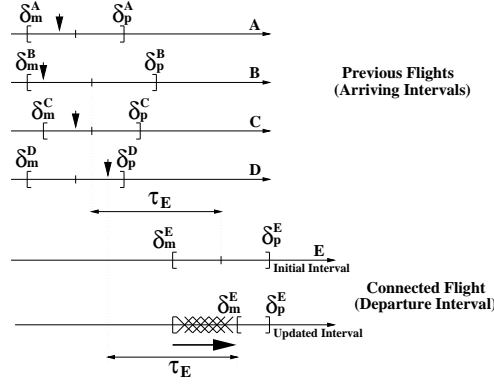


Figure 2: Example of connexion

### 3.3 Connected flights constraint

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 flight and separated from it by a minimum amount of time ( $\tau$ ). So, when a slot of departure is changed, one must first check that this new scheduling respects the connecting constraint. To reach this goal, the slot moving set of the connected flight must take into account the slot moving sets of the previous flights as it can be seen on figure 2.

On this figure, the flight “E” is connected to 4 previous flights :”A,B,C,D”. To always be able to assign a slot to a flight, the slot moving sets must satisfy the following properties :

- Maximum boundary

$$\delta_p^E > \max_{k \in \{A, B, C, D\}} (\delta_p^k + \tau_E)$$

This checking must be done by using the longest possible route for each previous flights  $A, B, C, D$ .

- Minimum Boundary

$$\delta_m^E > \max_{k \in \{A, B, C, D\}} (t_a^k + \tau_E)$$

where  $t_a^k$  is the actual arrival time, determined by the slot moving and routes assigned to the flight  $A, B, C, D$ .

So, before assigning a moving slot to a connected flight, one must first check that the previous associated flights have already been assigned a slot and a route from their own route and moving slots sets. This will ensure that the random point generated in the state domain will always satisfy the connexion constraint.

### 3.4 Problem complexity

Before investigating an optimization method, the associated complexity of our problem must be studied. The model previously developed is discrete and induces a high combinatoric in a huge space. As a matter of fact, if  $R_n, \Delta_n$  are the route set and the slot moving set associated with flight  $n$ , the number of possible choice ( $C_n$ ) for this flight is :

$$C_n = |R_n| \cdot |\Delta_n|$$

where  $|S|$  denotes the cardinality of the set  $S$ .

Then the number of possible decision variable combinaisons is given by :

$$|State| = \prod_{n=1}^{n=N} C_n$$

where  $N$  is the total number of flights. If the number of choices is the same for each flight ( $|R_n| = C^{te} = r$  and  $|\Delta_n| = C^{te} = \delta$ ), then the cardinality of the state domain becomes :

$$|State| = (r \cdot \delta)^N$$

For instance, for 20000 flights with 10 route choices and 10 possible slot movings, :  $100^{20000}$ . Moreover, those decision variables are not independent due to the connexion induced by the control workload, the airport congestions and the connexion constraint, so, decomposition methods cannot be applied.

It must be noticed that the objective function :

- is not continuous (then it is not convex) ;
- may have several equivalent optima (the objective is then said multimodal) ;

This problem is then a strong NP\_HARD problem with non-separable state variables.

### 3.5 Conclusion

The microscopic model previously developed takes into account the major features of the real problem with the associated constraints and is able to accommodate specific individual preferences. This model has been completed by a macroscopic description of the sector workload which induces a robust computation of sector congestion.

So, one must solve a multimodal combinatorial optimization problem in a huge space with non-separable decision variables. Currently, only stochastic optimization is well adapted to address this kind of problem.

## 4 Stochastic Optimization

### 4.1 Introduction

Stochastic optimization techniques usually addressed problem with strong complexity for which the objective function has no specific feature such as convexity, continuity etc.... The main characteristic of those stochastic optimization techniques is to randomly move in the state domain in a way to improve the objective criterium. At the beginning, local search algorithms randomly moved in the state domain toward the closest optimum without being able to avoid local optima. As a matter of fact, this technique consist in first selecting a random point in the state domain, second to find a neighbor point and to compare the associated objectives : if the neighbor is better it begins the current point else one must find another neighbor point till the objective is improved. Afterward, more sophisticated and more powerful methods have been developed in a way that avoid local optima and converge toward global optima (Simulated Annealing and Genetic Algorithm) and which can identified several global optima (Genetic Algorithms).

### 4.2 Genetic Algorithms

Genetic algorithms (GAs) [28, 31, 20, 36] are problem solving systems based on principles of evolution and heredity. A GA maintains a population of individuals,  $P(t) = x_1, x_2, \dots, x_n$  at iteration  $t$ . Each individual represents a potential solution to the problem at hand and is implemented as some (possibly complex) data structure  $S$ . Each solution  $x_i$  is evaluated to give some measure of fitness. Then a new population at iteration  $t + 1$  is formed by selecting the fitter individuals. Some members of the new population undergo transformation by means of genetic operators to form new solutions. There are unary transformations  $m_i$  (mutation type), which create new individuals by a small change of a single individual and higher order transformations  $c_i$  (crossover type), which create new individuals by combining parts from several (two or more) individuals. For example, if parents are represented by a five-dimensional vector  $(a_1, a_2, a_3, a_4, a_5)$  and  $(b_1, b_2, b_3, b_4, b_5)$ , then a slicing crossover of chromosomes after the second gene produces offspring  $(a_1, a_2, b_3, b_4, b_5)$  and  $(b_1, b_2, a_3, a_4, a_5)$ . The control parameters for genetic operators (probability of crossover and mutation) need to be carefully selected to provide better performance. The intuition behind the crossover operation is information exchange between different potential solutions. After some number of generations the program converges - the best individual hopefully represents the optimum solution. In [28], Goldberg presents a good summary of many recent GA applications in biology, computer science, engineering, operations research, physical sciences, and social sciences.

Genetic algorithms use a vocabulary borrowed from natural genetics. We would talk about genes (or bits), individuals (chromosomes or gene strings), and population (of individuals). Populations evolve through generations.

The execution steps of genetic algorithms are the following :

1. Initialize population and evaluate fitness :

To initialize a population, we need first to decide the number of genes for each individual and the total number of chromosomes (*popsize*) in the initial population. All the information contained in a state point must be summarized in the chromosome, this results from the coding which is one of the success keys of genetic algorithms.

2. Reproduction (Selection) :

Reproduction is the selection of individuals with respect to the probability distribution based on the the fitness values. Fitter individuals have better chances of getting selected for reproduction [36]. Different kinds of selection processes can be applied for a specific problem; one of the most simple is the "Roulette Wheel Selection Process" : Each chromosome has a certain number of slots proportional to its fitness value. The selection process is based on spinning the wheel many times; each time we select a single chromosome for a new population. Obviously, some chromosomes will be selected more than once. This is in accordance with genetic inheritance: the best chromosomes get more copies, the average stay even, and the worst die off.

3. Crossover :

Now we are ready to apply the first recombination operator, crossover, to selected individuals. The probability of crossover,  $p_c$ , gives us the expected number  $p_c \cdot \text{popsize}$  of chromosomes which should undergo the crossover operation. After having selected two parents (according to their fitness), we generate a random number  $r$  between 0 and 1; if  $r < p_c$ , then the parents undergo a crossover. We then generate a random number  $pos$  from the range of  $(1..m - 1)$ , where  $m$  is the total number of genes in a chromosome. The number  $pos$  indicates the position of the crossing point. The coupled chromosomes exchange genes at the crossing point as described earlier.

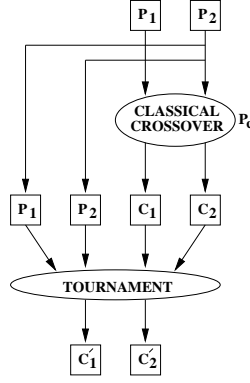


Figure 3: Simulated Annealing Crossover

#### 4. Mutation :

Chromosomes which have not undergone a crossover may be randomly mutated on the gene by gene basis. To perform a mutation, a gene is randomly selected and modified by a problem dependent process in a way to produce new genes. The probability of mutation,  $p_m$ , gives us the expected number of mutated gens  $p_m \cdot m \cdot popsize$ .

#### 5. Convergence :

Following reproduction, crossover, and mutation, the new population is ready for its next generation. The rest of the evolutions are simply cyclic repetitions of the above steps until the system reaches a predetermined number of generations or converges (i.e., no improvement in the overall fitness of the population).

This classical scheme can be improved by using simulated annealing in the crossover operator. This Simulated Annealing Crossover has been developed recently [14] and has been used in different applications with success [13, 15, 18]. It greatly improves the convergence speed of the genetic algorithm for all applications, so it is generic and not limited to a problem dependent improvement. The way to introduce Simulated Annealing in the crossover operator is straightforward. Two parents ( $P_1, P_2$ ) are selected in the population according to their fitness and randomly crossed using the classical crossover operator. If the crossover happened, two children are generated ( $C_1, C_2$ ). The first child ( $C_1$ ) is then randomly associated with one parent and the second one ( $C_2$ ) with the remaining parent. For each created couple, the “winner” is then selected according to a simulated annealing process in the following way (see figure3) :

$$\begin{aligned} \text{if } f(C) > f(P) &\Rightarrow C' = C \\ \text{if } f(C) < f(P) &\Rightarrow C' = C \text{ with the probability } P(C \rightarrow C') = e^{\frac{f(C) - f(P)}{T(gen)}} \end{aligned}$$

where  $T(gen)$  is the control parameter depending of the current generation  $gen$ .

### 4.3 Conclusion

The strong point of these stochastic optimization methods is their ability to give “very good” solution(s) to large instances of real problems. Usually the given solutions are very robust and very close to the optimum when it is known. They can manage very specific constraints and do not need any properties concerning the objective function.

## 5 Application to our problem

### 5.1 Introduction

As previously noted, Genetic Algorithms need a data coding, specific recombination operators and a fitness evaluation to be able to select best individuals. The way this specific genetic algorithm works is the following. A set of flight plans is generated from each chromosome candidate and the whole associated day of traffic is generated. Sector congestion and airport congestion are registered and the associated fitness is computed. This is an appropriate method to use because the more realistic the simulation of the day of traffic is the more relevant are the results.

The problem specific features of the Genetic Algorithm are now described.

### 5.2 Data Coding

This step consists of converting each point of the state domain into a chromosome used by the genetic algorithm. In our problem, the state variables (which contain all the information needed to compute the sector workload) consist of the set of flight plans. The possible new path and new slot moving have been supposed to be chosen in



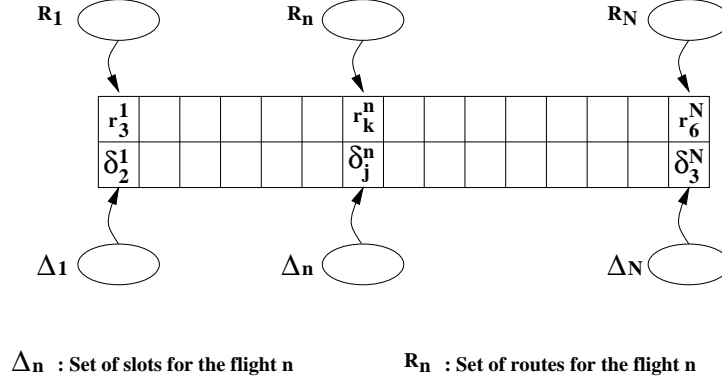


Figure 4: Structure of the Chromosome

two discrete-finite sets associated with each flight. In this case a straightforward coding has been used in the sense that each chromosome is built as a matrix which gather together the new slot moving (for the time of departure) and the new route number (for the flight path) (see figure 4).

With this coding, a population of individuals can be created by randomly choosing a new slot moving number and a new route number from individual sets associated with each flight.

### 5.3 Fitness Evaluation

The previous description of the chromosome and the associated recombination operators can now be used to randomly generate some points in the state domain. To apply the selection operator, a fitness must be associated with each chromosome in a way to evaluate the quality of each individual according to the optimization criterion. In our problem, one first considers the default flight plans and computes the maximum workload in each sector in the airspace.

$$MaxW(ref) = \sum_{k=1}^{k=K} \left( \max_{t \in T} W_{S_k}(t, ref) \right)$$

Afterward, for each generated chromosome, maximum sector workloads are computed the same way :

$$MaxW(chrom) = \sum_{k=1}^{k=K} \left( \max_{t \in T} W_{S_k}(t, chrom) \right)$$

The fitness is then defined by the ratio of the two previous expressions.

$$fitness(chrom) = \frac{MaxW(ref)}{MaxW(chrom)}$$

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

### 5.4 Recombinason Operators

To be able to recognize the aircraft involved in the biggest sector congestion peak or in the biggest airport congestion peak, new information must be added to the chromosome which indicates for each gene, the maximum level of sector congestion encountered during a flight and the level of congestion at the destination airport when the aircraft land. To reach this goal, the biggest peak of congestion is determined over all the sectors and airports, and a number of congestion levels is defined respectively for the airports and the sectors. With these two parameters, and for each slot, a level of congestion can be associated with each sector and each airport. So, after each evaluation, the associated list of sectors of each individual simplified flight plan is read in a way to identify the maximum level of congestion encountered during the flight and at the destination airport. Those congestion levels will be used to improve the recombination operators.

#### 5.4.1 Crossover

The successive steps of this new crossover operator are the following :

- two parents are first selected according to their fitness ;
- the summation of the congestion levels (sector and airport) is computed for each flight in both parents. For a flight  $n$ , total congestion level in the parent  $p$  will be noted  $W_n^p = WS_n^p + WA_n^p$  (with  $WS_n^p$  : maximum level of sector congestion encountered by the flight  $n$  in the parent  $p$ ,  $WA_n^p$  : airport congestion level encountered by the flight  $n$  when arriving at destination (in the parent  $p$ )) ;

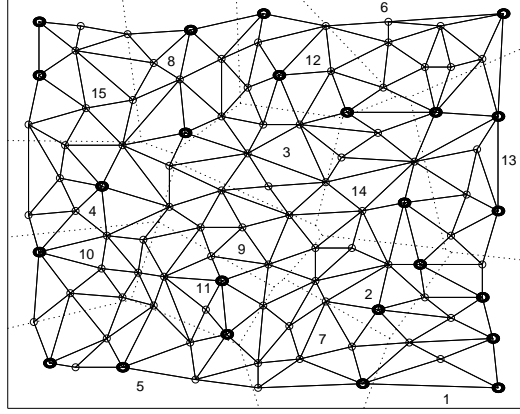


Figure 5: Structure of the test network

- an order relationship is then constructed with the total congestion level in the following way :
  - flight planing  $n$  in parent 1 is said to be “much better” than flight planing  $n$  in parent 2 if  $W_n^1 < \delta \cdot W_n^2$ ; where  $\delta \in [0.7, 0.95]$ ;
  - flight planing  $n$  in parent 2 is said to be “much better” than flight planing  $n$  in parent 1 if  $W_n^2 < \delta \cdot W_n^1$ ;
  - flight planing  $n$  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 second ;
- if the two flight plannings “are equivalent” they are randomly exchanged with a constant probability (0.5) ;

This is done for all the flights in the chromosome.

#### 5.4.2 Mutation Operator

As already noted, this operator only affect the flights involved in the highest peaks of congestion. This operator works in the following way :

- two threshold congestion levels are randomly chosen : one for the sector congestion ( $Th_S$ ) and one for the airport congestion ( $Th_A$ ) ;
- then for each flight  $n$  in the chromosome the following are applied :
- if  $(WA_n > Th_A)$  or  $(WS_n > Th_S)$  then the associated flight plan is modified by randomly choosing either a new route or a new slot moving (but not both) in the associated data set  $(R_n, \Delta_n)$  ;
- else the flight planing is unchanged;

## 6 Application to a simplified network

### 6.1 Introduction

A simplified network has been used to evaluate the performances of the algorithm. This example tries to reproduce the traffic density encountered usually in airspace. The evaluations have been done with flight plans generated in a two dimensional network but the model could be applied to the real flight plans in three dimensional airspace, due to the fact that only flight plan events are used to compute the objective function. In this example, only sector congestion has been taken into account but as it has been mentioned before, airport congestion could have been introduced by using airports with smaller capacity. The features of the network are the followings : 100 nodes, 544 links, 30 origin-destination pairs, 15 sectors, 24 airports, geographical size of 1000x1000 Nm. A physical description is given in figure 5 with the sectoring and the associated sector number. Each airport is symbolized with a larger node.

A hundred flights have been randomly generated for each Origin-Destination pair, so, a total of 3000 flight plans is used for this example (the speed of the aircraft has been fixed at 500 kts). Those flights have been spread over 6 hours and the number of slots has been fixed at 180. This induces a slot duration of 2 minutes. For each Origin-Destination, a set of possible different routes is randomly built in a way that never induces an extradistance greater

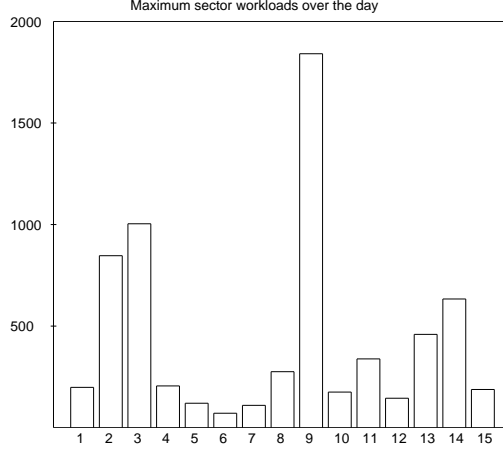


Figure 6: Max workload distributions over sectors

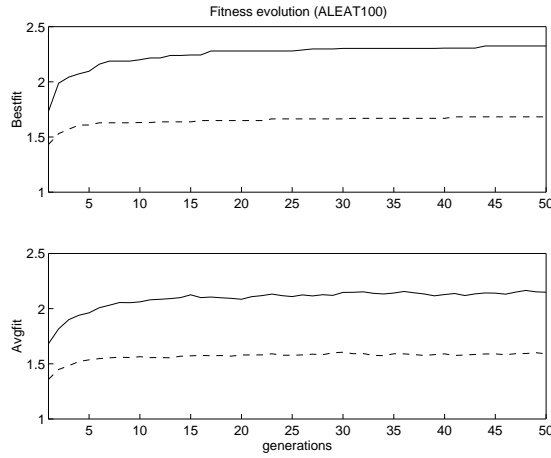


Figure 7: Best and average fitness evolution over generations

than 30 percent of the original distance with a maximum of 10 routes. For each flight the maximum slot moving has been fixed to 15 in the future and 15 in the past from the original slot of departure.

Two kinds of experiments have been done to compare the gain given by the delay allocation only (the departure slot can only be moved in the future (not in the past) and the original route is used) and the time-route allocation (different routes can be used and the moving of the slot of departure can be done in the past or in the future). The first is a kind of Ground Holding Policy(GHP) and the second one is a Full Planning Policy(FPP).

The original plannings induce a control workload in the airspace with a maximum for each sector at a given slot. The workload weight coefficients used for these experiment are :  $\eta = 1.0$ ;  $\beta = 1.0$ ;  $\alpha = 1.0$ . The half extension of the temporal window used to compute the flows from the discrete events has been fixed at  $D = 2$  and induces a window duration of 10 minutes.

The maximum control workload registered in each sector over the six hour duration is given in figure 6 (with this data set).

On this figure, sector indices are indicated on the horizontal axis and the associated maximum sector workload on the vertical axis. One can notice that sector 9 is the most overloaded. For this problem, with the definition of the fitness, the genetic algorithm is expected to reduce those peaks of congestion for the most congested sectors.

## 6.2 Results

The results presented in this section are obtained from runs on a SPARC5 workstation. The large size of the chromosomes (6000 bytes) limits the maximum size of the population used by the genetic algorithm at 90 due to memory limitations (a population of 50 individuals has been used for all the experiments). It must be noticed that this small population size will reduce the possibilities of genetic algorithms, and better results could be reached on a machine with more memory (even if the following results are already very good). The others parameters (probability of crossover, probability of mutation, number of generations) have been fixed by experimentation at  $P_c = 0.6$ ,  $P_m = 0.2$  and NB.gen=100. Given all these parameters, each run lasts about 15 minutes on a SPARC5 workstation. This running time is not critical because the problem solved has not any real time constraints (the improved time-route planning could be applied for several months). The associated fitness evolution is given in figure 7.

On this figure, the two kinds of policies (GHP and FPP) can be identified by the different lines used (plain for

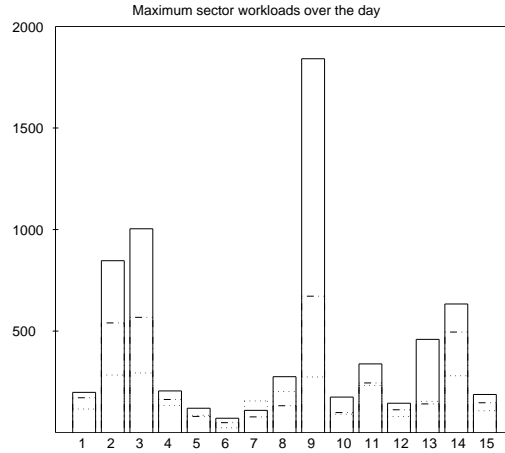


Figure 8: Max workload over sectors for the two associated policies (-. GHP ... FPP)

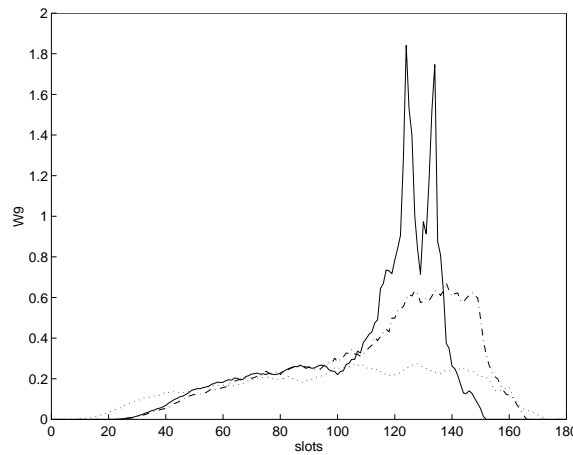


Figure 9: Sector 9 control workload evolution over the day (-. GHP ... FPP)

FPP and dashed for GHP). As expected, the FPP policy gives better results than the GHP policy with respectively a best fitness of 2.32(FPP) and 1.68(GHP). These results are relevant due to the fact that the FPP policy has more degrees of freedom. It means that original “congestion” in the sense given by the optimization criterion, could be respectively divided by 2.32 and 1.68 with the new flight plans. Even with the small population size used, the results given by the genetic algorithm are very encouraging. The associated max workload distributions over sectors is given on figure 8.

On this figure, it can be noticed that the max workload on the most overloaded sector (number 9) has been divided by 2.73 ( $\frac{1841}{672}$ ) for the GHP policy and by 6.71 ( $\frac{1841}{274}$ ) for the FPP policy. A more detailed description of the sector 9 control workload evolutions is given in figures 9 (on these diagrams workload has been divided by 1000 for presentation).

As expected, the workload is spread around the peak as in a smoothing process. The previous curves make us notice that the algorithm only affects the most congested sectors without modifying the less loaded ones. This is due to the definition of the criterion which takes into account only the most loaded sectors.

## 7 Conclusion

This difficult problem has been handled well by the proposed genetic algorithm even with a small population (50 individuals) and a small number of generations (100). Even better results could be expected with larger populations. A mathematical model has been developed to formulate this bi-allocation (route and slot of departure) problem in a way that could be adapted to the real flight plans in three dimensional airspace. So, the next step, is to try this optimization technique on a real traffic sample. The strength of this model is its ability to manage the constraint of the airline companies in a microscopic way by using individual sets of decision variables associated with each flight. Using those individual decision variables, macroscopic models of the sector control workload and of airport congestion have been developed by computing the associated flows with a time moving window on the strategic points (crossings, sector boundaries, airports). This reduces the sensitivity of the “optimum” given by the genetic algorithm with the proposed solution. This model can also take into account the connexion between flights (hub phenomenon) by changing the decisions variable sets of the following flight. This model can also use dynamic capacities for both

airports and sectors.

A fast simulation process has been used to compute the associated sector workload and the airport congestion to improve the performance of the genetic algorithms.

A realistic example has been treated and two policies have been tested : GHP (delay decision only) and FPP (route and slot of departure decision). As expected the second one gives much better results due to the larger state domain induced by the two extra degrees of freedom of the FPP policy. Two kinds of genetic algorithms have been tested : the first one has a classical crossover(slicing) and mutation( random gene) operators and the second one focuses its recombination actions on the most congested sectors by assigning two levels of congestion for each flight(one for the most congested sector encountered during the flight and one for the congestion of the destination airport when the flight is arriving). This new information greatly improves the quality of the given solutions. It must be noticed, that only aircraft involved in the most congested parts of the airspace (sector or airport) receive a flight plan modification. A FPL change cost could be added to classify aircraft and identified the one which must be modified first.

## References

- [1] G Andreatta, A.R Odoni, and O Richetta. Models for the ground holding problem. In L Bianco and A.R Odoni, editors, *Large Scale Computation and Information Processing in Air Traffic Control*, Transportation Analysis, pages 125–168. Springer-Verlag, 1993.
- [2] G Andreatta and G Romanin-Jacur. Aircraft flow management under congestion. *Transportation Science*, 21(4), 1987.
- [3] M Ben-Akiva. Dynamic network equilibrium research. *Transportation Research A*, 19A(5):429–431, 1985.
- [4] M Ben-Akiva, A DePalma, and P Kanaroglou. Dynamic model of peak period traffic congestion with elastic arrival rates. *Transportation Science*, 20(2):164–181, 1986.
- [5] M Ben-Akiva, A DePalma, and I Kaysi. Dynamic network models and driver information systems. *Transportation Research*, 25A(5):251–266, 1991.
- [6] D Bernstein, T.L Friesz, R.L Tobin, and B.W Wie. A variational control formulation of the simultaneous route and departure-time choice equilibrium problem. In *Transportation and Traffic Theory*, 1993.
- [7] Wie B.W., R.L Tobin, T.L Friesz, and D Bernstein. A discrete time, nested cost operator approach to the dynamic network user equilibrium problem. *Transportation Science*, 29(1):79–92, 1995.
- [8] G.E Cantarella and A Sforza. Methods for equilibrium network traffic signal setting. In *Flow Control of Congested Networks*, 1987.
- [9] E Cascetta. A stochastic process approach to the analysis of temporal dynamics in transportation networks. *Transportation Research*, 23B(1):1–17, 1989.
- [10] E Cascetta and G.E Cantarella. A day-to-day and within-day dynamic stochastic assignment model. *Transportation Research*, 25A(5):277–291, 1991.
- [11] S Dafermos. Relaxation algorithms for the general asymmetric traffic equilibrium problem. *Transportation Science*, 16(2):231–239, 1982.
- [12] S Dafermos and F.T Sparrow. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards*, 73B:91–118, 1969.
- [13] 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.
- [14] D Delahaye, J.M Alliot, M Schoenauer, and J.L Farges. Genetic algorithms for partitioning airspace. In *Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Application*. CAIA, 1994.
- [15] D Delahaye, J.M Alliot, M Schoenauer, and J.L Farges. Genetic algorithms automatic regrouping of air traffic sectors. In *Proceedings of the Fourth International Conference on Evolutionary Programming*. Natural Selection inc., 1995.
- [16] A DePalma, P Hansen, and M Labbé. Commuters’paths with penalties for early or late arrival time. *Transportation Science*, 24(4):276–286, 1990.
- [17] O Drissi-Kaitouni and A Hameda-Benchekroun. A dynamic traffic assignment model and a solution algorithm. *Transportation Science*, 26(2):119–128, 1992.
- [18] N Durand and Alliot. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the 11th Annual ACM Conference on Applied Computing*, Philadelphia, 1996. (ACM/SAC).

- [19] J.E Fernandez and T.L Friesz. Equilibrium predictions in transportation markets : the state of the ar. *Transportation Research B*, 17B(2):155–172, 1983.
- [20] L.J Fogel, A.J Owens, and M.J Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley and sons. NY, 1966.
- [21] T.L Friesz. Transportation network equilibrium, design and aggregation : Key developements and research opportunities. *Transportation Research*, 19A(5):413–427, 1985.
- [22] T.L Friesz, D Bernstein, T.E Smith, and B.W Wie. A variational inequality formulation of the dynamic network user equilibrium problem. *Operations Research*, 41(1):179–191, 1993.
- [23] T.L Friesz, J Luque, R.L Tobin, and B.W Wie. Dynamic network traffic assignment considered as a continuous time optimal control problem. *Operation Research*, 37(6):893–901, 1989.
- [24] N.H Gartner. Optimal traffic assignment with elastic demands : A review. part i : Analysis framework. *Transportation Science*, 14(2):174–191, 1980.
- [25] N.H Gartner. Optimal traffic assignment with elastic demands : A review. part ii : Algorithmic approaches. *Transportation Science*, 14(2):191–208, 1980.
- [26] M.O Ghali and M.J Smith. A model for the dynamic system optimum traffic assignment. *Transportation Research*, 29B(3):155–170, 1995.
- [27] E.P Gilbo. Airport capacity : Representation, estimation, optimization. *IEEE Transaction on Control Systems Technology*, 1(3):144–154, 1993.
- [28] D.E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading MA Addison Wesley, 1989.
- [29] M.P Helme. Reducing air traffic delay in a space-time network. IEEE, 1992.
- [30] C Hendrickson and G Kogur. Schedule delay and departure time decisions in a deterministic model. *Transportation Science*, 15(1):62–77, 1981.
- [31] J.H Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan press, 1975.
- [32] D.E Kaufman, J Nonis, and R.L Smith. A mixed integer linear programming formulation of the dynamic traffic assignment problem. IEEE, 1992.
- [33] H Mahmassani and R Herman. Dynamic user equilibrium departure time and route choice on idealized traffic arterials. *Transportation Science*, 18(4):362–384, 1984.
- [34] D.K Merchant and G.L Nemhauser. A model and an algorithm for the dynamic traffic assignment problems. *Transportation Science*, 12(3):183–199, 1978.
- [35] D.K Merchant and G.L Nemhauser. Optimality conditions for a dynamic traffic assignment model. *Transportation Science*, 12(3):200–207, 1978.
- [36] Z Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs*. Springer-verlag, 1992.
- [37] A Nagurney. An equilibrium scheme for the traffic assignment problem with elastic demands. *Transportation Research B*, 22B(1):73–79, 1988.
- [38] 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.
- [39] M Papageorgiou. Dynamic modeling, assignment, and route guidance in traffic networks. *Transportation Research*, 24B(6):471–495, 1990.
- [40] M. Papageorgiou. *Concise encyclopedia of traffic and transportation systems*. Pergamon Press, 1991.
- [41] B Ran, D.E Boyce, and L.J LeBlanc. A new class of instantaneous dynamic user-optimal traffic assignment models. *Operations Research*, 41(1):192–202, 1993.
- [42] O Richetta and A.R Odoni. Solving optimally the static ground holding policy problem in air traffic control. *Transportation Science*, 27(3):228–238, 1993.
- [43] O Richetta and A.R Odoni. Dynamic solution to the ground-holding problem in air traffic control. *Transportation Research*, 28A(3):167–185, 1994.
- [44] M.J Smith. The existence and calculation of traffic equilibria. *Transportation Research B*, 17B(4):291–303, 1983.

- [45] M Terrab and A.R Odoni. Strategic flow management for air traffic control. *Operations Research*, 41(1):138–152, 1993.
- [46] P.B.M Vranas, D Bertsimas, and A.R Odoni. Dynamic ground-holding policies for a network of airports. *Transportation Science*, 28(4):275–291, 1994.
- [47] P.C Vythoukas. A dynamic stochastic assignment model for the analysis of general networks. *Transportation Research*, 24B(6):453–469, 1990.
- [48] H Wang. A dynamic programming framework for the global flow control problem in air traffic management. *Transportation Science*, 25(4):308–313, 1991.
- [49] J.G Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers, Part II*, pages 325–378, 1952.
- [50] B.W Wie, R.L Tobin, and T.L Friesz. The augmented lagrangian method for solving dynamic network traffic assignment models in discrete time. *Transportation Science*, 28(3):204–220, 1994.
- [51] D.J Zawack and G.L Thompson. A dynamic space-time network flow model for city traffic congestion. *Transportation Science*, 21(3):153–162, 1987.