GECCO 2001 conference: Real-World Applications
Aircraft Ground Traffic Optimisation using a Genetic Algorithm

Brankica Pesic
Faculty of Transp and Traffic Eng
University of Belgrade
Vojvode Stepe 305, 11000 Belgrade, Yu
381 11 3091 352
dean@sf.sf.bg.ac.yu

Nicolas Durand
CENA
7, av Ed Belin
31055 Toulouse, France
(33) 562 17 40 54
durand@tls.cena.fr

Jean-Marc Alliot
CENA
7, av Ed Belin
31055 Toulouse, France
(33) 562 17 41 24
alliot@dgac.fr

Abstract

The development of air traffic during the last years, has greatly increased the density of aircraft in the airspace, and congestion on major airports. Indeed, on many airports, the taxi operation of aircraft between parking positions and runways, causes delays. The problem is increased by the development of hubs. In this article, a taxi optimisation tool using a Genetic Algorithm is introduced and tested on Roissy Charles De Gaulle Airport. The tool can help choosing the best taxiways to reduce the time spent from the gate to the runway or the runway to the gate, respecting the separation with other aircraft. It can also help choosing one way taxiways regarding to traffic and wind, and also measuring the impact of opening a new taxiway or closing an existing taxiway. Simulations are presented on a one day traffic at Paris Roissy. Delays are correlated to the traffic density on the airport.

1 Introduction

Development of hubs have generated new problems for ground operations, as all aircraft are tending to move at the same time on the airport. Thus, new delays are introduced on major airports due to ground congestion. Airport designers are tempted to build new taxiways to reduce congestion and improve the efficiency of ground operations, but by the moment, no tool can help them to measure the efficiency of their choices.

As many research projects are concentrated on decision making tools for airspace controllers, little work has been done on ground control. The SIMMOD[1] project developed by the FAA[2] is a heavy software that was not designed to give any advice to ground controllers. The SMA[3] project was developed by the FAA and NASA[4] to help current airport facilities to operate more efficiently. Many efforts were concentrated on improving the information sharing of the different operators on the ground. The DP[5] project ([IDA+98]) only focuses on improving the performance of departure operations, without taking into account the taxi problem. The TAAM[6] project ([Gro99]) is developed by The Preston Group. Trajectory optimisation partly exists and it uses notions of reachable gates. The conflict detection and resolution is not developed. Finally, a component of the TARMAC[7] project, developed by the DLR[8] Institute of Flight Guidance, focuses on the ATC[9]-related traffic planning systems for airport movements, but does not introduce any optimisation tool to taxi aircraft.

In this article, a taxi optimisation tool is introduced and tested on Roissy Charles De Gaulle Airport. The tool chooses the best trajectory to reduce the time spent from the gate to the runway or the runway to the gate, respecting the separation with other aircraft. It can also help choosing one way taxiways regarding to traffic and wind, and also measuring the impact of opening a new taxiway or closing an existing taxiway. The problem is introduced and modelled in the first part. The different algorithms used to solve the

[1] Simulation Model (http://www.atac.com/simmod/)

[2] Federal Aviation Administration

[3] Surface Movement Adviser (http://surface.arc.nasa.gov/sma/)

[4] National Aeronautics and Space Administration

[5] Departure Planer

[6] Total Airspace and Airport Modeller

[7] Taxi and Ramp Management And Control (http://dv.bs.dlr.de/ff/fl/24/tarmacs-as)

[8] Deutsches Zentrum fur Luft und Raumfahrt

[9] Air Traffic Control

problem are detailed in part 2. The last part gives the results of a full simulation on a one day traffic on Roissy Airport.

2 Problem modelling

The problem is to find for each aircraft an optimal path from its parking to a given runway holding position or from its runway exit to its parking, and respect a given separation between aircraft.

An optimal path can have different definitions as for example the length of the path or the total taxiing time. Holding on a taxiway can be more or less penalising than increasing the length of the path. It can be cheaper to hold at the parking position than on a taxiway.

It can be better, for example, to lengthen slightly the routes of two aircraft than to make one aircraft wait a long time. Therefore a global optimum criteria will have to be defined in the following. However, the purpose of this article is not to discuss the choice of such criteria, which depend on many different factors related to the airport geometry, the traffic, and airlines preferences...

By the way, it is quite difficult to predict with a good accuracy the future positions of aircraft on taxiways. First of all, the exact departure time is generally known only a few minutes in advance (many factors can cause delays), and the exact landing time depends on the runway sequencing. A modelling that can afford these uncertainties is necessary in order to build a realistic tool.

2.1 Airport structure

The airport is defined by a graph: links represent taxiway segments whereas nodes are taxiway intersections or connections, parking, holding positions, and runway exits. Figure 2 represents the graph of Roissy airport. The graph is obviously connected. A Dijkstra algorithm [AMO93] can be used to compute the minimum length from any node of the graph to every parking, holding point, or runway exit. An A^* algorithm [Pea84] can as well be used to compute the minimum length from any node of the graph to every parking, holding point or runway exit, taking into account the limitations in terms of turning rate. Therefore, extra time is added depending on the turning rate (see figure 1).

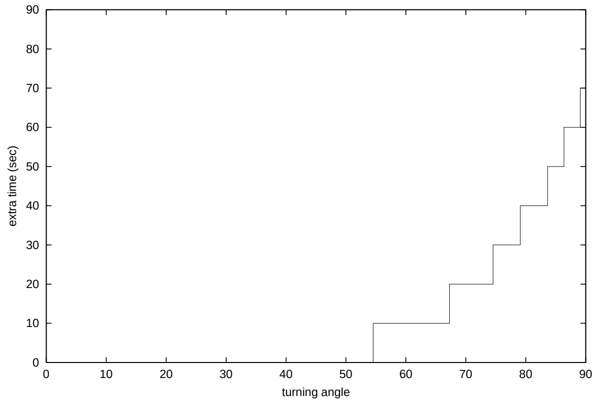

Figure 1: Additional delay as a function of the turning angle

2.2 Aircraft possible manoeuvres

In order to minimise the delays and ensure the separations, the path of aircraft can be modified, or aircraft can hold position at their parking, on taxiways or at the holding point before taking off. Two aircraft are in conflict if the distance between them is less than 60 meters at every time, except on the parking position on which this distance can be reduced. In order to have simple manoeuvres, only one holding order should be given to the pilot at a time (starting at t_0 and ending at t_1). The path should also respect some constraints: turning angles are limited by the aircraft performance, an aircraft should not use the same taxiway twice in the same direction, there cannot not be more than one aircraft on a runway at the same time ... In order to simplify the problem, aircraft are supposed to have constant speed except when they turn.

Alternate paths lengthening the trajectory less than a certain distance can be computed with a simple Branch and Bound algorithm [HT95].

Figure 2 gives an example of the shortest path calculated between a runway exit and a gate. The 467 alternate paths lengthening the trajectory less than 500 meters are also represented.

2.3 AGTO modelling

As the aircraft future positions and movements are not known with a good accuracy, it is necessary to regularly update the situation, every Δ minutes for example. By the same time, looking a long period ahead is not possible as predictions are not good enough. Consequently a time window $T_w > \Delta$ is defined.

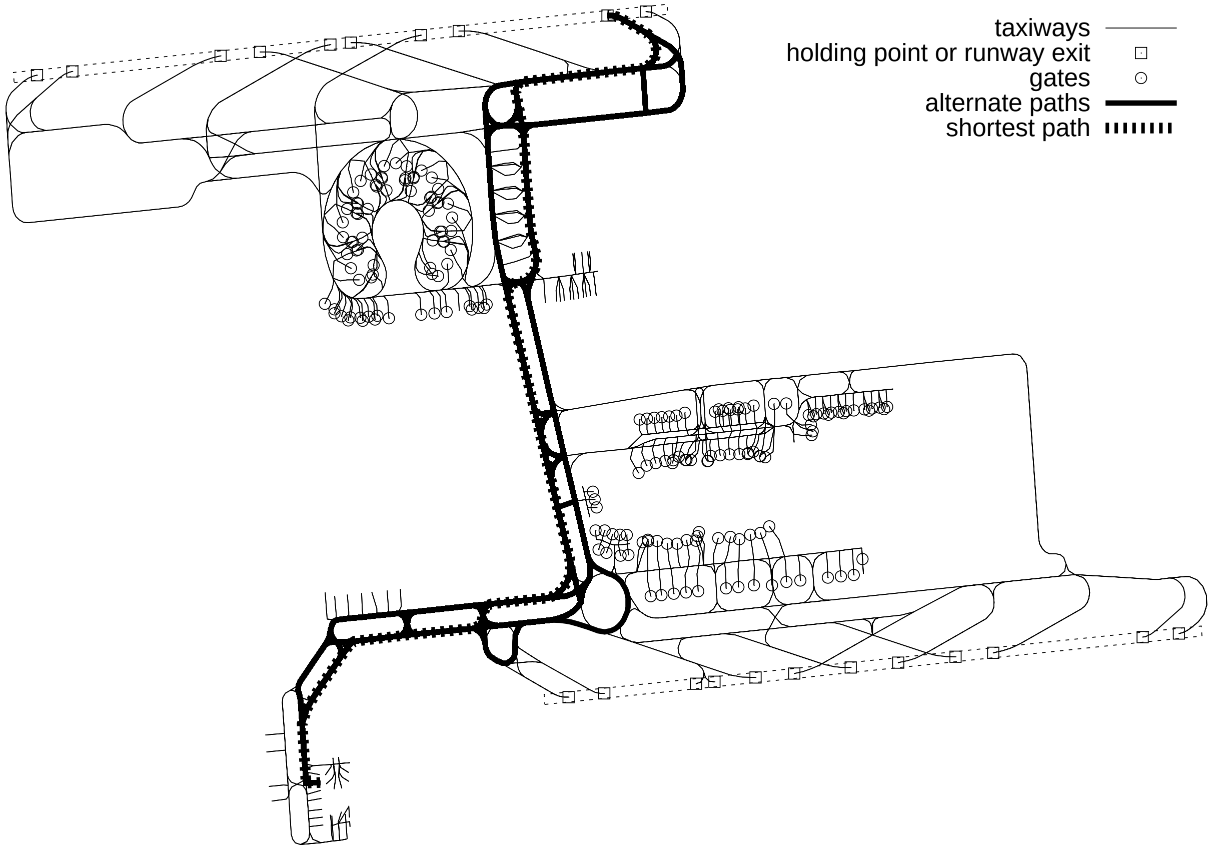

Figure 2: Roissy airport graph - Example of shortest and alternate paths

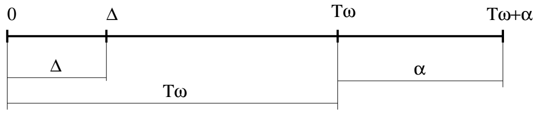

Figure 3: Time window

Only aircraft arriving or taking off in the time window will be considered. The time window will be shifted every Δ minutes, the problem reconsidered and a new optimisation performed. As an aircraft may taxi more than T_w minutes, the horizon effect problem can appear. Indeed, as we only look at aircraft during the T_w period, a conflict may appear a few seconds after the end of the T_w period without being detected. In order to prevent the optimisation process from building solution that would not be acceptable at the next shift, the time window is increased of α seconds during the optimisation process (see figure 3).

3 GAs applied to AGTO

In this paper, classical Genetic Algorithms and Evolutionary Computation principles such as described in the literature [Gol89, Mic92] are used. The algorithm is used every Δ minutes on the problem defined in section 2.3.

3.1 Data structure

During each optimisation process, each aircraft trajectory is described by 3 numbers (n, t_0, t_1). n is the number of the path : as detailed in section 2.2, all the alternate paths lengthening the aircraft trajectory less than some distance can be initially computed and sorted. The aircraft may hold position at t_0 and resume taxi at t_1 (if $t_0 = t_1$, the aircraft does not stop). When N aircraft are simultaneously taxiing, the problem is defined by $3N$ variables.

3.2 Fitness function

The fitness function must ensure that a solution without any conflict is always better than a solution with a conflict. Consequently it was decided that the fitness of a solution without conflict should be less than $\frac{1}{2}$ and the fitness of a solution with a conflict more than $\frac{1}{2}$. The different conflicts between each pair of aircraft can be initially computed in a $(n \times n)$ matrix (see table 1). A conflict during 3 time steps between aircraft i and j sets elements (i,j) and (j,i) to 3. Element (i,i) is filled with the trajectory lengthening due to the path chosen and holding time $(t_1 - t_0)$.

	(1)	...	...	(j)	...	(i)	...	...	(n)
(1)	80	0	0	0	0	0	0	0	0
(2)	0	0	0	0	0	0	0	0	0
...	0	0	30	0	0	0	0	0	0
(j)	0	0	0	0	0	3	0	0	0
...	0	0	0	0	25	0	0	0	0
(i)	0	0	0	3	0	0	0	0	0
...	0	0	0	0	0	0	0	0	0
...	0	0	0	0	0	0	0	0	0
(n)	0	0	0	0	0	0	0	0	0

Table 1: Fitness matrix

Using the fitness matrix M_f, it is possible to compute the fitness value as follows :

If the matrix is diagonal :

$$F = \frac{1}{2} + \frac{1}{1 + \sum_{i=0}^{n} M_f(i,i)}$$

Else :

$$F = \frac{1}{2 + \sum_{i<j} M_f(i,j)}$$

3.3 Crossover operator

The conflict resolution problem is partially separable as defined in [DA98, DAN96]. In order to increase the probability of producing children with a better fitness than their parents, principles applied in [DA98] were applied. For each aircraft i of a population element, a local fitness F_i value is defined as the sum of the i^{th} line (or column) of the fitness matrix (except the diagonal element).

$$F_i = \sum_{j \neq i} M_f i, j$$

The crossover operator is presented on the figure 4. First two population elements are randomly chosen. For each parent A and B, fitness A_i and B_i of aircraft i are compared. If $A_i < B_i$), the children will take aircraft i of parent A. If $B_i < A_i$, the children will take aircraft i of parent B. If $A_i = B_i$ children randomly choose aircraft A_i or B_i or even a combination of A_i and B_i.

3.4 Mutation operator

For each candidate to mutation, parameters of an aircraft having one of the worst local fitness are modified (n, t_0, t_1). If every conflict is solved, an aircraft is randomly chosen and its parameters changed.

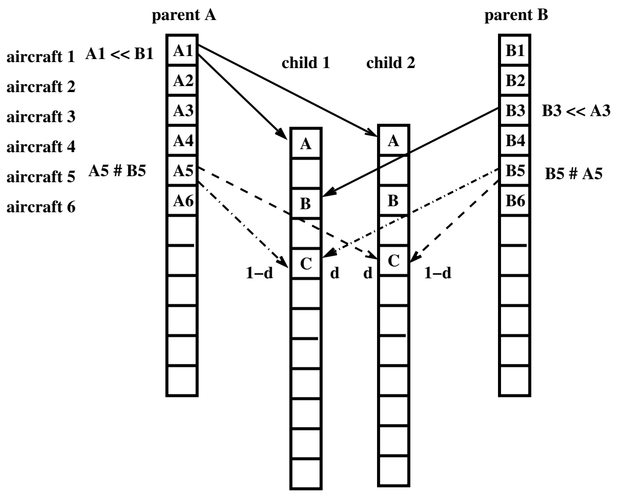

Figure 4: Crossover operator

The crossover and mutation operators are quite deterministic at the beginning because there are many conflicts to solve. They focus on making feasible solutions. When the solutions without conflict appear in the population, they become less deterministic.

3.4.1 Sharing

The problem is very combinatorial and may have many local optima. In order to prevent the algorithm from a premature convergence, the sharing process introduced by Yin and Germay [YG93] is used. The complexity of this sharing process has the great advantage to be in $n \log(n)$ (instead of n^2 for classical sharing) if n is the size of the population.

A distance between two chromosomes must be defined to implement a sharing process. Defining a distance between two sets of N trajectories is not very simple. In the experiments, the following distance is used introduced:

$$D(A,B) = \frac{\sum_{i=0}^{N} |l_{A_i} l_{B_i}|}{N}$$

l_{A_i} (resp l_{B_i}) is the i^{th} aircraft path length of chromosome A (resp B). As the paths are sorted according to their length, the distance increases with the difference of lengths.

3.5 Ending criteria

As time to solve a problem is limited, the number of generations is limited, as follows: as long as no available solution is found, the number of generation is limited to 100. The algorithm is stopped 20 generations after the first acceptable solution (with no remaining conflict) is found.

4 Experimental results

The experimental results presented in this section have been computed with real flight plans on a complete day at Roissy Airport (May 22^{nd} 1999). During that day, some aircraft land, other aircraft take off and some aircraft land and take off. Aircraft are assigned to terminals according to the airline they belong to (for example an Air France flight is assigned to Roissy 2).

When taking off or landing, aircraft are randomly assigned one of the two runways. They are sequenced on runways every minute using the first in first out principle.

Three hypotheses are done:

- in the **"random hypothesis"**, taking off and landing aircraft are randomly allocated both runways.
- in the **"deterministic hypothesis"**, taking off and landing aircraft are allocated the runway that minimises the distance to the allocated parking.
- in an **"middle hypothesis"**, taking off aircraft are randomly allocated both runways and landing aircraft are allocated the runway that minimises the distance to the parking.

The three hypotheses are tested with the genetic algorithm. The last hypotheses is tested with a 1-to-n strategy that uses an A^* algorithm: aircraft are sorted according to their time of departure or arrival, each aircraft trajectory is then optimised considering previous aircraft trajectory as a constraint.

4.1 Parameters

- $T_w = 12mn$
- $\alpha = \Delta = 3mn$
- Population size: 300
- Max number of generations: 100
- Crossover rate: 60%
- Mutation rate: 15%
- Selection principle: stochastic reminder without replacement

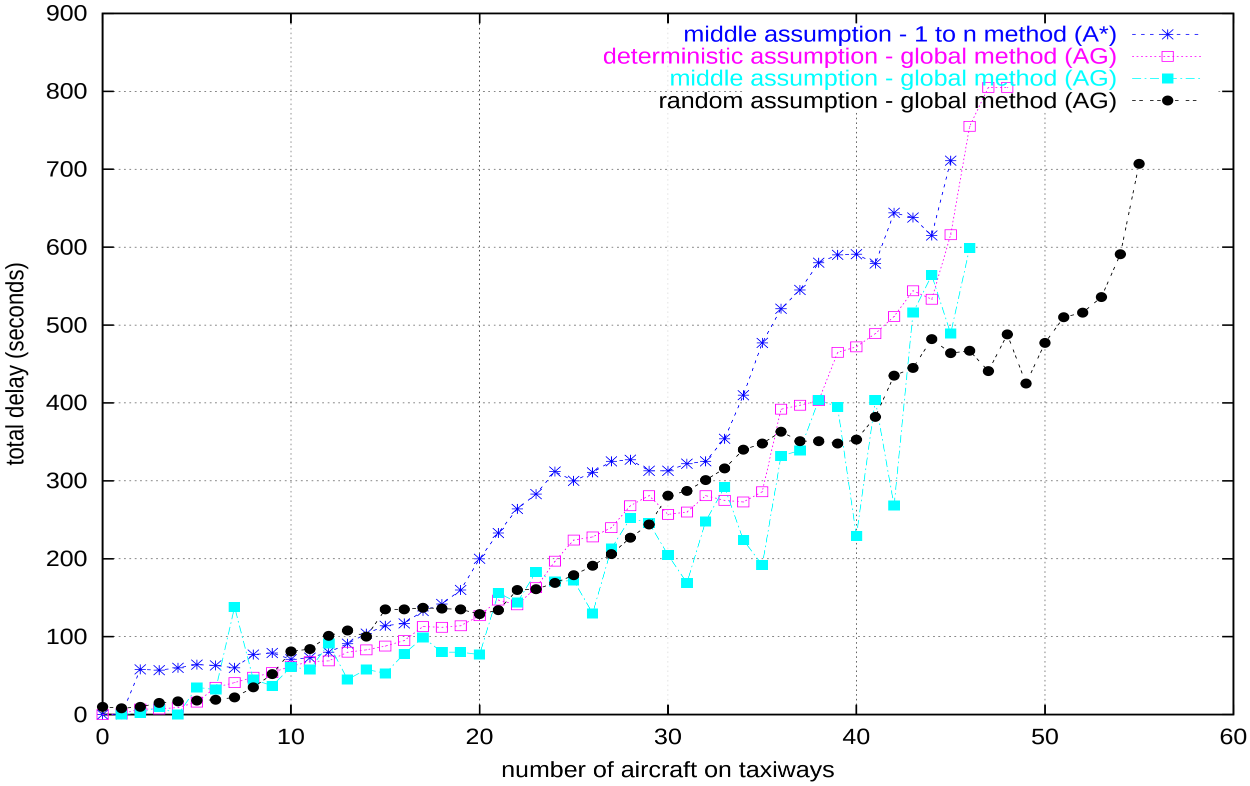

Figure 5: Total delay as a function of the number of aircraft on taxiways.

Hypothesis	mean delay	maximum number of aircraft
random (GA)	255	55
deterministic (GA)	198	48
middle (GA)	195	46
middle (A^*)	271	45

Table 2: Mean delay and maximum number of aircraft for the different hypotheses

4.2 Comparing 1-to-n to the global strategy

Figure 5 gives the mean delays as a function of the number of aircraft moving on the taxiways for the different hypotheses. The 1 to n method using an A^* algorithm produces more delays than the global method using the Genetic Algorithm, whatever the chosen hypothesis.

Table 2 gives for the different hypotheses the mean total delay and the maximum number of aircraft simultaneously moving.

The middle hypothesis (GA) penalises less aircraft than the other hypotheses and a smaller number of aircraft are moving at a time. The random hypothesis (GA), which is probably more in accordance with reality (the parking position does generally not influence the runway allocation), is more penalising (each aircraft is delayed 1 minute more). The 1-to-n strategy is more penalising for a number of aircraft that is not bigger, which can be explained by the weakness of the strategy.

4.3 Genetic algorithm efficiency

In order to observe the GA efficiency, figure 6 gives the number of generations required by the GA as a function of time. the different peaks appearing at 7, 8, 10, 11 am and 5 pm are the traffic peaks. Figure 7 shows the correlation between the number of generation required by the GA and the number of moving aircraft on the ground.

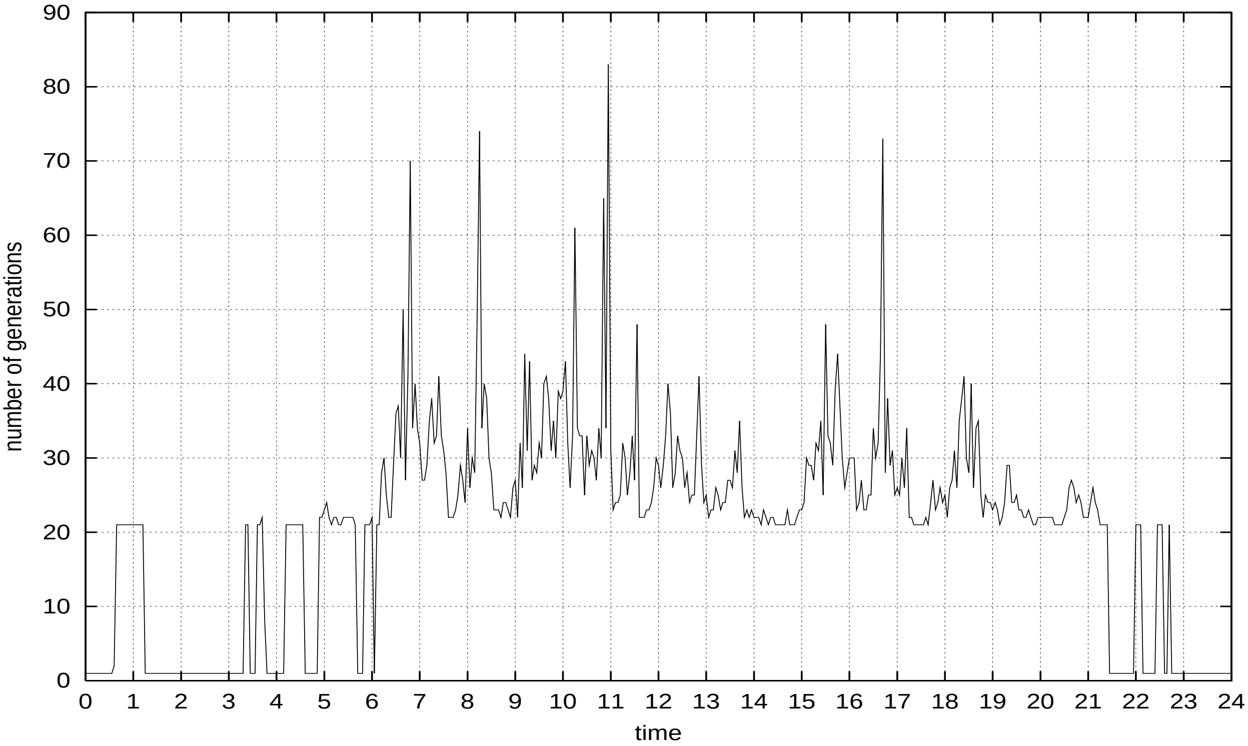

Figure 6: Number of generations (random strategy) as a function of time

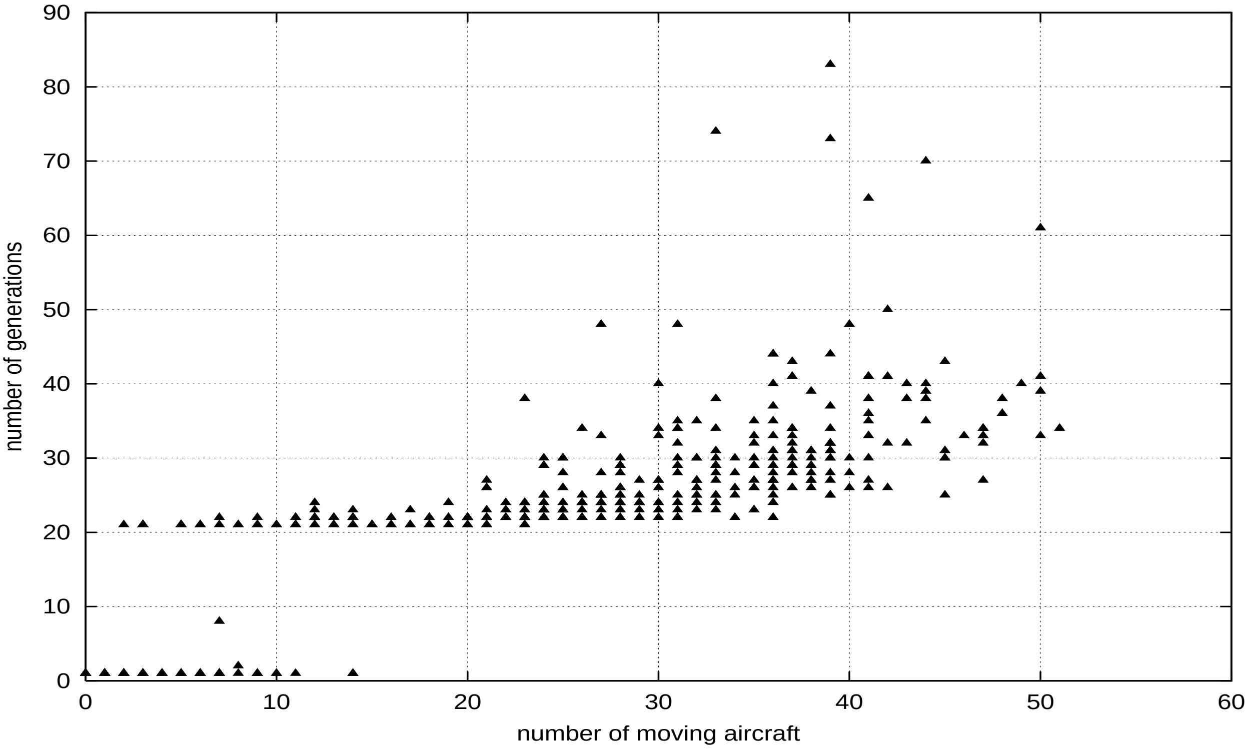

Figure 7: Number of generations (random strategy) as a function of the number of moving aircraft

5 Conclusion and further work

This preliminary work has shown that it was possible to build a taxiway adviser in order optimise the aircraft ground traffic on big airports such as Roissy Charles de Gaulle. If many hypotheses have been simplified in order to focus on the algorithm, it can be noticed that the modelling can be improved in order to take into account different speeds, uncertainties on speeds etc... without changing the algorithm itself. Further work will focus on these improvements. Genetic Algorithms are very efficient on the problem as they search the global optimum of the problem whereas a deterministic algorithm such as an A^* algorithm can only reasonably be used with a 1-to-n strategy, which is very poor.

References

[AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows, Theory, Algorithms and Applications.* Prentice Hall, 1993.

[DA98] N. Durand and J. M. Alliot. Genetic crossover operator for partially separable functions. In *Proceedings of the third annual Genetic Programming Conference*, 1998.

[DAN96] Nicolas Durand, Jean-Marc Alliot, and Joseph Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the Symposium on Applied Computing, Philadelphia.* ACM, 1996.

[Gol89] D.E Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading MA Addison Wesley, 1989.

[Gro99] The Preston Group. *TAAM Reference Manual.* The Preston Group, 1999.

[HT95] Reiner Horst and Hoang Tuy. *Global Optimization, Deterministic Approaches.* Springler, 1995.

[IDA$^+$98] A.H Idris, B Delcaire, I Anagnostakis, W.D Hall, J.P Clarke, R.J Hansman, E Feron, and A.R Odoni. Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools. In *2nd USA/Europe Air Traffic Management R & D Seminar*, Orlando, December 1998.

[Mic92] Z Michalewicz. *Genetic algorithms + Data Structures = Evolution Programs.* Springer-verlag, 1992.

[Pea84] Judea Pearl. *Heuristics.* Addison-Wesley, 1984. ISBN: 0-201-05594-5.

[YG93] Xiaodong Yin and Noel Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In C.R. Reeves R.F.Albrecht and N.C. Steele, editors, *In proceedings of the Artificial Neural Nets and Genetic Algorithm International Conference, Insbruck Austria.* Springer-Verlag, 1993.