Aircraft Ground Traffic Optimization

Jean-Baptiste Gotteland gotteland@recherche.enac.fr
Nicolas Durand durand@tls.cena.fr
Jean-Marc Alliot alliot@dgac.fr
Erwan Page page@cena.fr

Abstract

Air traffic growth and especially hubs development cause new significant congestion and ground delays on major airports.

Accurate models of airport traffic prediction can provide new tools to assist ground controllers in choosing the best taxiways and the most adapted holding points for aircraft. Such tools could also be used by airport designers to evaluate possible improvements on airport configurations and airport structure.

In this paper, a ground traffic simulation tool is proposed and applied to Roissy Charles De Gaulle and Orly airports. A global optimization method using genetic algorithms is compared to a 1-to-n strategy to minimize time spent between gate and runway, while respecting aircraft separation and runway capacity.

In order to compare the efficiency of the different optimization methods, simulations are carried out on a one day traffic sample, and ground delay due to holding points or taxiway lengthening is correlated to the traffic density on the airport.

1 Introduction

Traffic delay due to airport congestion and ground operations becomes more and more penalizing in the total gate-to-gate flight cycle. This phenomenon can be in a large part attributed to recent hubs development, as all departures and arrivals are tending to be scheduled at the same time. Moreover, many ATC problems and environmental inefficiencies can appear as a result of taxi queueing and take-off time uncertainty. As airport designers are in charge to build new taxiways to reduce congestion and improve ground operations, ground simulation tools become essential to validate their choices before realization.

Even if most research projects are concentrated on decision making tools for airspace controllers and do not consider ground operations utilities, highly detailed models of airport operations already exist, such as SIMMOD[1] or TAAM[2]. They can be useful to evaluate qualitatively the relative effects of various airport improvements. The DP[3] project ([IDA$^+$98]) focuses on improving the performance of departure operations. Finally, a component of the TARMAC[4] project focuses on the ATC-related traffic planning systems for airport movements.

In this paper, a ground traffic simulation tool with a conflict resolution module is introduced and tested on a one day traffic sample on Roissy Charles De Gaulle and Orly airports. Different optimization strategies are used to find the best trajectory and the most adapted holding points for taxiing aircraft. The goal is to minimize the time spent from gate to take-off or from landing to gate, respecting the separation with other aircraft and the runway capacity. During the optimization process, actual one way taxiways, operational airport configurations and speed uncertainty are considered.

2 Problem modeling

The problem is to find, for each aircraft, an optimal path from its gate to a given runway take-off position or from its runway exit to its gate position, respecting a given separation between aircraft.

An optimal path can have different definitions: for example, the length of the path or the total taxiing time. At the same topic, holding on a taxiway can be more or less penalizing than increasing the length of the path or holding at the gate position.

[1] SIMulation MODel (FAA)

[2] Total Airspace and Airport Modeler (Preston Group)

[3] Departure Planer

[4] Taxi and Ramp Management And Control (DLR)

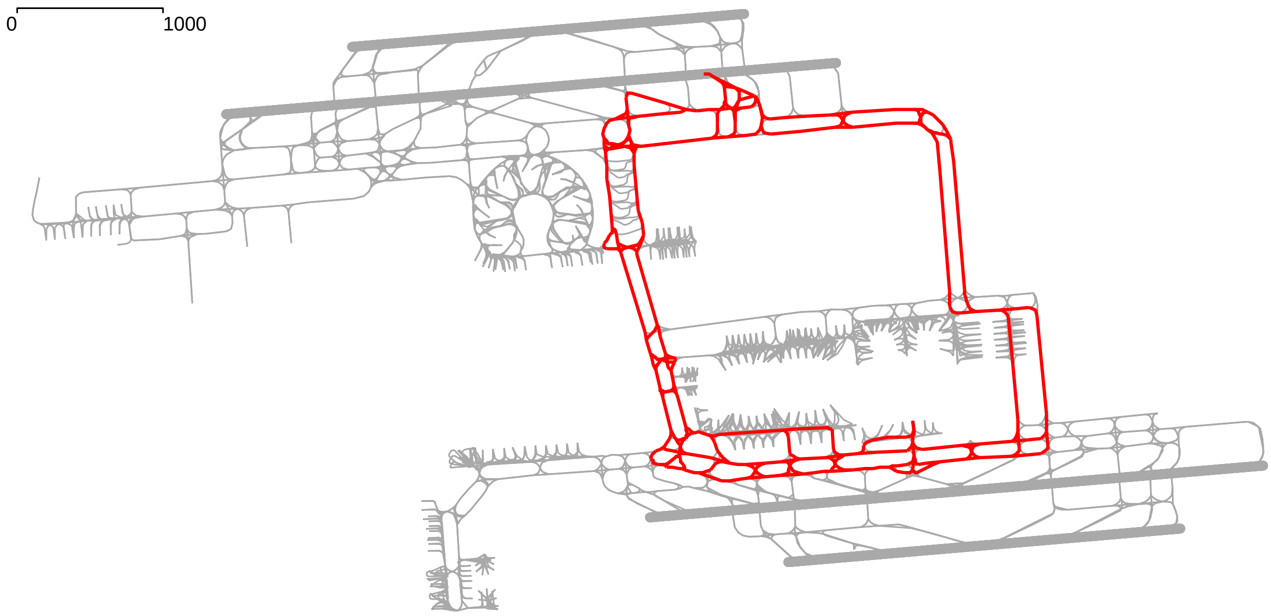

Figure 1: Roissy airport graph - Example of shortest and alternate paths

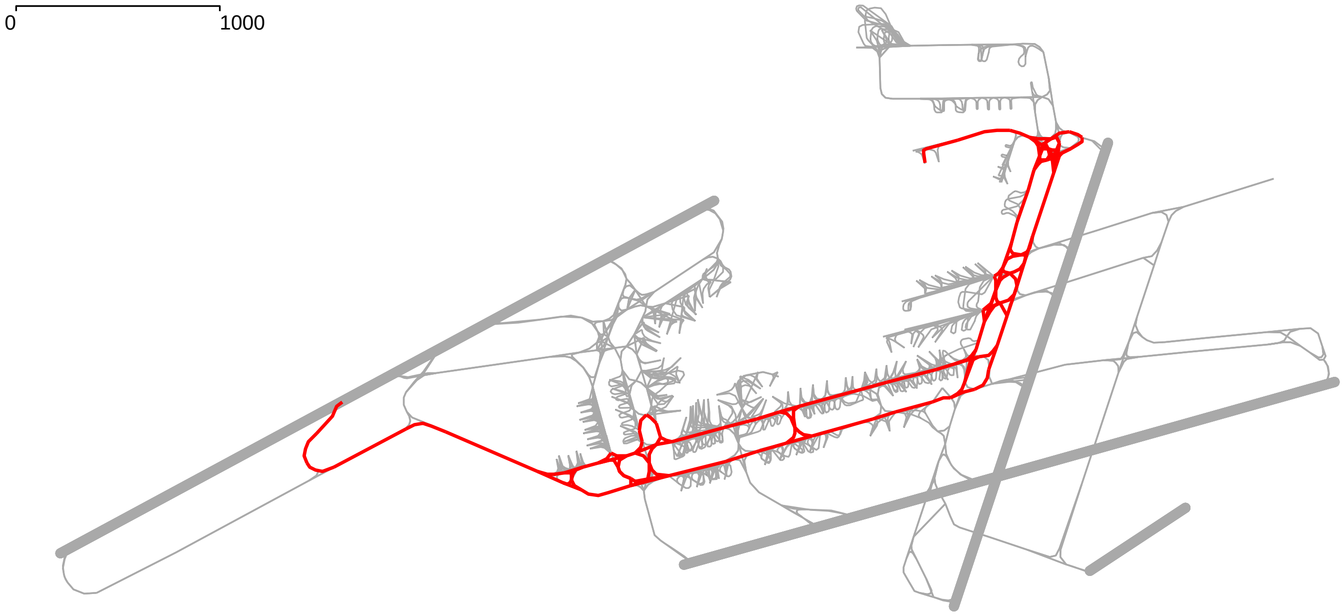

Figure 2: Orly airport graph - Example of shortest and alternate paths

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 can be refined without modifying the algorithm itself, considering many different factors related to the airport geometry, the traffic, or 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. Hence, the proposed model should take into account speed uncertainty and must be regularly updated with real aircraft positions.

2.1 Airport structure

An airport is described by its gates, taxiways and runways. Different kind of taxiways can be differenced:

- Gate specific access (entries, forward exits or push-backs), characterized by a very low speed;
- Runways access (entries and exits), containing the actual holding points before take-off and exit points after landing with specific speed limitations;
- Taxiways intersecting runways, with access restrictions ;
- Simple taxiways, where speed limitations is modeled as a function of the turning rate (figure 3).

Connections between taxiways are limited (it is not always possible to proceed from a taxiway to another, even if they are intersecting). The airport description specify usable taxiways connections.

Thus, the airport is defined by a graph: links represent connections between taxiways whereas nodes are taxiways themselves, gate positions, and landing or take-off points. The cost from a taxiway node to its connected nodes is the time spent to proceed via this taxiway, taking into account speed limitations due to this taxiway. The cost from the other nodes (gates and runway positions) to their connected nodes is null.

Figure 1 represents the graphs of Roissy and Orly airports. These graphs are obviously connected. Classic graph algorithms can be used to compute alternative paths for aircraft:

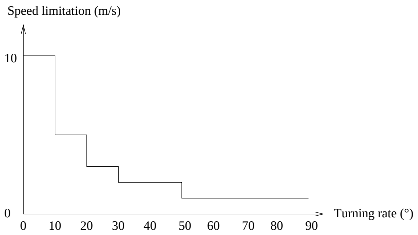

Figure 3: Speed limitation as a function of turning rate

- An A^* algorithm [Pea84] can compute the best path and the corresponding minimal time spent between two given nodes (gate and runway entry for example).
- A Dijkstra algorithm [AMO93] can compute best paths and corresponding minimal time spent from a given node to every other node.
- A Recursive Enumeration algorithm [MJ96] using the Dijkstra's result can compute the k best paths from a given node to another.
- A Branch and Bound algorithm [HT95] can compute all alternate paths lengthening the best path less than a given distance or time.

2.2 Aircraft model

Aircraft are described by their flight-plan (ident, departure or arrival time, gate position, requested runway, eventually their CFMU slot...), their wake turbulence category (low, medium or high) and their take off or landing distance.

In order to perform conflict detection, a model for aircraft separation is defined. This model takes into account runways area, 90 meters away from each side of the runway (or 150 meters away on bad weather conditions). On these area, aircraft are considered on the runway even if they are not taking off or landing.

Aircraft separation model is defined as follows :

- aircraft on gate position are separated with all other aircraft.
- The distance between two taxiing aircraft must never be lower than 60 meters.
- No more than one aircraft at a time can take off or land on a given runway.

- A time separation of 1, 2 or 3 minutes (depending on the aircraft category) is necessary after a take off to clear next take off or landing from wake turbulence.

- When an aircraft is proceeding for take off or landing on a given runway, other aircraft can be taxiing on the same runway area only if they are behind the proceeding one.

2.3 Speed uncertainty

Speed uncertainty is modeled as a fixed percentage of the initial defined speed (which is function of procedures and turning rate). Therefore, an aircraft is considered to occupy multiple possible positions at a given time.

Separation is ensured if all of the possible aircraft positions are separated from others, as defined before.

When an aircraft is following an other one, its speed uncertainty will be reduced as the pilot won't go faster than the first one.

Speed uncertainty reduces the validity period of predictions. Thus, simulations with speed uncertainty will be carried out with a lower time window (see 2.5).

2.4 Aircraft maneuvers

In order to minimize the total delay and to ensure separations, the path of an aircraft can be modified and aircraft can hold position at the gate, on taxiway or queue at the holding point before take off.

Thus, a ground control order is described by :

- The path that the aircraft must follow, choosen among the computed possible paths for the aircraft;

- Eventually, the holding position p on this path and the time t until which the aircraft must hold on.

In order to perform acceptable maneuvers, only one holding order should be given to the pilot at a time, and proposed alternative paths should not lead an aircraft to use the same taxiway twice.

With such a holding model (hold at position p until time t) uncertainties defined before can be reduced, while referencing a precise holding position and a precise end holding time (see figure 4).

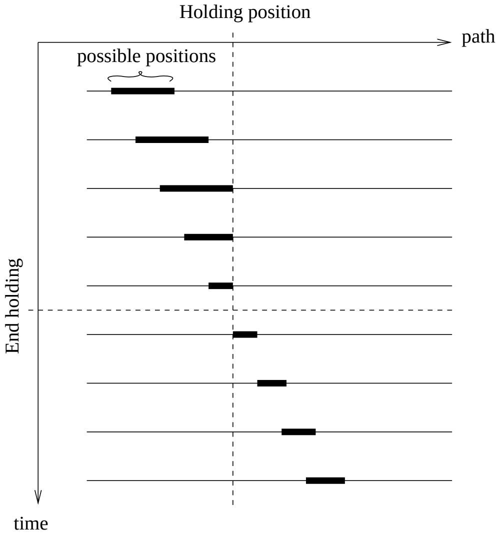

Figure 4: Uncertainty reduction on holding points

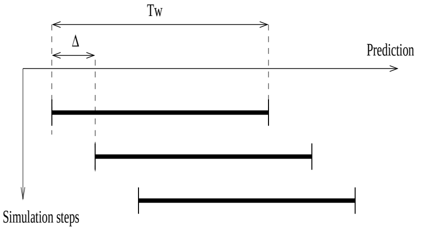

Figure 5: Time window

2.5 Simulation model

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. Only aircraft taxiing in the time window will be considered. The time window will be shifted every Δ minutes, the problem reconsidered and a new optimization performed (see figure 5).

At each simulation step (every Δ minutes), traffic prediction is performed for the next T_w minutes and pairs of conflicting aircraft positions are extracted. Conflict resolution for this simulation step consist in choosing for each aircraft a path among the given set of possible paths and an optional holding point and time to ensure separations.

2.6 Global optimum criteria

In the current version, the global criteria to minimize is defined by the total rolling time (including queueing for runway time), added to the time spent in lengthened trajectory. With this definition, lengthening trajectory appears to be twice more penalizing than holding position.

3 A*: 1-to-n strategy

In this strategy, aircraft are sorted and considered one after the other.

The optimization problem is reduced to one aircraft: the algorithm must find the best path and/or the best holding point for the aircraft, taking into account the trajectories of the other aircraft already considered. In this point of view, first considered aircraft have priority on last considered aircraft.

3.1 Graph modeling

The 1-to-n strategy for an aircraft can be modeled as a graph exploration problem :

- A node of the graph is a position in a path p_i of the aircraft at time t.
- An heuristic function for this node is the minimal remaining time to reach the end of the path.
- If a node represents a conflicting position with already considered aircraft, it has no son.
- Each non conflicting node has two sons :
  - The first son is the next position in the same path p_i at time $t+1$ (the aircraft go forward). The cost to reach this son is 1.
  - The second son is the same position at time $t+1$ (the aircraft holds position at time t). The cost to reach this son is 2, as a delay is given to the aircraft.
- The root nodes are the first position on each path p of the aircraft at current time t_0.
- The terminal nodes are the ones describing a non conflicting position of the aircraft at time $t_0 + T_w$.

An A* algorithm can easily find the best solution for the aircraft.

3.2 Sorting method

As last considered aircraft are extremely penalized (they must avoid all first considered aircraft) the way to sort aircraft is a determining factor.

A simple way to assign priority levels is to consider the flight-plan transmission time to the ground controllers.

This option seams the most realistic as ground controllers can hardly take into account an aircraft without its flight-plan. In the simulation context, this is equivalent with sorting aircraft by their departure or arrival time.

However, this option must be refined :

- As landing aircraft can't hold position before exiting runway, their priority level must be higher than all taking off aircraft.
- Queueing for runway aircraft should be sorted in their queue order.

In order to satisfy these principles, a time T_a is affected to each aircraft as a function of its begining time T_0 and its remaining time t_r:

$T_a = T_0 + t_r$ for departures,

$T_a = T_0 - 1hour$ for arrivals.

Aircraft are sorted by increasing values of T_a.

4 Genetic Algorithms

In these strategies, 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 2.5.

Two strategies are developed : in the first one, the algorithm finds a path and an optional holding position for each aircraft. In the secund one, the genetic algorithm finds a path and a priority level for each aircraft, and an A* algorithm is used to compute the resulting trajectories.

4.1 Data structure

During each optimization process, each aircraft trajectory is described by its own parameters:

- The first strategy needs 3 numbers (n, p, t) for each aircraft: n is the number of the path, p and t the evantual holding position for the aircraft (if

p is reached after t, the aircraft does not stop) as detailed in section 2.4.

- The secund strategy needs 2 numbers ($n, prio$) for each aircraft: n is the number of the path and $prio$ its priority level.

4.2 Fitness function

For the two strategies, 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 with a conflict should be less than $\frac{1}{2}$ and the fitness of a solution without any conflict more than $\frac{1}{2}$.

Thus, for a solution with n_c remaining conflicts,

$$F = \frac{1}{1 + n_c}$$

For a solution without any conflict,

$$F = \frac{1}{2} + \frac{1}{2 + \sum_{i=1}^{N} d_i + l_i}$$

where d_i is the delay of aircraft i and l_i the time spent by aircraft i in lengthened trajectory.

4.3 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 is defined as :

- for an aircraft with $n_c > 0$ conflicts,

  $F_i = 1000 * n_c$;

- for a non conflicting aircraft $F_i = d_i + l_i$.

The crossover operator is presented on the figure 6. 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 $A_i \ll B_i$, the children will take aircraft i of parent A. If $B_i \ll A_i$, the children will take aircraft i of parent B. If $A_i \equiv B_i$ children randomly choose aircraft A_i or B_i or even a combination of A_i and B_i.

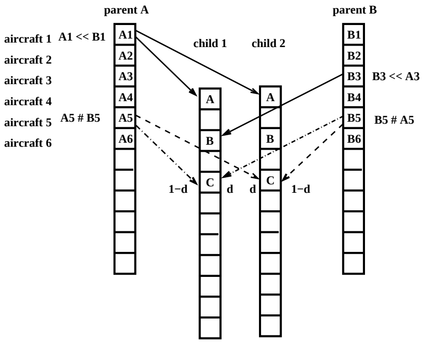

Figure 6: Crossover operator

4.4 Mutation operator

For each candidate to mutation, parameters of an aircraft having one of the worst local fitness are modified.

The crossover and mutation operators are quite deterministic at the beginning as many conflicts have to be to solved. They focus on making feasible solutions. When solutions without conflict come in the population, they become less deterministic.

4.5 Clusters

In order to lower the complexity of the problem as often as possible, a transitive closure is applied on conflicting aircraft pairs and gives the different clusters of conflicting aircraft [DAN96]. The different clusters will be solved independently at first. If the resolution of two clusters creates new conflicting positions between them, the two clusters are unified and the resultant cluster is solved.

4.6 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 introduced:

$$D(A,B) \quad = \quad \frac{\sum_{i=1}^{N} |l_{A_i} l_{B_i}|}{N}$$

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.

4.7 Ending criteria

As time to solve a problem is limited, the number of generations is limited: as long as no available solution is found, the number of generation is limited to 50. The algorithm is stopped 20 generations after the first acceptable solution (with no remaining conflict) is found.

5 Experimental results

5.1 Simulations

Simulations are carried out with real flight plans of Roissy Charles De Gaulle and Orly airports on a complete day (May 18^{nth} 1999).

Three strategies are compared :

- in the **“1-to-n method”**, aircraft are sorted as described in 3.2. They keep the same priority level during all the simulation and an A* algorithm finds the best solution.
- in the **“Global method”**, a genetic algorithm finds a path and an optional holding position for each aircraft in order to minimize the global criteria described in 2.6.
- in the **“Mixed global method”**, a genetic algorithm finds a path and a priority level for each aircraft and the fitness function is computed by an A* algorithm applied on sorted aircraft.

Simulations parameters:
Time window : $T_w = 5mn$
Simulation step : $\Delta = 2mn$
Speed uncertainty : $\delta = 10\%$
GA Population size: 200
GA number of generations: 50
GA Crossover rate: 60%
GA Mutation rate: 15%
GA Selection principle: stochastic reminder without replacement

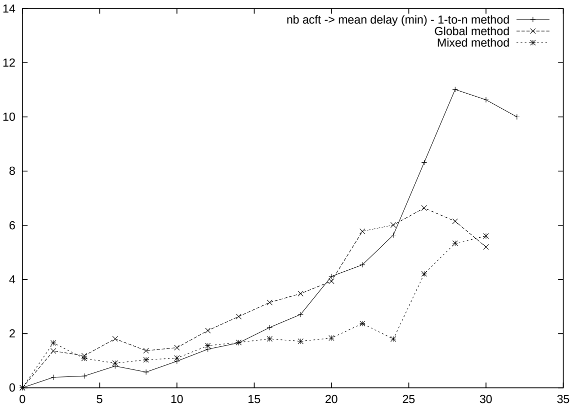

Figure 7: Mean delay as a function of the number of moving aircraft.

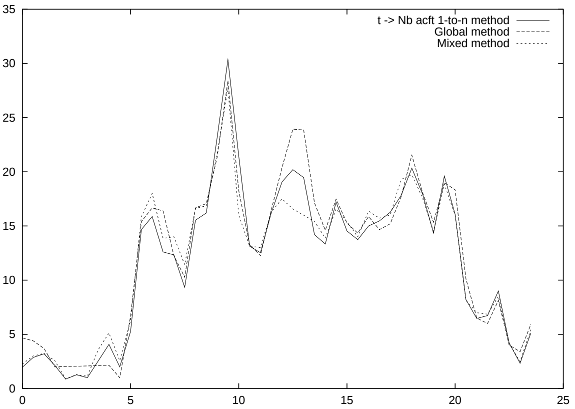

Figure 8: Number of aircraft as a function of time.

5.2 Comparing the strategies

As Roissy and Orly simulation results has given the same relative conclusions about the 3 strategies efficiency, figures related in this article only concerns Roissy Charles De Gaulle airport.

Figure 7 gives the mean delay as a function of the number of aircraft moving on the taxiways for the different methods. When number of aircraft increases, the mixed method appears to be the best one.

Figure 8 gives for the different strategies the number of aircraft simultaneously moving as a function of time. It appears that the mixed method keeps a lower number of moving aircraft during heavy time periods: a good resolution of ground traffic conflicts allows to decrease delay and then leads better situations with less moving aircraft.

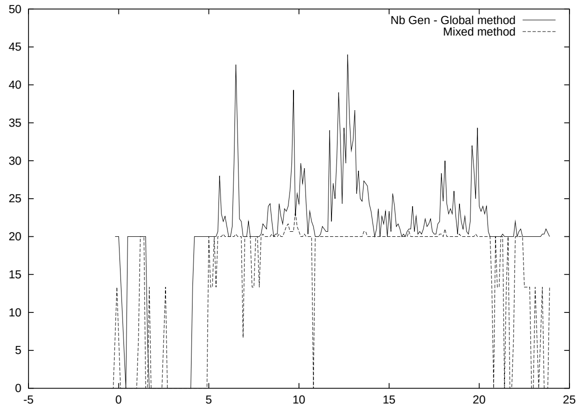

Figure 9: Number of generations as a function of time

5.3 Genetic algorithm efficiency

In order to observe the GA efficiency, figure 9 gives the number of generations required by the GA as a function of time for the two GA strategies.

For the global method, the different peaks which appears at 7, 10 am, 1 and 7 pm are the traffic peaks.

For the mixed method, the global optimum is always found with a few number of generations : sorting aircraft by evolutive priority levels seams to be very efficient as far as ground conflicts resolution is concerned.

6 Conclusion and further work

A preliminary work has shown that it was possible to build a taxiway adviser in order to optimize the ground traffic on busy airports such as Roissy Charles de Gaulle and Orly. It can be noticed that the modeling was easily improved with new runways on Roissy Charles De Gaulle, different speeds, uncertainties on speeds etc... without changing the algorithm itself. Genetic Algorithms are very efficient on the problem as they search the global optimum of the problem whereas a deterministic algorithm such as a 1-to-n strategy causes more delay.

Further work will concentrate in improving the global criteria for Genetic Algorithms, taking into account for example take off sequencing needs of approach sectors or priority levels for slotted departures.

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.

[MJ96] Victor M. Jimenez and Andres Marzal. *Computing the K Shortest Paths : A New Algorithm and an Experimental Comparison*.

[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.

Biography

Jean-Marc Alliot graduated from the Ecole Polytechnique de Paris and from the Ecole Nationale de l'Aviation Civile (ENAC). He also holds a Ph.D. in Computer Science (1992). He is currently in charge of the global optimization laboratory of CENA and ENAC in Toulouse.

Nicolas Durand graduated from the Ecole Polytechnique de Paris and from the Ecole Nationale de l'Aviation Civile (ENAC). He has been a design engineer at the Centre d'Etudes de la Navigation Aérienne (CENA) since 1992 and holds a Ph.D. in computer Science (1996).

Erwan Page graduated from the Ecole Nationale de l'Aviation Civile (ENAC). He has been in the design of Flow Management and Decision Support tools for French ATC for 15 years and is head of the "Aéroport, Tour et zone Terminale" (ATT) division at CENA/Paris.

Jean-Baptiste Gotteland graduated from the Ecole Nationale de l'Aviation Civile (ENAC). He is completing a Ph.D. in Computer Science at the Institut National Polytechnique de Toulouse.