Handling CFMU slots in busy airports

Jean-Baptiste Gotteland, Nicolas Durand, Jean-Marc Alliot

Handling CFMU* slots in busy airports

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

Abstract

In busy airports, too many departing flights take off out of the time slot required by the Central Flow Management Unit (CFMU). During ground traffic peaks, taxi out times and runway queuing delays increase, and it seems very hard and maybe unfeasible for ground controllers to organise properly the traffic so that aircraft take off at their scheduled time.

In a first part, this paper shows how a ground traffic simulator that includes a conflict resolution module can give accurate predictions of takeoff times. Then, different strategies taking into account these predictions so as to incorporate CFMU slot constraints are compared.

The deviation between the required slots and the final takeoff times is measured. The efficiency of the different strategies is compared by the correlation between the number of aircraft and the generated delay.

1 Introduction

Refering to the European Central Flow Management Unit (CFMU), a takeoff slot can be assigned to a departing flight in order to avoid congestion in some particular airspace. The given slot consists in a delayed time t and the aircraft involved is really expected to take off in the 15 minutes time range $[t-5; t+10]$.

It is important to underline the difficulties that ground controllers can meet to enforce these slots constraints. First, aircraft are frequently delayed when leaving the gate or taxiing. Next, the runway is sometimes shared with landing traffic and a minimal time is required between each takeoff or landing, so that, most of the time, departing aircraft must queue up before having access to the runway. When all these delays are added to any other kind of disturbances, it can become quite imposible to predict with a satisfying accuracy the time needed by an aircraft to reach the runway and take off. Thus, it is understandably difficult to really optimize the runway sequence as well as to deal with some required slots.

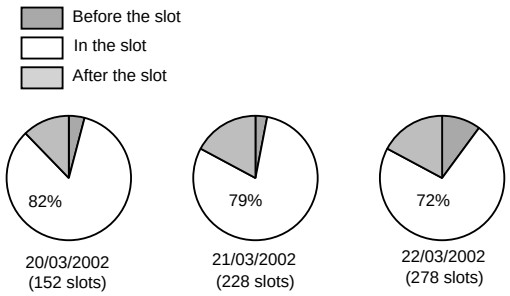

Figure 1: Actual respect of CFMU slots

For instance, figure 1 gives the repartition of constrained departures that took off before, in and after their scheduled CFMU slot. These radar tracks data from Roissy Charles De Gaule airport show that too many aircraft (between 20% and 30%) take off earlier or later than expected.

The aim of this study is to determine whether a ground traffic simulator can provide accurate take off times, so as to comply with the CFMU slots constraints in the global taxiing and runway sequencing resolution process.

2 Backgrounds

2.1 General description of the simulator

The ground traffic simulator used for this study is associated with a detailed airport modeling to assign a set of realistic paths to each aircraft.

Traffic prediction is regularly computed according to a given speed uncertainty. Algorithms estimate the best manoeuvers that can be executed by each aircraft to ensure taxiing separations and runway sequencing while optimizing a global criteria such as the total delay.

*Central Flow Management Unit

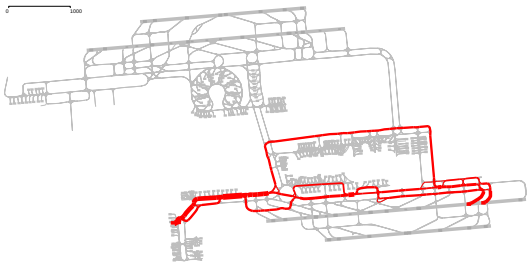

Figure 2: Airport graph

This part briefly sums up the main topics concerning the ground traffic simulator. More specific details can be found in [GDA01].

2.1.1 Problem modeling

The simulator inputs are the airport topological description, the aircraft types with their corresponding weight categories, and a one day traffic sample at the airport.

A flight is described by its recorded flightplan, containing the departure or arrival time, the runway and the gate used, the aircraft type, and sometimes a CFMU slot as a required takeoff time.

The aircraft type information enables the evaluation of the needed takeoff or landing distance, in order to select the possible runway entry or exit points. Moreover, it is also associated with a wake turbulence category, which determines the time separation between two aircraft (one, two or three minutes) on the runway.

The airport description is used to fix a realistic set of paths for each aircraft when its flightplan becomes active. Thus, the airport is described by a graph linking its gates, taxiways and runways. Each link is assigned a cost, which is an evaluation of the time spent by an aircraft when it proceeds via the corresponding taxiway, added to some particular penalty related to runway areas, gate access or forbidden directions.

The Dijkstra algorithm [AMO93] is combined to a Recursive Enumeration Algorithm [JM99] to compute a reduced and acceptable set of different paths (figure 2).

2.1.2 Simulation steps

The simulator works with a prediction time window T_w shifted every Δ minutes.

When the simulated current time is t, the flightplans expected to land or to leave the gate in the time range $[t; t + T_w]$ are activated: a set of appropriate paths is assigned to each of them and the resulting best trajectories are added to the ones of the already taxiing aircraft.

The conflict detection and resolution is performed in the time window, and the resulting manoeuvers are applied to build the new situation Δ minutes later. The problem is then reconsidered at the new simulation step $t + \Delta$.

2.1.3 Speed uncertainty

Each aircraft trajectory is predicted with a given speed uncertainty, fixed as a constant percentage of the nominal speed (which is function of procedures and turning rate).

Therefore, an aircraft is considered to occupy several possible positions (included in a segment) at a given time.

2.1.4 Conflict detection

The separation rules are defined as follows:

- aircraft on gate position are separated with all other aircraft.
- The distance between two taxiing aircraft must never be less than 60 meters.
- A time separation of 1, 2 or 3 minutes 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.

When the traffic prediction doesn't respect one of these rules, the two aircraft involved are said to be "in conflict".

2.2 Conflict resolution methods

2.2.1 Aircraft possible manoeuvers

In order to ensure separations, aircraft trajectories can be modified by ground control orders. For each aircraft, a control order is defined by:

- The path that the aircraft must follow, chosen among the set of remaining possible paths for this aircraft;

- Eventually, a position where the aircraft must wait and a time ending this wait.

Thus, the length of the segment representing the several possible positions of an aircraft is reduced when and where the aircraft is expected to wait, as the reference is a precise position and a precise end waiting time.

For an arriving aircraft, the waiting position can be in the air (this means that the aircraft is asked to slow down before landing). In this special case and to keep the simulation realistic, the delay can't exceed λ seconds (usually $\lambda = 30$ seconds).

2.2.2 The 1 against n method: BB

This method deals with a simplified problem, which is to find the best solution for one aircraft to avoid other ones (already handled). Thus, all taxiing or scheduled aircraft are sorted and solved one after the other (this means that the first ones have priority on the last ones).

A graph exploration algorithm (Branch and Bound [HT95]) can quickly find the best path and the best wait for the considered aircraft (if a solution exists).

Of course the efficiency of this method is largely affected by the priority order assigned to the aircraft (most of the configurations don't even have a solution). In the last version, aircraft are initially sorted out according to their expected runway access time. When the algorithm doesn't find any solution for an aircraft, the classification is modified (this aircraft is given a higher priority level) and the process is restarted.

2.2.3 Genetic Algorithms: $GABB$

In this method, classical Genetic Algorithms as described in the literature [Gol89, Mic92] are used to choose the path and the priority level of each aircraft. For a situation involving N aircraft, an element of the population (chromosome) is described by:

$$(n_i, p_i)_{1 \leq i \leq N}$$

where n_i is the path of the aircraft i and p_i its priority level.

The Branch and Bound algorithm (BB) described in the previous section (limited to one path per aircraft) is used to compute the feasibility and an evaluation for each element of the population. The fitness function f is therefore given by:

- If there are $n_c > 0$ unsolved aircraft:

$$f = \frac{1}{2.n_c}$$

- When all the aircraft are solved:

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

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

In this way, the evaluation is always better ($f > 0.5$) when the chromosome describes a solved situation.

Moreover, the fitness function is partially separable as defined in [DA98, DAN96] so that the classical adapted crossover and mutation operators can be applied.

3 Takeoff time predictability

3.1 The needed anticipation

The problem is to estimate at each simulation step t_0, the takeoff time of each active departing aircraft. An aircraft is active if it is already moving at t_0 or if it will be moving in the prediction time range $[t_0; t_0 + T_w]$.

Applied to the question of total delay minimization, previous simulations ([GDA01]) have already established that both BB and Genetic Algorithms had their efficiency increased with a reduced time window ($T_w \leq 10$ minutes). When speed uncertainty is fixed to $\pm 10\%$, it is not even realistic to solve the situation in a time window exceeding 5 minutes, as the set of the possible positions for an aircraft rapidly becomes a very long segment (over 500 meters, as the growing rate of this segment is then 120 meters per minute for an aircraft moving at 10 m/s).

However, such a reduced time window (5 minutes) is clearly not enough to predict with a good accuracy all taxi out time variations due to incoming departing and arriving traffic flows. Therefore, the solution is to take into account incoming flightplans starting in the next $[t_0 + T_w; t_0 + \alpha]$ time interval ($\alpha \gg T_w$) in order to estimate correctly each runway sequence. This new parameter α is called **takeoff time anticipation** and will be used only for takeoff times predictions.

Two different approaches trying to estimate runways sequences and takeoff times are developed and compared below.

In order to measure the prediction accuracy of these approaches, the anticipated takeoff times calculated at each simulation step t_0 are memorized. After computing the predictions, the usual conflict resolution method ($GABB$) is applied to the restricted time range $[t_0; t_0 + T_w]$, with the nominal speed uncertainty value. The resulting manoeuvers are applied to forward the simulation step to $t_0 + \Delta$. When an aircraft takes off in the simulation, its relative predictions and its final takeoff time are stored in the output file.

3.2 The runway sequence approach

3.2.1 Principles

This approach focuses on runway occupancy and makes the hypothesis that it will be maximal: the aim is to sort in the same queue and in an adapted order all the active aircraft and the incoming flightplans associated with a given runway.

Feasible runway slots are then assigned sequentially, following the order of the queue, and considering that aircraft will take off or land as fast as possible.

3.2.2 Implementation

In order to sort out every active aircraft and the incoming set of flightplan, a runway access time Γ_i will be assigned to each of them.

As arriving aircraft can't be delayed more than λ seconds, they must be inserted into the departing aircraft queue in a feasible slot. As a consequence, the evaluation (Γ_i) is divided in four steps:

1. A first estimation of takeoff times (Γ_i^0) is computed considering the direct trajectories (without any delay) of departing aircraft.

2. An intermediary takeoff slot Γ_i^1 is assigned sequentially to each departure, considering that the runway occupancy will be maximized and following the order defined by (Γ_i^0).

3. Arriving aircraft (if there are some) are inserted into the departing aircraft queue with respect to their required landing time (λ_i).

4. Resulting takeoff time (Γ_i) are finally computed, considering the global runway queue ordered by $(\Gamma_i^1) \cup (\lambda_i)$.

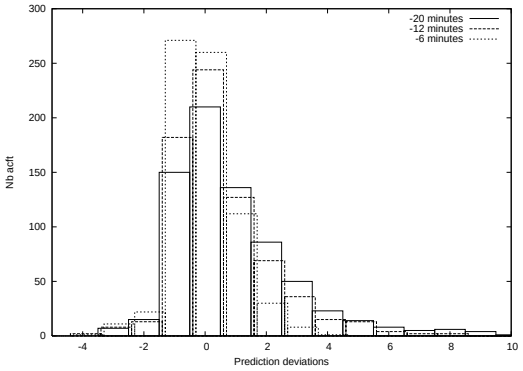

Figure 3: Runway sequence approach

3.2.3 Simulation results

The simulation was carried out on Roissy Charles De Gaule airport with a heavy day traffic sample (22/03/2002) and the following parameters:

- Traffic sample: 1433 flightplans, 695 departures.
- Solver: $GABB$
- Time window: $T_w = 5$ minutes
- Takeoff time anticipation: $\alpha = 30$ minutes
- Simulation step: $\Delta = 2$ minutes
- Speed uncertainty: $\delta_s = \pm 10\%$

Figure 3 gives the deviations distribution for different predictions (6, 12, and 20 minutes earlier). It appears that the deviations are always in the range $[-4; +10]$ minutes, even if only 30% of them are really exact.

3.3 The simulation based approach

3.3.1 Description

In this approach, the idea is to ignore speed uncertainty in order to apply directly a simplified conflict resolution method to the extended time range $[t_0; t_0 + \alpha]$.

The resulting runway aircraft queues will be extracted from this global resolution.

3.3.2 Implementation

At each simulation step t_0 (every Δ minutes), each flightplan starting before $t_0 + \alpha$ is activated: a set of possible paths is computed for the aircraft, and the direct

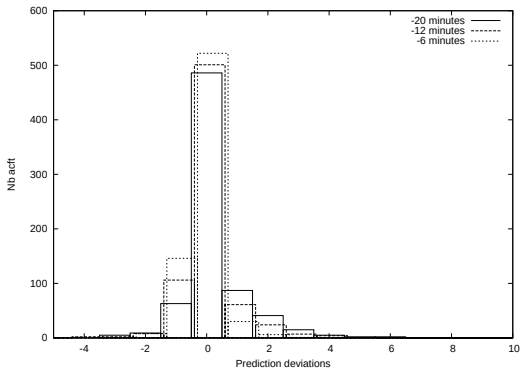

Figure 4: Simulation based approach

trajectory (without any delay) is included in the traffic prediction.

The Branch and Bound conflict resolution algorithm (BB) is applied to the extended time range $[t_0; t_0 + \alpha]$ without considering any speed uncertainty.

The resulting aircraft sequence for each runway is deduced from this BB resolution and the predicted takeoff times of departing aircraft that are moving in the time interval $[t_0; t_0 + T_w]$ are memorized.

3.3.3 Simulation results

The simulation was carried out with the same traffic sample and the same parameters than in the runway sequence approach.

Figure 4 gives the different distributions for each prediction. We can observe that the deviation is always in the range $[-4; +6]$ minutes and that 71% of the prediction is exact. Moreover, more than 90% of the deviations is in the range $[-1; +1]$ minutes.

3.4 Conclusions

The first observation is that the deviations of the two different prediction methods (20 minutes earlier) are included in the CFMU required precision ($[-5; +10]$ minutes). This is interesting as the $GABB$ resolution algorithm was absolutely not correlated with the predictions made at each simulation step: the global criteria to minimize was the total delay but not the deviation to the predictions.

The next point to underline is that the second approach is far more precise: as already shown in lot of studies ([IDA98] for instance), taxi out times are not a simple function of runway queuing delays, but are largely affected by arriving traffic near runways and near gates.

As a consequence, the second takeoff time prediction method will be selected in the next part. Takeoff time predictions will be correlated with the $GABB$ algorithm in order to take into account CFMU slot constraints.

4 Handling CFMU slots

4.1 Introduction

4.1.1 Operational issues

Two main difficulties concerning the respect of CFMU slots must be underlined:

- As the CFMU slot is a delayed takeoff time (sometimes more than 1 hour after the initially asked one), the first difficulty is to find an appropriate waiting position for the aircraft. For the crew and the passengers, the most convenient position is obviously at the gate. However, such an unpredicted gate occupancy is not always possible, and once passengers are on board, they cannot be disembarked anyway as the aircraft has to stay ready.

- During peak periods, access to runways means queuing. Each aircraft has to be given a very precise position in the awaiting queue. If an aircraft reaches the runway threshold too soon, it will have to hold on untill it slot comes, and the other aircraft will be stuck in the queue. Conversely, if the aircraft reaches the runway too late, the slot is missed. In that case, it theoretically should ask and wait for a new CFMU slot. Both cases currently happen and cause important unexpected delays and ATM unefficiency.

Theses difficulties urge airport designers to develop some special waiting areas near runways (see figure 5). These areas are frequently used by controllers to elaborate runway sequences dynamically, taking into account the needs of approach sectors and of course CFMU slot constraints.

Another way to improve the scheduling of CFMU slots is to keep some specific runway access free for the concerned departures. Of course, this is only possible on large airports, where infrastructure offers several runway entries.

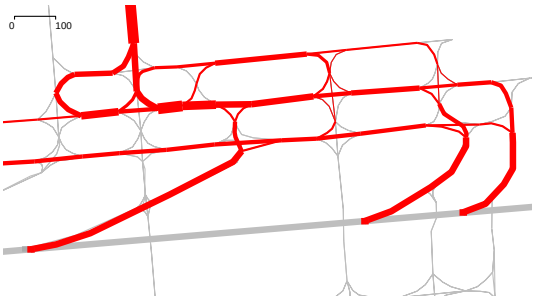

Figure 5: Waiting areas

4.1.2 The simulation context

The aim is to incorporate the CFMU slots constraints in the ground traffic simulator and to evaluate the ability of the resolution algorithm to deal with the described operational concepts.

As waiting areas take part in the topological description of the airport, they are directly included in the set of alternative paths proposed to an aircraft (see figure 5). The resolution algorithm can already assign some waiting positions in these areas when they contribute to optimize the global criteria.

Thus, the CFMU slot problem will be implemented in the ground traffic simulator as follows:

- The first topic is to take advantage of the available takeoff time predictions by estimating a *required wait* for each constrained departure. This implies some modifications of the BB (1 against n algorithm) itself, as the first interest of an aircraft can be to hold position.
- Then, the global criteria for the $GABB$ optimization method will be refined, in order to include the minimization of the deviations to CFMU slots as a priority.
- Finally, the runway access strategies will be implemented in conformity with Roissy Charle De Gaule facilities.

4.2 Required wait

4.2.1 Definition

At each simulation step and for each departing aircraft i that is constrained to a CFMU slot T_i, the resolution algorithm needs an evaluation of the *required wait* τ_i, which is the period of time the aircraft should wait before trying to reach the runway, in order to take off exactly in its slot.

An approximation of this *required wait* is simply given by:

$$\tau_i = Max(0, T_i - \Gamma_i)$$

where Γ_i is the takeoff time prediction computed in 3.3.

4.2.2 The adapted Branch and Bound

Considering the *required wait* τ_i of a slot assigned departure i, the Branch and Bound algorithm must be modified: in the set of all separated trajectory Ω_i, the best trajectory θ_i is the one that minimizes $|\tau_i - delay(\theta)|$, where $delay(\theta)$ is the delay resulting from the trajectory θ:

$$\theta_i = Argmin_{(\theta \in \Omega_i)}\{|\tau_i - delay(\theta)|\}$$

As a consequence, the Branch and Bound algorithm is applied to a modified graph:

- A node of the graph is a current delay d of the aircraft on its n_{th} path at time t.
- If a node represents a conflicting position with an already solved aircraft, it has no son.
- Each non conflicting node has two sons:
  - The first son is the same delay d in the same path n at time $t+1$ (the aircraft go forward). If $d < \tau_i$, the cost to reach this son is 2, otherwise 1.
  - The second son is the delay $d+1$ at time $t+1$ (the aircraft holds position at time t). If $d < \tau_i$, the cost to reach this son is 1, otherwise 2.
- The root nodes are a null delay on each alternative path of the aircraft at current simulation step t_0.
- The terminal nodes are the ones describing a non conflicting position of the aircraft at time $t_0 + T_w$.

4.3 The refined global criteria for $GABB$

The global criteria to be minimized for N aircraft has the following expression:

$$f = \sum_{i=1}^{N} f_i$$

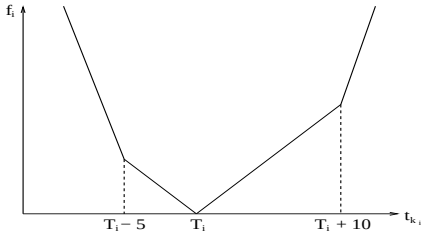

Figure 6: Local criteria for constrained flights

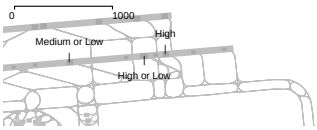

Figure 7: CAT strategy

where f_i is the local criteria corresponding to aircraft i.

In order to take into account CFMU slots, the feasibility of an element of the population is estimated by the adapted Branch and Bound algorithm described in the last part.

The evaluation of a solution is also modified: for an aircraft i which has a slot and which is solved (separated from other aircraft), the new local criteria will be a function of the estimated takeoff time t_{k_i} relative to the current resolution, and the CFMU slot T_i, as defined figure 6.

4.4 Assigning runway entries

Usually, the runway entries recommended to an aircraft depends on its wake turbulence category (low, medium, or high): thus, ground controllers can insert a light or medium aircraft category (which needs only 1 minute for takeoff) before a high aircraft category.

It also seems useful to assign some special runway entries to departures that are constrained to a CFMU slot. As they must takeoff at a very precise time, they can be sorted in the order defined by their slot on these runway access.

In the next simulations, two runway access strategies are compared, specifically adapted to Roissy Charle De Gaule airport:

- CAT: this runway access strategy only takes into account aircraft categories (see for instance figure 7);

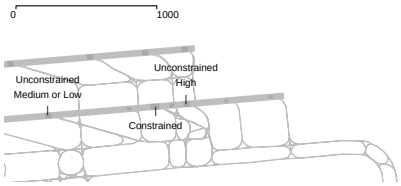

Figure 8: $SLOT$ strategy

- $SLOT$: this strategy also takes into account if the aircraft is constrained by a CFMU slot (see figure 8).

4.5 Simulations results

4.5.1 Strategies

Four strategies taking into account CFMU slots constraints are compared:

- $BB.CAT$ and $BB.SLOT$: BB resolution algorithm with the two different runway access strategies.
- $GA.CAT$ and $GA.SLOT$: $GABB$ algorithm with the two different runway access strategies.

4.5.2 Parameters

The simulations where carried out on the same heavy day traffic sample (22/03/2002) with the following parameters:

- 1433 flights, 695 departures, 278 slots
- Time window: $T_w = 5$ minutes
- Takeoff time anticipation: $\alpha = 30$ minutes
- Simulation step: $\Delta = 2$ minutes
- Speed uncertainty: $\delta_s = \pm 10\%$

4.5.3 Results

Figure 9 gives the distribution of the deviations between takeoff times and CFMU slots for each strategy.

The first point to notice is the similarity of CAT and $SLOT$ runway access strategies. For both BB and GA algorithms, it doesn't seem relevant to assign some specific runway entries to constrained aircraft. This means that when some accurate predictions are available, a

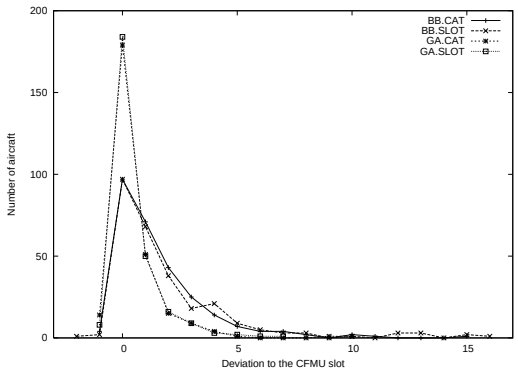

Figure 9: Slot deviations distributions

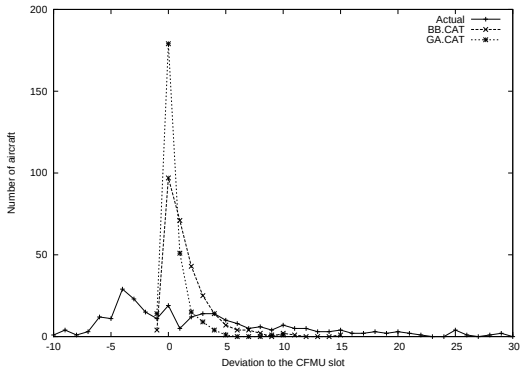

Figure 10: Comparing with actual distribution

concrete dispatching of aircraft is not really necessary to handle CFMU slots (at least for a computer).

The second point is the superiority of Genetic Algorithms on the Branch and Bound method. The deviations of the two GA strategies are always included in the interval $[-1; +9]$ minutes, and more than 80% of the flights are sloted in a $[-1; +1]$ time interval.

For information, Figure 10 makes the comparison of the deviations for the simulated and the actual traffic. The range of the real deviations extends to [-33; +37] minutes.

Figure 11 gives the correlation between the number of taxiing aircraft and the generated delay for each strategy. These figures only concern unconstrained aircraft, and put in light the differences between CAT and $SLOT$ runway access strategies: CAT is clearly more efficient as far as the total delay is concerned. The reason is obviously that the runway access restrictions defined in the $SLOT$ strategy are very penalizing.

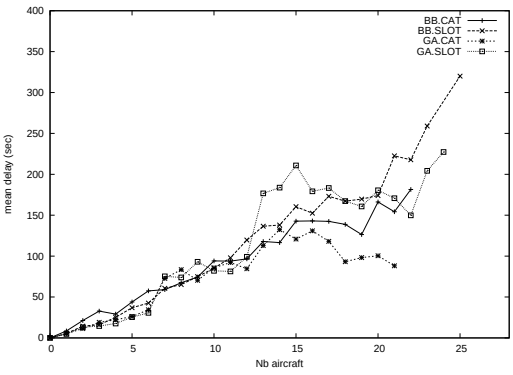

Figure 11: Unconstrained flights delays

5 Conclusions

The first simulations show that during traffic peaks, the runway sequences are largely affected by what can happen around gates areas and taxiway intersections. Hence, a ground traffic simulator taking into account both arriving and departing traffic can be very useful to provide accurate predictions for takeoff times, as well as to optimize the runway sequences.

Moreover, last simulations show how such a simulator could help ground controllers to schedule all constrained flights in their CFMU slot. The resolution method can be linked to the takeoff time predictions. It is therefore possible to anticipate the required wait of each constrained flight in order to insert the aircraft into a satisfying position into the runway queue. Applied to this problem, Genetic Algorithms (as opposed to the classical 1 against n method) give some very interesting results: the optimization of a global criteria, including both the total delay and the CFMU slots constraints is completely feasible.

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.

[GDA01] Jean-Baptiste Gotteland, Nicolas Durand, Jean-Marc Alliot, and Erwan Page *Ground Traffic Optimization* In Air Traffic Management R & D Seminar, December 2001

[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.

[IDA98] 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 Air Traffic Management R & D Seminar, Orlando, December 1998.

[JM99] Victor M. Jimenez and Andres Marzal. *Computing the K Shortest Paths : A New Algorithm and an Experimental Comparison*. In Proc. 3rd Worksh. Algorithm Engineering, Springer-Verlag, 1999

[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.

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).

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.