Deconfliction with constraint programming

Nicolas Barnier, Cyril Allignol

Deconfliction With Constraint Programming

Nicolas BARNIER and Cyril ALLIGNOL
barnier@recherche.enac.fr,allignol@tls.cena.fr
École Nationale de l'Aviation Civile – DSNA/DTI/R&D
Toulouse, France

Abstract

Current European Air Traffic Control (ATC) system is far exceeded by the demand and the resulting delays are a financial and psychological burden for airlines and passengers. One of their main sources is the hourly capacity constraints, defined on each en-route ATC sector, but poorly representing the workload of controllers. Whereas previous works were mainly focused on optimizing the ground delay slot allocation process performed by the Central Flow Management Unit (CFMU) to meet these constraints, we propose to directly solve all conflicts occurring above a given flight level by ground delaying, while minimizing the maximal delay. We present a Constraint Programming (CP) model of this large scale combinatorial optimization problem and the results obtained by its implementation with the FaCiLe constraint library.

Keywords: Slot Allocation, Conflict Resolution, Constraint Programming

1 Introduction

In an already saturated European sky, the predicted growth of air traffic volume urges to improve Air Traffic Management (ATM) efficiency, as attested by the ACARE Strategic Agenda 2 [ACA04] and the European Single Sky programme SESAR. Current ATM optimization strategies, like reducing the size of sectors or the distance of separation (RVSM, P-RNAV), seem to have reach the structural limits of the system, while the automation of Air Traffic Control (ATC) has known few significant improvements over the last decades [GD05].

The Central Flow Management Unit (CFMU) in Brussels is in charge of reducing these congestion costs by, among other strategic or tactical measures, delaying departure slots for the flights involved in overloaded en-route sectors. The purpose of delaying is to respect the en-route capacity constraints provided by each ATC Centre (ATCC) as a number of aircraft per hour, according to their daily schedule. Former studies like [DPJL97, BBR01] aimed at optimizing this process over the greedy algorithm used at the CFMU. However, one of the limitations of this process is that the definition of sectors capacities (hourly rate of aircraft entering the sector) is poorly related to the complexity of the traffic with respect to the controllers workload, as assessed by [GG07].

Instead of trying to satisfy en-route capacity constraints, we propose to directly solve the potential conflicts occurring between any two intersecting trajectories with ground delaying. A single delay would be associated with each flight such that all potential conflicts occurring above a given flight level would be avoided. This very fine grain model would of course generates much larger constraints sets than the macroscopic (at the sector level) capacitated ones, but would garantee conflict-free trajectories all along the flight path... provided that aircraft were able to scrupulously follow their predicted route in the four dimensions.

Obviously, the latter hypothesis is far from being met nowadays, even if a lot of studies about ATC automation have demonstrated that the accuracy of Flight Management Systems is a crucial issue for future Air Traffic Flow Management (ATFM) and ATC systems, as advocated by [AC03]. Nevertheless, we believe our approach may reduce air traffic complexity by "deconflicting" it in advance, the remaining conflicts due to deviation from the flight plan or occurring in the lower airspace being taken care of by more standard ATC procedures.

Several optimization paradigms are being evaluated for this purpose, namely meta-heuristics, local search and Constraint Programming (CP). We will focus here on the CP approach as it offers to obtain proved bounds on the maximal delays needed to solve the conflicts, which can be used to draw conclusions on the feasibility of this kind of regulations. Moreover, CP is a technology of choice for implementing such preliminary work, as it allows to easily refine the problem by adding new constraints (*e.g.* connection constraints between flights using the same aircraft) and to experiment with various search strategies without changing the rest of the model.

In the following sections, we first briefly present ATC and ATFM, focusing on ground delaying. Then we describe our model of a conflict-free slot allocation, starting by the details of the conflict constraints generation and search strategy, which lead to the presentation of our first results on a few instances for which optimal solutions were found. We end with planned further works to enhance the approach before concluding.

2 Context

2.1 ATC and ATFM

Air Traffic Control (ATC) is a ground-based service provided to ensure the safety and efficiency of the flow of aircraft. The first goal of ATC is to maintain aircraft *separated*: outside Terminal Areas (TMA) around airports, two aircraft should remain distant from each other at least by 5 Nm horizontally and 1000 ft vertically, as illustrated by the safety volume of figure 1.

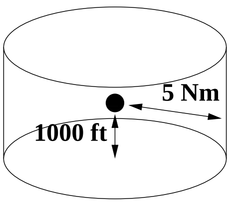

Figure 1: Vertical and horizontal separation. Another aircraft cannot be inside the cylinder at the same time.

The overall system currently implemented to achieve this goal can be conceptually divided in several layers or filters with decreasing time horizon with respect to the flight date of the traffic concerned:

1. *Strategic* (several months), ASM (Air Space Management): design of routes, sectors and procedures (*e.g.* reduced separation RVSM since 2002, Area Navigation (RNAV) with fictive beacons...).
2. *Tactical* (a few days to a few hours), ATFM: ATC Centres opening schedules define hourly capacities of each open sectors (or groups of sectors). To respect these capacity constraints, the CFMU computes and updates flow regulations and reroutings according to the posted flight plans and resulting workload excess.
3. *Real time* (5/15 min), ATC: surveillance, coordination with adjacent centres, conflict resolution by various simple manœuvres (heading, flight level, speed) transmitted to the pilots.
4. *Emergency* (less than 5 min), safety nets: ground-based (Short Term Conflict Alert, Minimum Safety Altitude Warning) and airborne (Traffic Alert and Collision Avoidance System, Ground Proximity Warning System).

We will focus in the following section on the kind of regulations performed by the CFMU by postponing the takeoff of aircraft.

2.2 Ground Delays

As aircraft obviously cannot be paused while airborne whenever the traffic complexity becomes to high to be safely handled by a controller, one of the simplest way to leverage ATC workload is to postpone the takeoff of aircraft[1]. This kind of measure is however quite unpopular among airlines, as it can be very costly and may propagate in terms of missed correspondances (see [Uni04]), so the delays should be minimized as much as possible.

2.2.1 Satisfying En-Route Sectors Capacity Constraints

The aim of CFMU regulations is to maintain the number of en-route aircraft entering a given subset of sectors below some bound over given time periods (usually one hour), according to the constraints declared by experts (FMP) in each ATCC for the day of traffic. The CFMU experts first identify the overloaded sectors and responsible flows with the PREDICT tool, then compute a slot allocation as ground delays for the involved flights with the CASA tool (cf. [CFM00]).

CASA is able to take into account many operational constraints and updates to optimize its allocation process, but the algorithm used has greedy properties and thus cannot guarantee to find a correct solution (which satisfies all the constraints) or

[1]Note that flights might be delayed for other reasons than en-route capacity violation, like bad weather or equipment failures.

an optimal one. CP technology has been applied with good results to prove and optimize the allocation process with a relaxed model [DPJL97] or to smooth the resulting load profiles [BBR01] with a tighter model.

However, traffic complexity is very hard to define precisely, and sector capacities, expressed as a maximum number of aircraft entering the sector over a given time period, does not take into account many parameters relevant to accurately represent the performance of ATC. Observed actual capacities, as well as merging and splitting subset of sectors, symptomatically present very different profiles than the predicted ones.

To overcome this issue, recent works such as [FPA$^+$07] use a much more precise and complex workload CP model to dynamically balance the traffic over the sectors of an ATCC in the upper airspace. Other works, like [Bar02] uses CP technology as well to optimize the ATCC opening schedules to match the predicted traffic more closely, or even attempt to redesign airspace sectorisation with better balancing like [TB03].

2.2.2 Solving the Conflicts

Within the SESAR programme, the trend currently is the design of conflict-free 4D tubes within crowded airspace (whereas separation could be delegated to aircraft in less dense areas). So instead of only respecting sectors capacities macroscopically, we propose to evaluate the cost of precisely solving all potential conflicts, only with ground delays, while minimizing the worst allocated delay to maintain equity among airlines.

This conflict-free model will of course yield much larger problem instances as all the conflicting trajectories intervals above a given flight level will be taken into account as constraints. The resulting problem is intrinsically disjonctive as for each potential conflict between two flights i and j, either i must precede j or j precede i at each pair of points concerned (see section 3.2).

It is expected that such regulations will result in too high a total amount of delay, as we will optimize the maximum delay only, minimizing the sum of delays being exponentially more search intensive. However, the use of CP technology will provide proved bounds that other optimization techniques like local search or meta-heuristics cannot produce.

Other approaches have been presented to solve conflicts in real-time, automating the task of controllers. Some of the most promising ones are centralized techniques that compute simple horizontal or vertical manœuvres [Gra02] and small speed adjustements as proposed by the ERASMUS project [Arc04]. These solvers, based around a meta-heuristic (Genetic Algorithm), can take uncertainties on ground and vertical speed into account and repeatedly compute solutions for a sliding time window.

3 Conflict-Free Slot Allocation

3.1 Conflict Detection

Our input data are provided by the CATS simulator [ABDM97] which takes all filed flight plans concerning the French airspace for a given day of traffic and the relevant airspace configuration (sectors, waypoints...), and outputs the corresponding 4D trajectories. Trajectories are sampled with a 15s time step, which is the largest interval to guarantee that at least two points of the trajectories of facing aircraft at the highest possible speed will be closer than one separation norm, *i.e.* even the shortest conflicts will be detected.

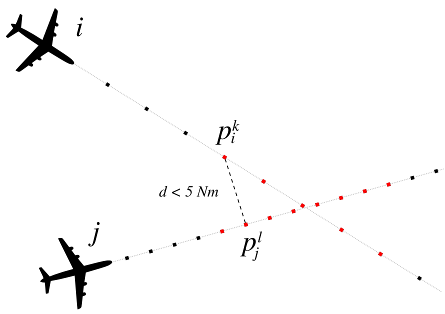

Figure 2: Conflicting Points Detection

Trajectories are then probed pairwise for potential conflicts, taking the maximal allowed delay into account. The separation norm is thus tested for each pair of points of the two probed trajectories (up to $p = 900$ points per trajectory for up to $n = 9500$ flights in $\mathcal{O}(n^2p^2)$) as illustrated on figure 2 in the horizontal plane.

Though the maximal allowed delay can be seen as a parameter of the search algorithm only, it also affect the conflict detection. Actually, when the maximal allowed delay is increased, the size of the problem grows as well, as more and more flights tend to be in potential conflict. Ultimately, if a 24h delay would be allowed, the conflict detection could be done in 3D, regardless of time, as any two geometrically conflicting trajectories would generate a constraint. So, whenever a particular instance has been proved inconsistent, it has to be generated again with higher values of the maximal delay, which will capture later potential conflicts on the trajectories pairs.

Operationally, flights originating outside the Eurocontrol countries cannot be delayed and we will thus fix their delay variable to 0 in our constraint model, reducing the number of variables but tightening the constraints as well and offering less opportunity for optimization. Constraints corresponding to conflicts occurring between two such flights will of course be discarded as we cannot delay the flights to solve them. Such remaining conflicting cases would have to be taken care of by other ATC or ATFM techniques that we will not address in this study.

3.2 Constraints Model

3.2.1 Conflicts

To compute the constraints of our model, the trajectories are pairwise probed for couples of conflicting points. Given a flight i, we note the input data:

- $\{p_i^k\}$ the chronologically ordered sequence of the 3D points of its trajectory;
- t_i^k the time at which the flight will be at point p_i^k, should it not be delayed.

We define a set D of decision variables:

$$D = \{\delta_i, \forall i \in [1, n]\}$$

of finite domain $[\text{min\_delay}, \text{max\_delay}]$ that represent the delay associated with each of the n flights, and we will describe our model using the following auxilliary variables:

- $\theta_i^k = t_i^k + \delta_i$ the date at which flight i will be at point p_i^k if it is delayed by δ_i;
- $d_{ij} = \delta_j - \delta_i$ the difference of the delays of flight j and i.

For any geometrically conflicting points p_i^k and p_j^l such that the separation norm is violated (d_h being the distance in the horizontal plane and d_v in the horizontal plane):

$$d_h(p_i^k, p_j^l) < 5\,\text{Nm} \quad \text{and} \quad d_v(p_i^k, p_j^l) < 1000\,\text{ft}$$

we must temporally ensure that:

$$\theta_i^k \neq \theta_j^l$$

which can be rewritten with the difference variables d_{ij}:

$$d_{ij} \neq t_i^k - t_j^l$$

Starting at the first such point p_i^k that conflicts with a point of flight j, we take into account the whole continuous segment of trajectory j conflicting with p_i^k:

$$\{p_j^l, \forall l \in [l_k, l_{k+r}]\}$$

for some r, and we impose that:

$$\begin{array}{rcl} d_{ij} & \notin & \{t_i^k - t_j^l, \forall l \in [l_k, l_{k+r}]\} \\ d_{ij} & \notin & [lb^k, ub^k] \end{array}$$

with lb^k and ub^k being respectively the lower and upper bound of the set of $t_i^k - t_j^l$.

If the next point p_i^{k+1} of the trajectory of flight i conflicts with a further segment of flight j, we will obtain another forbidden segment for d_{ij}:

$$d_{ij} \notin [lb^{k+1}, ub^{k+1}]$$

taking part in the same potential conflict. To ensure separation we must then impose:

$$d_{ij} \notin [\min(lb^k, lb^{k+1}), \max(lb^k, lb^{k+1})]$$

as the conflicting segments of flight j overlap.

So if we take into account all the successive points of flight i, starting at p_i^k, that conflict with overlapping segments of flight j, up to some last point p_i^{k+s}, with $\underline{lb}_1 = \min\{lb^{k+u}, u \in [0, s]\}$ and $\overline{ub}_1 = \max\{ub^{k+u}, u \in [0, s]\}$ being the overall lower and upper bounds of the corresponding forbidden intervals for d_{ij}, we can define the first conflict between flights i and j:

$$d_{ij} \notin [\underline{lb}_1, \overline{ub}_1]$$

Note that we take as parameters of the problem instance the minimal and maximal allowable delays $\delta_i \in [\text{min\_delay}, \text{max\_delay}]$, therefore the domain of $d_{ij} = \delta_j - \delta_i$ is $[-w, w]$, with $w = \text{max\_delay} - \text{min\_delay}$. We simply discard the conflict whenever $\overline{ub} < -w$ or $\underline{lb} > w$.

A pair of flights may conflicts several disjoint times over their entire trajectories (as illustrated on figure 3), so several such disjoint intervals may be forbidden for the difference of their delays. For two flights i and j conflicting σ times over their entire trajectories:

$$d_{ij} \notin [\underline{lb}^1, \overline{ub}^1] \cup \cdots \cup [\underline{lb}^\sigma, \overline{ub}^\sigma]$$

or, rewritten as a disjonctive constraint over the decision variables:

$$\begin{array}{rcl} (-w \leq & \delta_j - \delta_i & < \underline{lb}^1) \vee \\ (\overline{ub}^1 < & \delta_j - \delta_i & < \underline{lb}_2) \vee \cdots \vee \\ (\overline{ub}^{\sigma-1} < & \delta_j - \delta_i & < \underline{lb}_\sigma) \vee \\ (\overline{ub}^\sigma < & \delta_j - \delta_i & \leq w) \end{array}$$

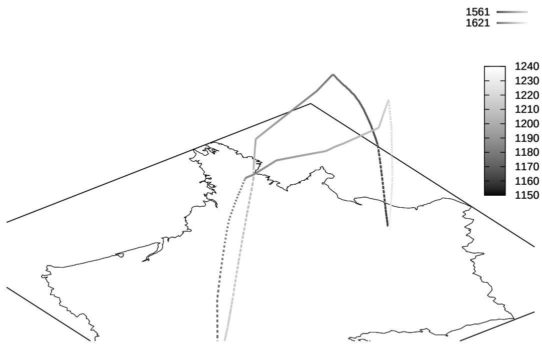

Figure 3: Three potential conflicts between to flights. The gray scale corresponds to time along the trajectory, the lighter the later.

provided that $\underline{lb}^1 > -w$ and $\overline{ub}^\sigma < w$, otherwise the first or last part of the disjonction is removed.

3.2.2 Cost

The cost of a solution is then defined as:

$$\text{cost} = \max\{|\delta_i|, \forall i \in [1, n]\}$$

Note that min_delay $= 0$ and max_delay > 0 for all the results presented in section 3.4 such that the absolute values were not needed for constraining the cost and such that $w =$ max_delay. Nevertheless, small negative values of the minimal delay could be of interest in an operational context, as aircraft can be ready for takeoff a few minutes before their slot in some cases.

We can notice as well that if the cost was defined as $|\max D - \min D|$, *i.e.* the range of the delay values, equivalent solutions would be obtained by translating the domain [min_delay, max_delay], all the constraints being defined over differences between the decision variables δ_i.

3.2.3 Other Parameters

The takeoff and landing part of trajectories are truncated around airports within a given radius (usually 10 Nm) as the traffic is considered handled with specific procedures by the Terminal Area control services in these zones.

After the computation of the conflict constraints, the whole instance is scaled down to a more reasonable time step (*e.g.* 1 min) than the 15 s used during conflict detection, ensuring that the original forbidden intervals are strictly included in the scaled ones.

Moreover, the flight level of the detected conflicts can be filtered, for example to only take into account conflicts occurring within the upper airspace (from FL290 and above). The minimal and maximal altitude of each conflict is recorded during the detection stage and a conflict is discarded if it entirely occurs below or above the specified airspace slice.

We allow as well to filter the time interval during which the conflicts may occur, taking the time bounds of the allowable delay into account. Any conflict strictly occurring outside the given time interval is then discarded.

For conflict constraints with several exclusion intervals, we add an extra parameter min_gap to prevent using very small allowed interval between two disjoints but very close conflicts. Such solutions would be too sensitive to small perturbations of the original schedule to be operationally robust.

Eventually, all the flights that do not have any conflict with any other flight are withdrawn from the instance.

3.3 Search Strategy

The constraints of the problem are reminiscent of the disjonctive mutual exclusion constraints modelling scheduling problems. At a coarse grain, we could consider each conflicting area as a machine on which to process two tasks of different lengths (depending on the speed of the aircraft). Several conflicts along a trajectory could be seen as the ordered tasks of a given job, as in the Jobshop Scheduling Problem (JSP).

However, the comparison does not hold much further. First, the time intervals between any two conflict tasks of the same trajectory is fixed, as only one delay variable is associated with each flight (unlike the JSP where all tasks are only related with precedence constraints). Secondly, to consider a potential conflict in 3D only, as the transitive closure of the overlapping conflicting segments, with task lengths proportional to the time spent by the aircraft within the area, is misleading. In this setting, the conflict associated with two catching-up flights on the same route would be the entire trajectory, preventing them from being airborne at the same time! Obviously, our model is much more precise and allows two aircraft on the same route being only separated by 5 Nm. Third, the number of “conflict machines”, if not quadratic in the number of “flight jobs” as it could ultimately grow for arbitrary instances, is quite huge anyway as shown on figure 5.

Nevertheless, the branching scheme of our search strategy is inspired by standard scheduling techniques, because the essentially disjonctive nature of our problem shares some issues with scheduling ones. Trying to start the search by labelling the delay variables δ_i would be highly inefficient, because the constraints are expressed over the differences d_{ij}. Much more filtering is obtained by feeding the propagation of the arithmetic constraints with new domain bounds for the d_{ij} auxilliary variables.

Similarly to some scheduling branching schemes, where tasks performed on the same machine are ordered pairwise (either task A precedes task B or B precedes A), we either add the constraint $d_{ij} < \underline{lb}$ or $d_{ij} > \overline{ub}$ in the case of a single conflicting interval. If there are several holes in the domain of d_{ij}, branching is repeated with the bounds of the remaining holes. The variable d_{ij} with highest sparsity, *i.e.* the smallest ratio between the domain size and the difference of the domain bounds, is chosen first for branching.

When all conflicts are ordered and there is no more holes in the domain of the d_{ij}, we start labelling the decision variables δ_i with a standard dom/deg selection heuristic.

After the first solution is found, the branch and bound algorithm then proceeds by dichotomy on the cost domain to find the optimal solution with respect to minimization of the maximal allocated delay.

3.4 Results

We have implemented this CP model with the FaCiLe library [BB01] and obtained the following results on various day of traffic of years 2006 and 2007, with up to 8 000 flights and 285 000 intersecting pairs of trajectories taken into account for the lowest minimal flight level we could solve. About 10% of the flights are non-European flights, so their delays are fixed to 0.

The resulting constraint graphs typically exhibit only one single large connected component of maximal degree that can be greater than 300, *i.e.* a single flight may conflict with more than 300 other flights. Large cliques can also be found, as large as involving more than 60 flights, which indicates the presence of very entangled and dense traffic areas. The hardness of the conflict constraints is quite unevenly distributed, with two peaks for very small and very big (w.r.t. the maximal delay) forbidden intervals.

We mainly tune the size of our instances by decreasing the minimal flight level for which the conflicts are taken into account, typically aiming at the upper airspace (above FL290) where most of the cruising traffic occurs. However, we were able to optimally solve one instance with 6 600 flights and 260 000 conflicting pairs, taking all conflicts into account for the whole day, regardless of the flight level. Figure 4 shows the number of flights of our instances as a fonction of the minimal FL. The plots present two parts, one very steep from the maximal FL to FL300, then flights add up at a slower rate in the lower airspace (plots are labelled with their date, *e.g.* "070123" for the day of traffic on the 23rd of january 2007).

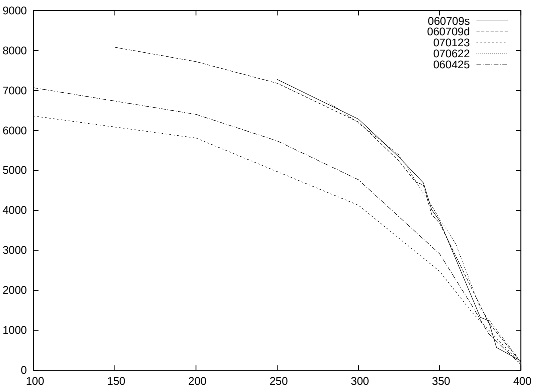

Figure 4: Minimal flight level vs number of flights.

The number of conflicting pairs is not quite quadratic with the number of flights, as mentioned in section 3.2 and shown on figure 5, but is quite huge anyhow, almost reaching 300 000 for our largest instance.

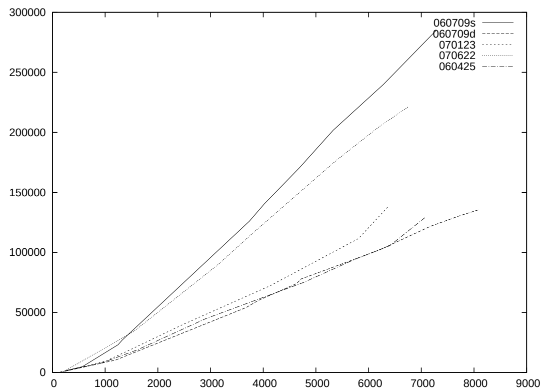

Figure 5: Number of flights vs number of conflicts.

We were systematically able to obtain optimal solutions within affordable computation times for all the instances where the 4 GB memory of our Core 2 Duo at 2.4 GHz were not exhausted. As shown in figure 6, small instances are solved in a few seconds whereas the biggest ones could take almost half an hour, growing only quadratically with the number of flights. We plan to address larger instances on a computer with more memory.

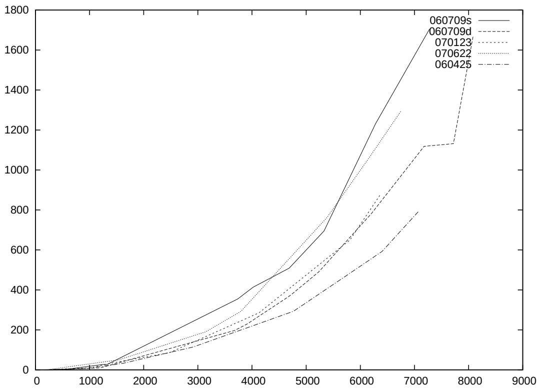

Figure 6: Number of flights vs computation time (optimality proof) in seconds.

However, the cost of this conflicts-free slot allocation can be quite high for the busiest days (our worst case is 162 min above FL250), but may be more reasonable (around 60 min) for less crowded days. Figure 7 shows that the cost grows steadily for small instances at high minimal FL, but jumps as soon as we add the main flows of traffic around FL350. The optimal cost then seems to be stable for larger instances, triggered only by the flights added around FL350. The corresponding means exhibit of course a smoother behaviour, but as we do not try to optimize the sum of delays, it is a poor indicator for our algorithm (and so is not represented on our graphs).

Note that for september 2008, Eurocontrol reported a mean delay of 25-30 min per delayed flights and a percentage of about 40-55% delayed departures. Some of these graphs show that some allocated delay may be as high as 90 min, but no precise figures are given on the maximal delay, though.

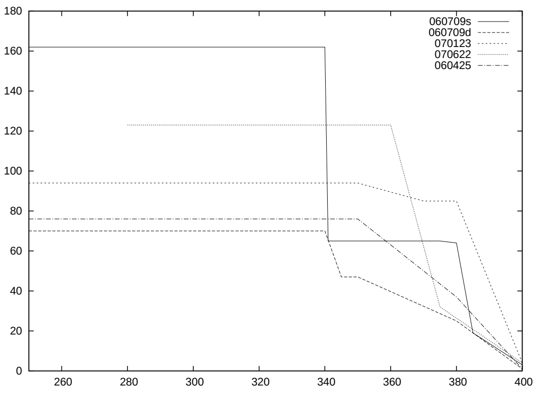

Figure 7: Minimal flight level vs optimal cost.

For one of the days of traffic (plots labelled "060709s" and "060709d"), we have also tested our model on *direct routes*, *i.e.* aircraft fly in straight line in the horizontal plane from origin to destination, disregarding the waypoints of their flight plan (which we call *standard routes*). Direct routes are the ideal trajectories for airlines, with respect to operational cost, but such a traffic would be hardly controllable for human operators and ATC would have to be fully automated in this context. However, they tend to generate constraint graphs with a much lower tightness, and it is interesting to observe that the bounds obtained with direct routes are much better than the standard ones (70 min vs 162 min above FL250 for the chosen day, with respectively 121 819 and 284 646 conflicting pairs). Flights following standard routes tend to be on closer trajectories, suitable for current ATC procedures to be efficient, but not using airspace to its full capacity.

4 Further Work

These first results are encouraging for the efficiency of CP technology applied to such large combinatorial optimization problems from ATFM. However, the use of other optimization techniques like Local Search (LS) or meta-heuristics could be needed to further optimize the total amount of delay generated. Several such solutions are being investigated, including an hybrid LS algorithm that uses CP to filter large neighborhoods.

4.1 Validation With CATS

We also plan to validate our model and algorithm by taking the generated solutions as input delays for the CATS simulator. If the model is correct, only the non-European conflicting pairs and the conflicts occurring below the chosen flight level should remain. CATS can also be tuned to add uncertainty on the ground and vertical speed of aircraft. The robustness of our solutions could then be checked, *e.g.* verifying that the min_gap parameter (see section 3.2) has an influence on the remaining number of conflicts in case of perturbations.

4.2 Sliding Time Windows

We've only addressed so far the resolution of conflicts within the French airspace alone. However, in a unified European ATC context, all conflicting traffic throughout the Eurocontrol countries should be taken into account. Such instances would comprise up to $30\,000$ flights per day.

To be able to address larger instances we plan to adapt our algorithm to repeatedly solve slices of the problem on a limited time window T_w, then to keep only the earliest part of the solution over a smaller interval λ and to slide the resolution window by λ. Parameters T_w and λ must be carefully chosen according to the computation time of the resolution. Similar approaches are used for dynamic conflict resolution in the CATS solver as mentioned in [GDA01].

4.3 Combined Flight Level Allocation

To be able to address such large instances as aforementioned, while maintaining reasonable maximal delay figures, we also plan to combine our slot allocation algorithm with a prior flight level allocation, possibly using CP technology as described in [BB02]. This first step, computed to minimize horizontally conflicting flows by separating them vertically (trying as well to deviate as little as possible from requested FLs), is expected to deconflict the traffic in a substantial amount before time slot allocation. The optimal cost should then remain within much better bounds than with the raw traffic.

5 Conclusion

We have presented a new ground delaying approach to solve all potential conflicts occurring above a given flight level for a day of traffic in the French airspace. Rather than trying to respect sector capacity constraints, we model each possibly conflicting situations between any two aircraft and impose delays to keep them separated, with the hypothesis that aircraft can precisely follow their planned 4D trajectories.

The resulting problem size is huge, but our CP algorithm is able to reach optimal solutions for all conflicts occurring inside the upper airspace. The resulting maximal delay can be comparable to delays allocated by the CFMU, but for the busiest days, solving all conflicts by ground delaying can be far too costly.

We plan to address larger (European) instances and reduce the costs of solutions with different techniques like repeatedly solving the problem on a sliding time windows and combining our delay algorithm with a prior flight level allocation.

Uncertainties remain a major issue in our approach, and our solutions still have to be checked on the CATS simulator to estimate their behaviour (remaining conflicts) with various degrees of noise in vertical and ground speeds.

References

[ABDM97] Jean-Marc Alliot, Jean-François Bosc, Nicolas Durand, and Lionnel Maugis. CATS: A complete air traffic simulator. In *16th DASC*, 1997.

[AC03] Jean-Marc Alliot and Dominique Colin de Verdière. Atm: 20 ans d'effort et perspectives. In *Symposium de l'Académie Nationale de l'Air et de l'Espace : vers l'automatisation du vol et sa gestion*, 2003.

[ACA04] ACARE. Strategic research agenda 2 (SRA 2). Technical report, Advisory Council for Aeronautics Research in Europe, 2004.

[Arc04] Nicolas Archambault. Speed uncertainty and speed regulation in conflict detection and resolution in air traffic control. In *ICRAT'2004*, 2004.

[Bar02] Nicolas Barnier. *Application de la programmation par contraintes à des problèmes de gestion du trafic aérien*. PhD thesis, Institut National Polytechnique de Toulouse, December 2002.

[BB01] Nicolas Barnier and Pascal Brisset. FaCiLe: a Functional Constraint Library. In *Colloquium on Implementation of Constraint and LOgic Programming Systems CICLOPS'01 (Workshop of CP'01)*, Paphos, Cyprus, December 2001.

[BB02] Nicolas Barnier and Pascal Brisset. Graph coloring for air traffic flow management. In *CPAIOR'02: Fourth Internation Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems*, pages 133–147, Le Croisic, France, March 2002.

[BBR01] Nicolas Barnier, Pascal Brisset, and Thomas Rivière. Slot allocation with constraint programming: Models and results. In *International Air Traffic Management R&D Seminar ATM-2001*, Santa Fe (NM), USA, December 2001.

[CFM00] Eurocontrol CFMU, Brussels. *Basic CFMU Handbook - General & CFMU Systems*, 6.0 edition, February 2000.

[DPJL97] M. Dalichampt, E. Petit, U. Junker, and J. Lebreton. Innovative slot allocation (ISA). Technical report, Eurocontrol, 1997.

[FPA+07] P. Flener, J. Pearson, M. Agren, C. Garcia Avello, M. Celiktin, and S. Dissing. Air-traffic complexity resolution in multi-sector planning using constraint programming. In Ch. Pusch and S. Saunders-Hodge, editors, *Proceedings of ATM'07, the 7th USA/Europe R&D Seminar on Air Traffic Management*, Barcelona, Spain, July 2007.

[GD05] Jean-Marc Garot and Nicolas Durand. Failures in the automation of air traffic control. In *Colloque de l'AAA*, 2005.

[GDA01] Géraud Granger, Nicolas Durand, and Jean-Marc Alliot. Optimal resolution of en route conflicts. In *ATM 2001*, 2001.

[GG07] David Gianazza and Kévin Guittet. Selection and evaluation of air traffic complexity metrics. In *25th DASC*, 2007.

[Gra02] Géraud Granger. *Détection et résolution de conflits aériens : modélisations et analyse*. PhD thesis, Thèse doctorat informatique de l'Ecole Polytechnique, 2002.

[TB03] Huy Tran Dac and Philippe Baptiste. Airspace sectorization by constraint programming. In *RIVF'03*, 2003.

[Uni04] University of Westminster. Evaluating the true cost to airlines of one minute of airborne or ground delay. Technical report, Eurocontrol, May 2004.