The Knowledge Engineering Review, Vol. 27:3, 291–307. © Cambridge University Press, 2012
doi:10.1017/S0269888912000227

Trajectory deconfliction with constraint programming

NICOLAS BARNIER[1] and CYRIL ALLIGNOL[2]

[1]*École Nationale de l'Aviation Civile, 7 Av. Édouard Belin, 31055 Toulouse, France;*
e-mail: barnier@recherche.enac.fr;
[2]*Direction des Services de la Navigation Aèrienne/Direction de la Technique et de l' Innovation, 7 Av. Édouard Belin, 31055 Toulouse, France;*
e-mail: allignol@tls.cena.fr

Abstract

As acknowledged by the SESAR (Single European Sky ATM (Air Traffic Management) Research) program, current Air Traffic Control (ATC) systems must be drastically improved to accommodate the predicted traffic growth in Europe. In this context, the Episode 3 project aims at assessing the performance of new ATM concepts, like 4D-trajectory planning and strategic deconfliction.

One of the bottlenecks impeding ATC performances is the hourly capacity constraints defined on each en-route ATC sector to limit the rate of aircraft. Previous works were mainly focused on optimizing the current ground holding slot allocation process devised to satisfy these constraints. We propose to estimate the cost of directly solving all conflicts in the upper airspace with ground holding, provided that aircraft were able to follow their trajectories accurately.

We present a Constraint Programming model of this large-scale combinatorial optimization problem and the results obtained with the FaCiLe (Functional Constraint Library). We study the effect of uncertainties on the departure time and estimate the cost of improving the robustness of our solutions with the Complete Air Traffic Simulator (CATS). Encouraging results were obtained without uncertainty but the costs of robust solutions are prohibitive. Our approach may however be improved, for example, with a prior flight level allocation and the dynamic resolution of remaining conflicts with one of CATS' modules.

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 (Advisory Council for Aeronautics Research in Europe) Strategic Agenda 2 (ACARE, 2004) and the European Single Sky program SESAR (Single European Sky ATM Research). Current ATM optimization strategies, like reducing the size of control sectors or the distance of separation (reduction of vertical separation with RVSM (Reduced Vertical Separation Minima), reduction of horizontal separation with P-RNAV (Precision Area Navigation)), seem to have reached the structural limits of the system, while the automation of Air Traffic Control (ATC) has known few significant improvements over the last decades (Garot & Durand, 2005).

In this context, the European Commission has launched the Episode 3 (Graham & Young, 2006) research project to assess the concepts studied within SESAR definition phase. Among the key concepts identified to meet SESAR performance objectives, the planning of 4D-trajectories could result in increased en-route capacities, while preserving the current level of safety. One of the goals of the Work Package 4 (WP4) of Episode 3 is to estimate how strategic deconfliction

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

schemes could benefit from such regulations over the current Air Traffic Flow Management (ATFM) process.

Currently, the Central Flow Management Unit (CFMU) in Brussels is in charge of optimizing the traffic by, among other strategic or tactical measures, allocating departure slots (i.e. a departure time to be honoured within a $-5/+10$ minutes margin) to the flights involved in overloaded en-route sectors. The purpose of this ground holding scheme 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 Dalichampt *et al.* (1997) and Barnier *et al.* (2001) aimed at improving this slot allocation over the greedy algorithm used at the CFMU. However, one of the limitations of this regulation model 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 Gianazza and Guittet (2007).

Instead of trying to satisfy en-route capacity constraints, we propose to directly solve the potential conflicts occurring between any two intersecting trajectories with departure time adjustments. A single delay would be associated with each flight such that all potential conflicts occurring above a given flight level[1] (FL) would be avoided. This very fine grain model would of course generate much larger constraints sets than the macroscopic (at the sector level) capacitated ones, but would guarantee 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, but the accuracy of Flight Management Systems will be a crucial issue for future ATFM and ATC systems, as advocated by Alliot and Colin de Verdière (2003) and acknowledged by the Airbus-driven 'Technological Enablers' WP6 of Episode 3. Nevertheless, we believe that 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) would then be solved dynamically, either by automated resolution systems as proposed by Granger *et al.* (2001) and Archambault (2004), or 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 the problem can be easily refined 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. A preliminary version of this work with a smaller data set has been presented in Barnier and Allignol (2009).

In the following sections, we first briefly introduce ATC and ATFM in Europe, focusing on ground holding policies and related research projects. Then we describe our model of a conflict-free slot allocation, starting by the details of the constraints generation and search strategy. Next, our first results on instances of the French Traffic are presented, as well as the effect of small takeoff time uncertainties. We end with planned further works to enhance the approach before concluding.

2 Context and related works

2.1 *Air traffic control and air traffic flow management*

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), that is, airspace in which approach control service is provided, two aircraft should remain distant from each other at least by 5 NM (Nautical Mile, 1 NM = 1852 m) horizontally or 1000 ft (1 ft = 0.3048 m) vertically, as illustrated by the safety volume of Figure 1.

[1] A flight level is a standard nominal altitude, expressed in hundreds of feet from the international standard pressure datum of 1013.25 hPa.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

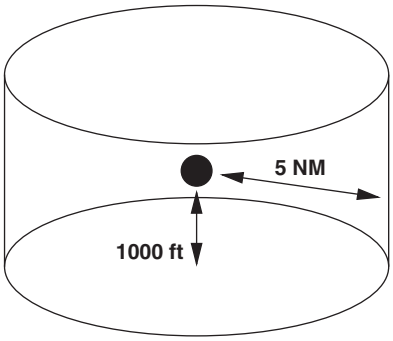


Figure 1 Vertical and horizontal separation: another aircraft cannot be inside the cylinder at the same time.

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

1. *Strategic* (several months before departure time): Airspace Management. Includes the design of routes, sectors and procedures (e.g. reduced separation RVSM since 2002, Area Navigation (RNAV) with fictive beacons, etc.).
2. (*Pre-*)*Tactical* (a few days to a few hours before departure time): ATFM. ATCCs opening schedules define hourly capacities for each open sectors (or groups of sectors). To respect these constraints, the CFMU computes and updates flow regulations (ground holding delays) and reroutings according to the posted flight plans whenever and wherever the resulting local workload would exceed the capacity.
3. *Real-time* (2/15 minutes before conflict): ATC. The controllers are in charge of the surveillance of the traffic, its coordination with adjacent centres, of potential conflict resolution by various simple manœvres (heading, FL, speed) transmitted to the pilots, as well as the dynamic reconfiguration of open sectors[2] to balance the workload.
4. *Emergency* (<2 minutes before conflict): safety nets. Includes ground-based (e.g. Short Term Conflict Alert, Minimum Safety Altitude Warning) and airborne (Traffic alert and Collision Avoidance System (TCAS), Ground Proximity Warning System) systems issuing alarms (as well as simple resolution manœuvres for the TCAS).

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 cannot easily be paused en-route[3] whenever the traffic complexity becomes too high to be safely handled by a controller, one of the simplest way to regulate the ATC workload is to postpone the takeoff of aircraft[4]. This kind of measure is however quite unpopular among airlines, as it can be very costly and may propagate in terms of missed transfers and aircraft rotation (see Cook *et al.*, 2004), 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 defined by local experts (the Flow Management Position, FMP) in each ATCC for the

[2] Note that this activity is usually considered as part of tactical flow management rather than ATC.

[3] They may be put on a racetrack pattern before landing if the destination airport is congested though.

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

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

day of traffic. The ATFM service of the CFMU first identifies the overloaded sectors and responsible flows with the PREDICT (Variant of TACT (Tactical System) used for Pre-Tactical Work) tool, then computes a slot allocation as ground delays for the involved flights with the CASA (Computer-Assisted Slot Allocation) tool described in CFMU (2000).

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 an optimal one. CP technology has been applied with good results to prove and optimize the allocation process with a relaxed model (Dalichampt *et al.*, 1997) or to smooth the resulting load profiles (Barnier *et al.*, 2001) 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, chronically present very different profiles than the predicted ones, as shown in Gianazza and Guittet (2007).

To overcome this issue, recent works such as Flener *et al.* (2007) 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 Barnier (2002) 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 Tran Dac and Baptiste (2003).

2.2.2 Solving the conflicts

One of the key ATM operational concept of SESAR that should be investigated by the Episode 3 project 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, by ground holding only, that is, without changing the requested trajectory, while minimizing the worst allocated delay to maintain equity among airlines.

Optimality proofs for the overall sum of the delays can be exponentially harder than our max criterion, and therefore out of reach for such large problems. However, our search strategy will focus on maintaining the overall amount of delay as low as possible, while the use of CP technology will provide proved maximal delay bounds that other optimization techniques (e.g. local search or meta-heuristics) cannot produce.

This conflict-free model will of course yield much larger problem instances as all of the conflicting trajectories intervals above a given FL will be taken into account as constraints. The resulting problem is intrinsically disjunctive 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).

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 (Granger, 2002) and small speed adjustments as proposed by the ERASMUS (E-Route Air Traffic Soft Management Ultimate System) project (Archambault, 2004). 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 data are provided by the CATS[5] simulator (Alliot *et al.*, 1997), which takes as input all of the filed flight plans concerning the French airspace for a given day of traffic and the relevant airspace configuration (sectors, waypoints, etc.), and can output the corresponding 4D-trajectories.

[5] The Complete Air Traffic Simulator developed at DSNA/DTI.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

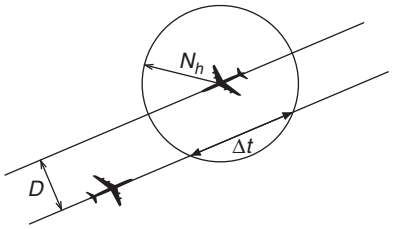


Figure 2 Influence of time step in conflict detection.

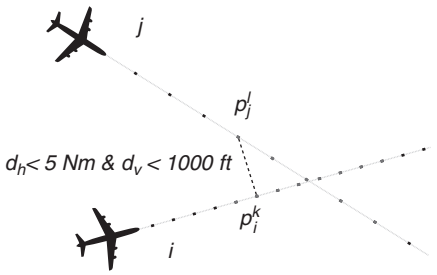


Figure 3 Detection of conflicting points.

To detect all 'significant' conflicts, CATS discretizes trajectories with a $\Delta t = 15$ seconds time step. As depicted in Figure 2, the worst case for conflict detection occurs when two aircraft are following parallel trajectories separated by a distance $D \leqslant N_h$ ($N_h = 5$ NM being the horizontal separation norm) and heading in opposite directions at maximum possible speed $V_{max} = 500$ kt. Given a time step Δt for trajectories discretization, undetected conflicts may occur when

$$\Delta t > \frac{N_h}{V_{max}} \cos\left(\arcsin\left(\frac{D}{N_h}\right)\right)$$

Therefore, by choosing $\Delta t = 15$ s, false negative detections would only occur for

$$D > N_h \sin\left(\arccos\left(\frac{\Delta t V_{max}}{N_h}\right)\right) > 4.5 \text{ NM}$$

that is, for $4.5 \text{ NM} < D \leqslant 5 \text{ NM}$, which is commonly accepted for fast-time simulations as such routing schemes are not allowed by standard ATM practice.

Trajectories are then probed pairwise[6] for potential conflicts, ignoring those that could only occur for greater delays than the given maximal one. The separation norm is thus tested for each pair of points of the two probed trajectories (up to $p = 1300$ points per trajectory for up to $n = 9500$ flights in $\mathcal{O}(n^2p^2)$, as observed in the largest instances simulated by CATS) as illustrated in Figure 3 in the horizontal plane. Note that trajectory enclosing bounding boxes or sweep line techniques (de Berg *et al.*, 1998) could be used to lower the detection complexity, but it is here considered as a static data production phase and its efficiency is not a primary concern of the present study.

Though the maximal allowed delay can be seen as a parameter of the search algorithm only, it also affects 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 24 hours delay would be allowed, the conflict detection could be done regardless of time, as any two space-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

[6] Note that the conflict detection for two given flights is symmetrical, so that only ordered pairs are considered.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

maximal delay, which will capture later potential conflicts on the trajectories pairs and increase the size of the instance.

Operationally, flights originating outside the Eurocontrol countries cannot be delayed, so their delay variable will be fixed to 0 in our constraint model, reducing the number of variables but tightening the constraints as well and offering less opportunities 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 will not be addressed in this study.

3.2 Constraint programming model

Our 4D-trajectory deconfliction scheme by ground holding is modelled in the following sections using the versatile formalism of finite domain Constraint Satisfaction Problem, which is the underlying modelling component of the Constraint Programming (CP) technology (Van Hentenryck *et al.*, 1992) that we have used to solve the resulting combinatorial optimization problem.

3.2.1 Conflicts constraints

To compute the constraints of our model, the trajectories (up to 9500 flights for one day of traffic in the French airspace, see Section 4) 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 [0, max_delay] that represent the delay associated with each of the n flights (between 6000 and 8000 after processing for our experiments). The value of max_delay is typically chosen as 90 minutes and increased as needed when no solution is found—up to 300 minutes in our experiments, see Section 4. As explained in Section 3.1, the size of the instance grows with max_delay.

We will describe our model using the following auxiliary variables (defined for each unique pair of delay variables, i.e. exactly $\frac{1}{2}n(n-1)$):

- $\theta_i^k = t_i^k + \delta_i$ 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 vertical 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, beginning at point $p_j^{l_k}$ and ending at some point $p_j^{l_k + r}$:

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

for some r, and we impose on the difference of the delays of flight i and j that

$$d_{ij} \notin \{t_i^k - t_j^l, \forall l \in [l_k, l_k + r]\}$$

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

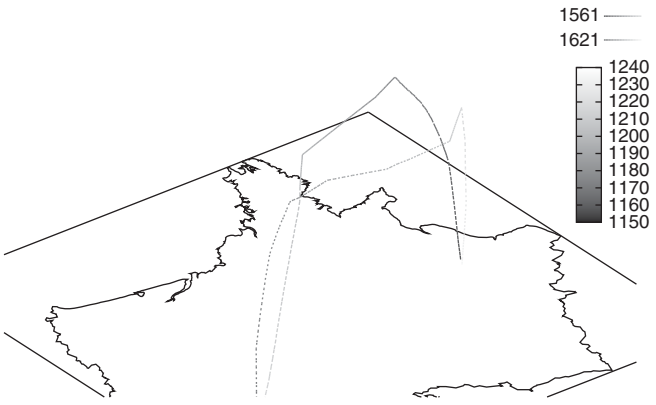


Figure 4 Three potential conflicts between two flights: one near Paris airport at low altitude and two other en-route at the cruising altitude of the lower flight. The grey scale corresponds to time (in minutes) along the trajectory, the lighter the later.

which can be rewritten as

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

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

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

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 maximal allowable delay max_delay, therefore the domain of $d_{ij} = \delta_j - \delta_i$ is the interval [−max_delay, max_delay]. We simply discard the conflict whenever $\overline{ub} < -\text{max_delay}$ or $\underline{lb} > \text{max_delay}$.

A pair of flights may conflict several disjoint times over their entire trajectories (as illustrated in Figure 4), 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 whole trajectories, we then have

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

or, rewritten as a disjunctive constraint over the decision variables

$$(-\text{max_delay} \leqslant \delta_j - \delta_i < \underline{lb}^1) \vee$$
$$(\overline{ub}^1 < \delta_j - \delta_i < \underline{lb}_2) \vee \cdots \vee$$
$$(\overline{ub}^{\sigma-1} < \delta_j - \delta_i < \underline{lb}_\sigma) \vee$$
$$(\overline{ub}^\sigma < \delta_j - \delta_i \leqslant \text{max_delay})$$

provided that $\underline{lb}^1 > -\text{max_delay}$ and $\overline{ub}^\sigma < \text{max_delay}$, otherwise the first or last part of the disjunction is discarded.

As aforementioned, the cost is simply defined as the maximal allocated delay, to ensure equity among the various postponed flights:

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

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

However, the overall sum of the delays is of utmost importance as well for the quality of a solution and we will take it into account within the search strategy to provide realistic max-optimal solutions during the resolution of the problem.

3.2.2 *Further instance processing*

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 TMA control services in these zones.

After the computation of the conflict constraints, the whole instance is scaled down to a more reasonable time step than the 15 seconds used during conflict detection. We ensure that the original forbidden intervals are strictly included in the scaled ones, with appropriate interval arithmetic operations which can possibly merge conflicting disjoint intervals. A 1-minute time step was chosen for the resolution: a smaller time step would not have been realistic for a departure time, whereas a greater one would have reduced the set of feasible solutions. Besides, a 1-minute time step seems compliant with the SESAR objectives of $\pm$3-minute precision for takeoff time.

Moreover, the FL of the detected conflicts can be filtered, for example to only take into account conflicts occurring within the upper airspace (from FL200 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.

It is also possible 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.

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

3.2.3 *Conflict extension*

To improve the robustness of our solutions towards uncertainty on the departure times of the flights, we add an extra parameter *ext* that extends conflicting intervals by a fixed amount of time. Such an extension of *ext* minutes stretching the start and the end of a conflict will be able to manage uncertainties of $\pm\frac{ext}{2}$ minutes on the departure slot (see Section 4.4), at the price of an increase in the cost of the solutions.

We expect that this extension scheme may as well be able to diminish the effects of other sources of uncertainty (e.g. vertical and ground speed) regarding the number of remaining conflicts during simulation.

3.3 *Search strategy*

The constraints of the problem are reminiscent of the disjunctive mutual exclusion constraints often used to model scheduling problems (Baptiste *et al.*, 2001). 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 even be seen as the ordered tasks of a given job, as in the Job-shop Scheduling Problem (JSP).

However, the comparison does not hold much further. First, the time intervals between any two conflict-tasks of the same trajectory are fixed, as only one delay variable is associated with each flight (unlike the JSP where all tasks are only related with precedence constraints). Second, to consider a potential conflict in three dimensions 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 to be separated by 5 NM only. 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 in Figure 6 and cannot be easily related with any known standard scheduling problem.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

Nevertheless, the branching scheme of our search strategy to solve this essentially disjunctive problem is inspired by standard scheduling techniques. For instance, trying to start the search by directly labelling the delay variables δ_i may impede the search because of thrashing (i.e. repeatedly fail over the same constraint), as the constraints are expressed over the differences d_{ij}. In this case, a more efficient filtering can be obtained by feeding the propagation of the arithmetic constraints with new domain bounds for the d_{ij} auxiliary variables.

In this respect, our search strategy first try to order pairs of conflicting flights by adding 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, that is, the smallest ratio between the domain size and the difference of the domain bounds, is chosen first for branching.

To compensate for the cost being defined as the maximal delay only, disregarding the total amount of time, we choose to branch first within the d_{ij} interval corresponding to the minimum potential increase for its delay variables δ_i and δ_j. Such an interval would be the closest to 0, while if d_{ij} were far from 0, then at least one delay would be large. Whenever the search backtracks over such a decision, this interval is discarded and we branch on the next one recursively.

When all conflicts are ordered and there is no more hole in the domain of the d_{ij}, we start labelling the decision variables δ_i with a standard dom/deg selection heuristic: the variable with the smallest domain, and the highest number of constraints in case of tie, is dynamically selected. Then, the values closest to 0 are probed first to attempt to keep the total amount of delay as low as possible.

After the first solution is found, the branch and bound algorithm proceeds by dichotomy on the cost domain to find the optimal solution with respect to minimization of the maximal allocated delay, while keeping low the overall amount of delay thanks to the search strategy.

4 Results

We have implemented this CP model with the FaCiLe (Functional Constraint Library) (Barnier & Brisset, 2001) and obtained the following results on various day of traffic in 2007. Our data set consists in full days of traffic within the French airspace for all ATCCs, with up to 9500 flights and 600 000 intersecting pairs of trajectories taken into account for the largest instances we could optimally solve. About 10% of the flights are non-European flights, their delays will therefore be fixed to 0 as aforementioned. Among these flights, the remaining unsolvable conflicts amounts to 5–10% of the number of undelayable flights.

4.1 Data processing

The resulting constraint graphs typically exhibit only one single large connected component of maximal degree that can be >300, that is, a single flight may conflict with >300 other flights. Large cliques can also be found, as large as involving >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, respectively, very small and very large forbidden intervals.

We mainly tune the size of our instances by choosing the minimal FL for which the conflicts are taken into account, aiming at the upper airspace (above FL200) where most of the cruising traffic occurs. However, we were able to optimally solve one instance with >6600 flights and 630 000 conflicting pairs, taking all conflicts into account for the whole day, regardless of the FL. Figure 5 shows the number of flights of our instances as a function 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—note that plots are labelled with their date and the type of routes (standard or direct), for example, '070123s' for the day of traffic on the 23rd of January 2007 with standard routes.

The number of conflicting pairs is not quite quadratic with the number of flights, as mentioned in Section 3.2 and shown in Figure 6, but is quite huge anyhow, reaching 630 000 for our largest instance.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

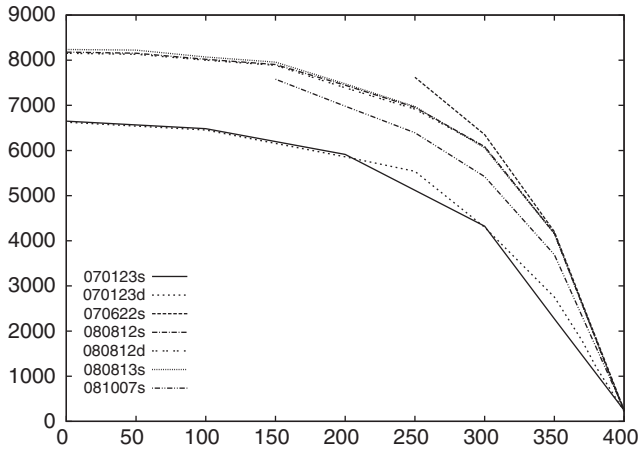


Figure 5 Number of flights w.r.t. minimal flight level.

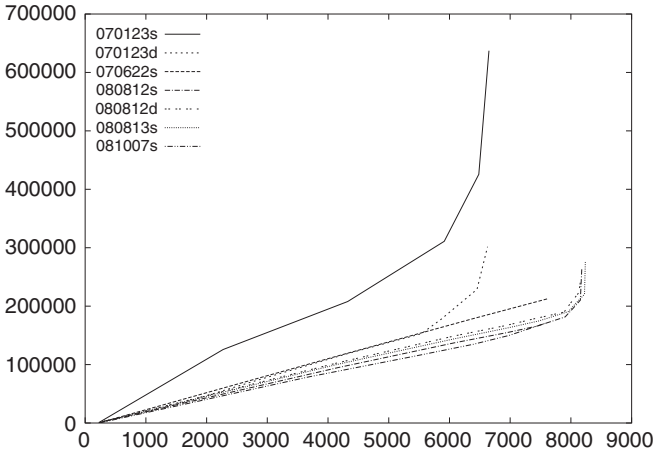


Figure 6 Number of conflicts w.r.t. number of flights.

4.2 Numerical results

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. However, the search strategy exposed in Section 3.3 was not always the best, and for some instances, a chronological labelling breaking ties with a standard dom/deg ordering could obtain solutions faster or with a lower overall amount of delay. The following graphs only exhibits the results of the best search strategy for conciseness reasons.

As shown in Figure 7, small instances are solved in a few seconds whereas the biggest ones could take almost 1 minute, growing only quadratically with the number of flights (Figure 7(a)). We plan to address larger instances, hopefully European ones, on a computer with more memory.

However, the cost of this conflict-free slot allocation can be quite high for the busiest days (our worst case above FL350 is 182 minutes for the most delayed flight with a max_delay of 300 minutes to compute the instance), but may be more reasonable (around 60–90 minutes) for less crowded days. Figure 8 shows that the cost grows steadily for small instances (i.e. at high minimal FL), but jumps as soon as we add the main flows of traffic around FL350 for bigger instances. The optimal cost then seems to be stable for larger instances, triggered only by the flights added around FL350 (see Figure 8(b)).

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

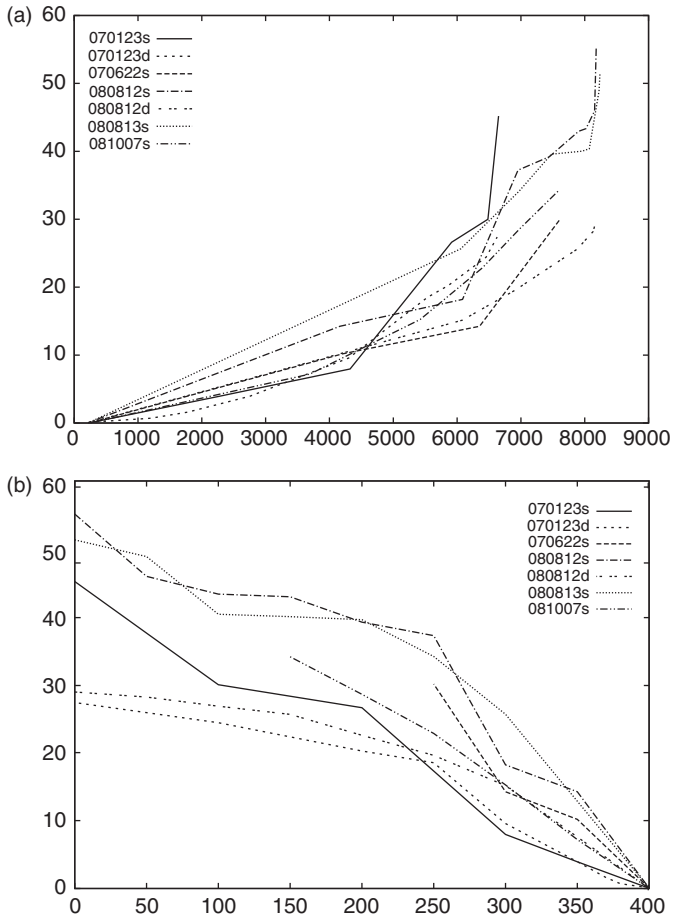


Figure 7 Computation times: (a) Computation time (optimality proof) in seconds w.r.t. number of flights, (b) Computation time (optimality proof) in seconds w.r.t. minimal flight level.

The corresponding overall delay sum (Figure 9) and percentage of delayed flights (Figure 10) exhibit of course a more steady behaviour, dramatically increasing with the largest instances only.

Central Office for Delay Analysis (2009) provides some statistics on observed ATFM delays. In this report, a flight is considered delayed if its observed delay is >5 minutes. For 2009, the average delay per delayed flight (with respect to the previous definition) was 20 minutes. Even if we do not optimize the mean delay, we can see in Figure 11 that our figures remain comparable to the actual CFMU delays: in most cases, the mean delay per delayed aircraft is strictly under 20 minutes. The worst cases occur for extremely small instances with very few flights (or high minimal FL) where the number of delayed flights is so low that their statistical distribution is meaningless.

4.3 Standard and direct routes

For one of the days of traffic (plots labelled '070123s' and '070123d'), we have also tested our model on *direct routes*. We say that aircraft follow a direct route when they fly in straight line at the requested FL from origin to destination, as opposed to *standard routes* where their path from origin to destination is a succession of segments between waypoints. 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 lower tightness (see Figure 6), and it is interesting to observe that overall delay sums are smaller than the ones associated with standard

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

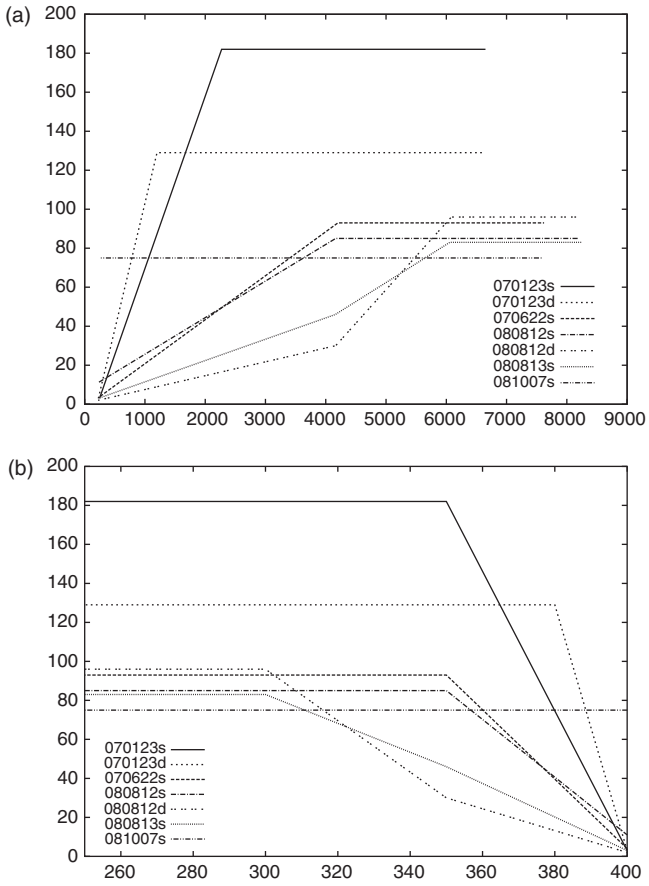


Figure 8 Optimal costs: (a) Optimal cost w.r.t. the number of flights, (b) Optimal cost w.r.t. the minimal flight level.

routes in Figure 9(b). Flights following standard routes tend to be on closer trajectories, suitable for the efficiency of current ATC procedures, but not using airspace to its full capacity. The max cost can be greater with direct routes though, depending on the day of traffic and on the minimal FL considered, as observed in Figure 8 for 08/12/2008 (labelled '080812s' and '080812d').

4.4 Robustness towards uncertainty

Our model and algorithm were validated by taking the generated solutions as input delays for the CATS simulator. We observed that the few remaining conflicts occur below the chosen minimal FL only, or involve flights originating outside the Eurocontrol area, which cannot be delayed in our model (see Section 3.1).

However, this first validation does not take into account any uncertainty with respect to departure time or aircraft navigation (i.e. aircraft not following its 4D-trajectory precisely enough). To assess the robustness of our solutions, we have added an uncertainty parameter *err* on departure times within CATS. Takeoffs are randomly shifted by a bounded amount of time uniformly[7] chosen in the interval $[-\frac{err}{2}, +\frac{err}{2}]$.

New validation tests were carried out with various values of *err* to assess the effect of the conflict extension parameter *ext* (defined in Section 3.2.3) to compensate for the uncertainties. As expected, Figure 12 shows that for the region where $ext \geqslant err$, no conflict remains: all points

[7] A better approach would involve a statistical analysis to approximate the probability distribution of the discrepancy between scheduled and actual takeoff times.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

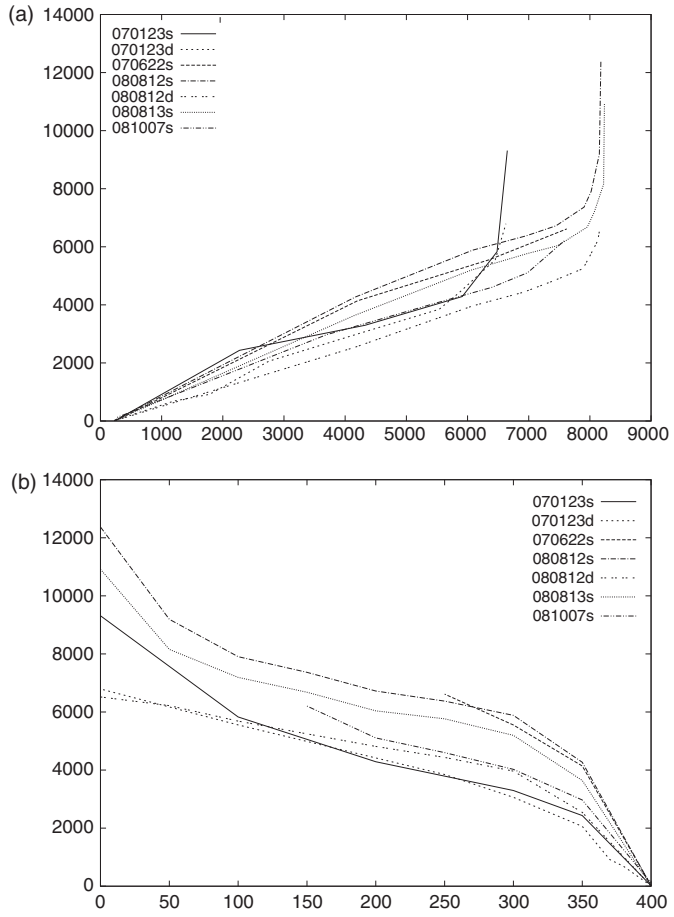


Figure 9 Total amount of delay: (a) Total amount of delay in minutes w.r.t. the number of flights, (b) Total amount of delay in minutes w.r.t. the minimal flight level.

below the $ext = err$ dashed line on the xy-plane exhibit a conflict percentage equal to zero. Above this line, the ratio of remaining conflicts increases with err for a given ext and when ext diminishes for a given err, reaching 75% for the highest point ($err = 6$ and $ext = 0$).

However, increasing the ext parameter leads to an increase in the total amount of delay as illustrated in Figure 13. The added delays can be far too costly for higher values of ext, especially for large instances. The target figures for efficiency within SESAR are:

- at least 98% of flights departing on time (on-time departure being defined as actual departure $<$3 minutes before or after scheduled departure),
- the average departure delay of delayed flights must not exceed 10 minutes.

For the traffic sample used in Figure 13, the delays are higher than 15 minutes per delayed flight, with $>$60% delayed flights, far from the above objectives. Thus we cannot hope to solve all conflicts with this technique alone within SESAR time slot objectives ($\pm$3 minutes precision on takeoff time), let alone CFMU margins ($-5/+10$ minutes slot around takeoff time), as operational delay cost would be prohibitive. As presented in the next section, other regulation or dynamic resolution techniques may be used to overcome this issue.

5 Further work

These first results are encouraging but we have only addressed so far the resolution of conflicts within the French airspace. However, in a unified European ATC context, all conflicting traffic

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

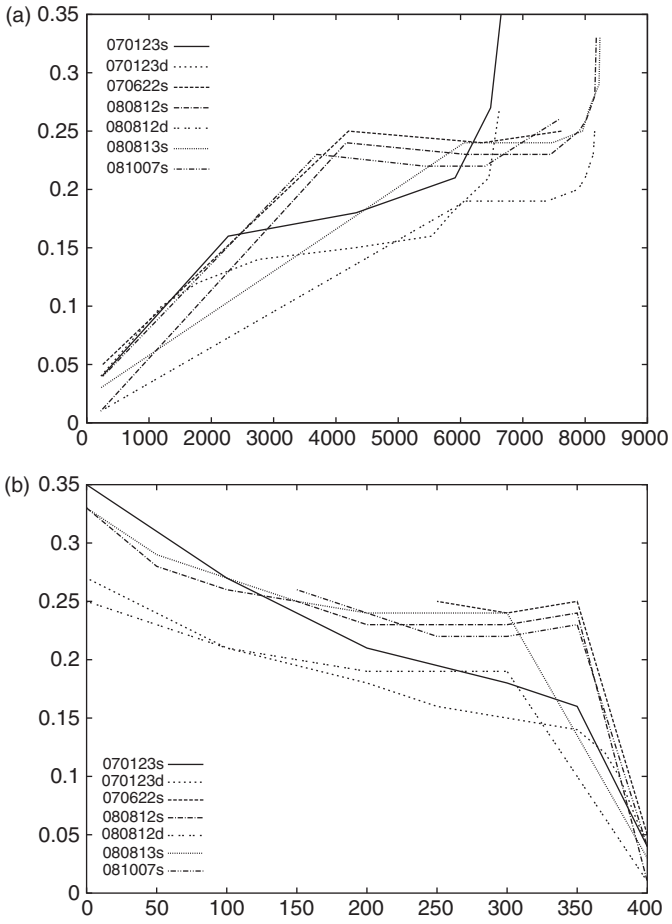


Figure 10 Ratio of delayed flights: (a) Ratio of delayed flights w.r.t. the number of flights, (b) Ratio of delayed flights w.r.t. the minimal flight level.

throughout the Eurocontrol countries should be taken into account. Such instances would comprise up to 30 000 flights per day. We plan to experiment with various refinements of our algorithm to address such large-scale problems.

5.1 Prior flight level allocation

To be able to address such large instances, while maintaining reasonable maximal delay figures, we plan to combine our ground holding algorithm with a prior FL allocation stage, possibly using CP technology as well, as described in Barnier and Brisset (2002).

This first step, which tries to minimize the number of 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 by a substantial amount before time slot allocation. The optimal cost should then remain within much better bounds than with the raw traffic.

5.2 Sliding time windows

For large European instances, we also 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 and the cumulative effects of uncertainties. Similar approaches are used for dynamic conflict resolution in the CATS solver as described in Granger *et al.* (2001).

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

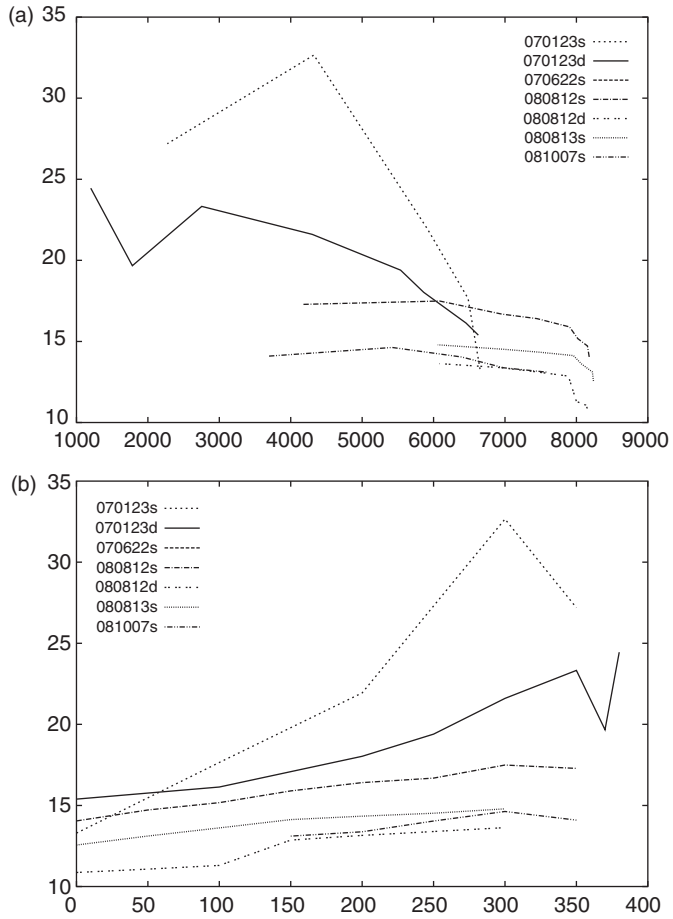


Figure 11 Mean delay per delayed flight. In this figure, for comparison with CFMU delay, we consider that a flight is delayed if its delay is greater or equal to 5 minutes. (a) Mean delay per delayed flight w.r.t. the number of flights, (b) Mean delay per delayed flight w.r.t. the minimal flight level.

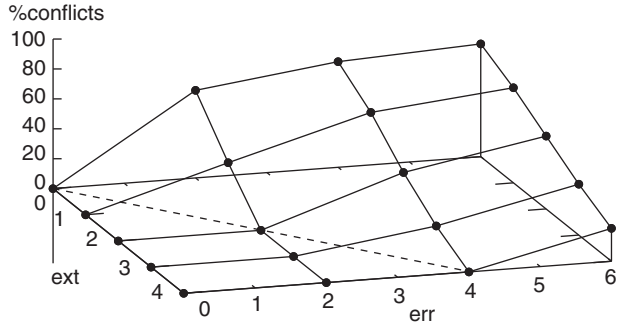


Figure 12 Percentage of remaining conflicts w.r.t. conflict extension and departure time error in minutes.

5.3 Resolution with complete air traffic simulator modules

Various real-time resolution algorithms have already been implemented within the CATS simulator (Granger *et al.*, 2001; Granger, 2002; Archambault, 2004) with very good results. We would like to assess our regulation schemes further by observing the workload of these algorithms when provided with our solutions and various kind of uncertainties, and whether the automatic resolution can cope with our delays or if the resulting manoeuvres will warp our planning entirely.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

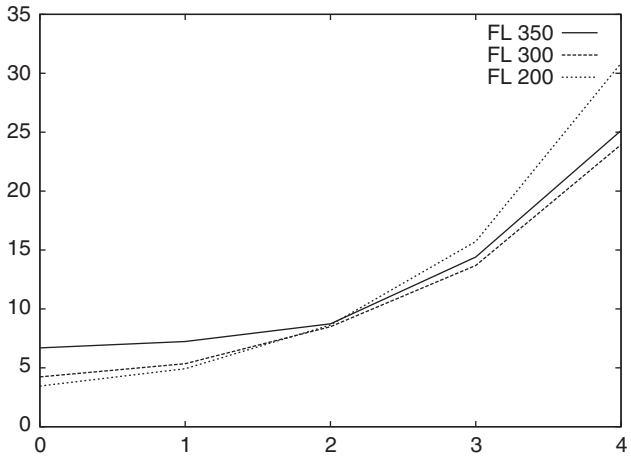


Figure 13 Mean delay per delayed flight w.r.t. conflict extension in minutes. The indicated flight levels (FLs) are the minimal FLs used for conflict detection.

6 Conclusion

We have presented a new ground holding approach to solve all potential conflicts occurring above a given FL 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 adjustments of departure times to keep them separated, with the hypothesis that aircraft could 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, overall delay sum and ratio of delayed flights 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. Nevertheless, our solutions were validated with the CATS simulator, checking that no conflict under the given FL for a delayable aircraft remains.

We have also presented a first step towards taking uncertainties into account by extending the forbidden intervals of conflicting flights. However, an extension as small as 4 minutes, which is able to cope only with a ± 2-minute uncertainty on the departure time generates tremendous amounts of delays, far above SESAR performance objectives.

We plan to overcome these issues and further assess the possible outcomes of 4D-trajectory planning in the context of Episode 3 WP4 and address larger (European) instances with various techniques like combining our delay algorithm with a prior FL allocation, repeatedly solving the problem on a sliding time windows or solving the remaining conflicts with a CATS resolution module.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable suggestions and constructive comments that helped to improve this paper.

References

ACARE 2004. *Strategic Research Agenda 2 (SRA 2)*. Technical report, Advisory Council for Aeronautics Research in Europe.

Alliot, J.-M., Bosc, J.-F., Durand, N. & Maugis, L. 1997. CATS: a complete air traffic simulator. In *Proceedings of the 16th Digital Avionics Systems Conference DASC'97*. Irvine (CA), USA.

Alliot, J.-M. & Colin de Verdière, D. 2003. 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*. Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press

Archambault, N. 2004. Speed uncertainty and speed regulation in conflict detection and resolution in air traffic control. In *Proceedings of the International Conference on Research in Air Transportation ICRAT'2004*. Zilina, Slovakia.

Baptiste, P., Le Pape, C. & Nuijten, W. 2001. *Constraint-Based Scheduling, Applying Constraint Programming to Scheduling Problems*. Kluwer's International Series in Operations Research & Management Science, Springer.

Barnier, N. 2002. *Application de la programmation par contraintes à des problèmes de gestion du trafic aérien*. PhD thesis, Institut National Polytechnique de Toulouse. Toulouse, France.

Barnier, N. & Allignol, C. 2009. 4D-trajectory deconfliction through departure time adjustment. In *Proceedings of the 8th USA/Europe R&D Seminar on Air Traffic Management ATM'2009. Napa, CA, USA*.

Barnier, N. & Brisset, P. 2001. FaCiLe: a Functional Constraint Library. In *Proceedings of the Colloquium on Implementation of Constraint and LOgic Programming Systems CICLOPS'01 (Workshop of CP'01)*. Paphos, Cyprus.

Barnier, N. & Brisset, P. 2002. Graph coloring for air traffic flow management. In *Proceedings of the CPAIOR'02: 4th International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems*. Le Croisic, France, 133–147.

Barnier, N., Brisset, P. & Rivière, T. 2001. Slot allocation with constraint programming: models and results. In *Proceedings of the 4th USA/Europe R&D Seminar on Air Traffic Management ATM'2001*. Santa Fe (NM), USA.

Central Office for Delay Analysis 2009. *CODA Digest – Delays to Air Transport in Europe*. Technical report, Eurocontrol.

Central Flow Management Unit (CFMU) 2000. *Basic CFMU Handbook – General & CFMU Systems*, 6.0 edn. Eurocontrol (CFMU), Brussels.

Cook, A. J., Tanner, G. & Anderson, S. 2004. *Evaluating the true cost to airlines of one minute of airborne or ground delay: final report*. Technical report, Eurocontrol.

Dalichampt, M., Petit, E., Junker, U. & Lebreton, J. 1997. *Innovative slot allocation (ISA)*. Technical report, Eurocontrol.

de Berg, M., van Kreveld, M., Overmars, M. & Schwarzkopf, O. 1998. *Computational Geometry – Algorithms and Applications*. Springer.

Flener, P., Pearson, J., Ågren, M., Garcia Avello, C., Çelitkin, M. & Dissing, S. 2007. Air-traffic complexity resolution in multi-sector planning. *Journal of Air Transport Management* **13**(6), 323–328.

Garot, J.-M. & Durand, N. 2005. Failures in the automation of air traffic control. In *Proceedings of the Colloque de l'Association Aéronautique et Astronautique de France 3AF*. Toulouse, France.

Gianazza, D. & Guittet, K. 2007. Selection and evaluation of air traffic complexity metrics. In *Proceedings of the 25th Digital Avionics Systems Conference DASC'07*. Portland (OR), USA.

Graham, R. & Young, D. 2006. *Preparing an Initial Assessment of the SESAR Concept of Operations "EP3: Single European Sky Implementation Support through Validation"*. Technical report, Eurocontrol Experimental Centre, Brétigny, France.

Granger, G. 2002. *Détection et résolution de conflits aériens : modélisations et analyse*. PhD thesis, École Polytechnique. Palaiseau, France.

Granger, G., Durand, N. & Alliot, J.-M. 2001. Optimal resolution of en route conflicts. In *Proceedings of the 4th USA/Europe R&D Seminar on Air Traffic Management ATM'01*. Santa Fe (NM), USA.

Tran Dac, H. & Baptiste, P. 2003. Airspace sectorization by constraint programming. In *Proceedings of the Rencontres en Informatique Vietnam-France RIVF'03*. Hanoi, Vietnam.

Van Hentenryck, P., Simonis, H. & Dincbas, M. 1992. Constraint satisfaction using constraint logic programming. *Artificial Intelligence* **58**(1–3), 113–159.

https://doi.org/10.1017/S0269888912000227 Published online by Cambridge University Press