Combining flight level allocation with ground holding to optimize 4D-deconfliction

Nicolas Barnier, Cyril Allignol

Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)

Combining Flight Level Allocation with Ground Holding to Optimize 4D-Deconfliction

Nicolas BARNIER
ENAC
7, Av. Édouard Belin - BP4005
F31055 Toulouse Cedex 4
barnier@recherche.enac.fr

Cyril ALLIGNOL
DSNA/DTI
7, Av. Édouard Belin - BP4005
F31055 Toulouse Cedex 4
allignol@tls.cena.fr

***Abstract*—As acknowledged by the SESAR program, current 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.**

Building on a preliminary ground holding algorithm aimed at directly solving all conflicts (instead of satisfying sector capacity constraints), a prior flight level allocation program is used to reduce the complexity of the traffic input, dramatically improving the quality of the solutions.

We present Constraint Programming (CP) models of these large scale combinatorial optimization problems and the encouraging results obtained with the FaCiLe constraint library. However, our approach does not yet address uncertainties and we plan to overcome this issue by improving the robustness of our conflict model and iteratively solving the problem over a sliding time window.

Keywords: Flight Level Allocation, Ground Holding, Deconfliction, Constraint Programming

I. 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 [1] and the European Single Sky program SESAR. Current ATM optimization strategies, like reducing the size of control sectors or the distance of separation (RVSM, P-RNAV), 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 [2].

In this context, the European Commission has launched the Episode 3 [3] 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 would allow to increase en-route capacity, while preserving the current level of safety. One of the goals of the WP4 of Episode 3 is to estimate how such regulations could benefit strategic deconfliction schemes 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, delaying departure slots for 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 [4], [5] 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 [6].

Therefore, in [7] we proposed to directly solve by ground holding the potential conflicts occurring between any two intersecting trajectories instead of trying to satisfy en-route capacity constraints. 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 generates much larger constraint sets than the macroscopic (at the sector level) capacitated ones, but guarantees conflict-free trajectories all along the flight path. This scheme proved to generate acceptable delays (w.r.t. CFMU figures) if no uncertainty on the takeoff times is taken into account, but they dramatically increase otherwise.

Regulation with ground holding benefits from only one degree of liberty (time), but 4D-trajectories could be deconflicted along the spatial dimensions as well. If we leave to airlines the choice of the flight paths of their flights, there still is the cruise altitude, or flight level (FL), on which allocation schemes could attempt to deconflict the traffic. Previous works like [8], [9] present approaches to optimize the flight level allocation within a direct route network to avoid conflicting flows, but without trying to optimize on delays as well. Trying to solve the deconfliction problem simultaneously on both dimensions, with such large instances as the French or European traffic, is still out of reach for current combinatorial optimization technology, but it is possible to combine the two kind of deconfliction schemes successively to obtain better solutions than one of the two techniques taken alone, even if the resulting regulations are suboptimal w.r.t. the huge 2-dimensional problem.

So to further optimize our solution or be able to handle takeoff uncertainties, we propose to combine our ground holding scheme with a prior flight level allocation, which aims at vertically separating intersecting flows of aircraft and therefore reduces the number of temporal constraints of the slot allocation. However, the Requested Flight Level (RFL) of a flight plan usually corresponds to an optimal cruise altitude (w.r.t. fuel consumption / CO_2 emission) and thus the distance between the allocated flight level and the RFL should be kept as low as possible, should such a scheme be accepted by airlines. Of course, this first FL allocation step, which approximates trajectories in the horizontal plane without taking the climbing or descent phases into account, only prevent conflicts occurring between levelled flows in their cruise phase. So conflicts involving at least one vertically evolving trajectory, or catch up flights belonging to the same flow, will still have to be resolved by the ground holding algorithm.

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 deviation from RFLs to vertically separate the flows and the maximal delay to solve the conflicts, which can be used to draw conclusions on the feasibility of this kind of regulation. 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 introduce ATC and ATFM in Europe, focusing on ground holding and flight level allocation policies and related research projects. Then we describe our models of an intersection-free flight level allocation for flows of aircraft and of a conflict-free delay allocation. Next, our results on instances of the French Traffic are presented. We end with planned further works to enhance the approach before concluding.

II. Context and Related Works

A. 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.

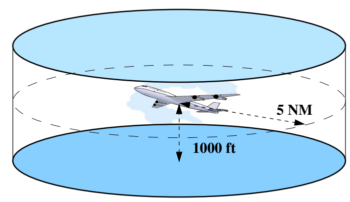

Fig. 1. Vertical and horizontal separation. Another aircraft cannot be inside the cylinder at the same time.

The overall system currently implemented in Europe to achieve this goal can be conceptually divided in several layers or filters 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) *(Pre-)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.

B. Ground Holding

As aircraft obviously cannot be paused while airborne whenever the traffic complexity becomes too 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 correspondences and aircraft rotation (see [10]), so the delays should be minimized as much as possible.

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. [11]).

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 [4] or to smooth the resulting load profiles [5] 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, do 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 [12] 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 [13] 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 [14].

2) Solving the Conflicts: One of the key ATM operational concept of the SESAR program that Episode 3 should validate 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, 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.

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 [15] and small speed adjustments as proposed by the ERASMUS project [16]. 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.

C. Flight Level Allocation

Even if CFMU experts can balance traffic load by separating flights of the same flows over alternative routes, no real optimization of the flight level allocation is currently performed at the strategic/pre-tactical level in Europe, except for a few static rules attributing the parity of FL according to the heading of flights to prevent face-to-face conflicts. At the real time level, Air Traffic controllers can issue temporary FL changes to separate aircraft, and this kind of vertical manœuvres has been integrated in the ERCOS conflict solver [17].

Our work is more similar to the approaches presented in [8], [9] where graph coloring techniques are applied to optimize the FL allocation of French and European networks of direct routes, or in [18] where a Genetic Algorithm and an $A^\star$ algorithm are used to optimize

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

traffic flows with possible changes of FL along the route (as well as lateral deviations for direct routes).

However, contrary to these studies, our model does not fit well within standard graph coloring problems as the choice of FL is very much restricted for operational reasons and each flow will only have a couple of possible FL available above or below its RFL. Moreover, the FL allocation phase should handle over-constrained instances as well to obtain the best possible solution to feed the ground holding phase, even if some conflicts remain, so classic coloring techniques like the use of cliques as lower bounds or "all-different" constraints (cf. [9]) cannot be used to speed up the search. Furthermore, the cost of solutions is measured w.r.t. the number of still conflicting flows and the sum of distances to RFLs (weighted by the number of flights in the flow), which are uncommon criteria for graph coloring problems.

III. Deconfliction with Ground Holding

The ground holding CP model uses as input a set of temporal conflict constraints computed for each pair of flights that intersect in the three spatial dimensions. This conflict detection will be used as well as input of the FL allocation phase, but without taking the vertical dimension into account during intersection computation.

The following section describes the processing of flight plans to compute the conflict constraints and the modelling of deconfliction by ground holding as a constraint program.

A. Conflict Detection

Our data are provided by the CATS[2] simulator [19], which takes as input 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 15 s 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.

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 $n = 8600$ flights in $\mathcal{O}(n^2p^2)$) as illustrated on figure 2 in the horizontal plane.

To reduce the complexity of this detection phase, trajectories are encapsulated into bounding boxes: each trajectory is split into segments (a segment being here defined as a constant heading portion of the trajectory); then each of these segments is encapsulated into a bounding box, such that every point of the segment is farther than half the separation norm from each side of the box, as illustrated in figure 3 in the horizontal plane.

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

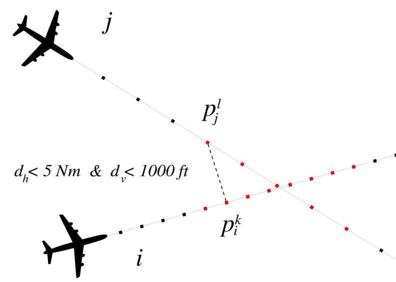

Fig. 2. Conflicting points detection.

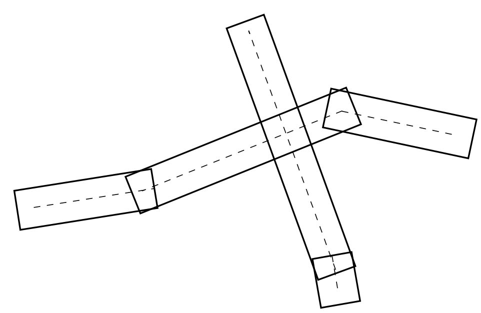

Fig. 3. Two intersecting trajectories and their associated bounding boxes. Conflicting point detection is only performed for points in the intersecting boxes.

Consider two flights i and j with trajectories encapsulated in bounding boxes $(b_i^1, ..., b_i^n)$ and $(b_j^1, ..., b_j^m)$ respectively. If there is an intersection between boxes b_i^k and b_j^l, then the pairwise tests for conflicting points is only performed for points contained in b_i^k and b_j^l, thus saving a lot of useless tests for the rest of the trajectories. This filtering proved to reduce computing time for conflict detection dramatically.

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 techniques beyond the scope of this study.

B. CP Model

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 of decision variables $\{\delta_i, \forall i \in [1, n]\}$ of finite domain $[0, \delta_{\text{max}}]$ that represent the delay associated with each of the n flights, and we will describe our model using the following auxiliary 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;
- $d_{ij} = \delta_j - \delta_i$.

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

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:

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

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 $d_{ij} \notin [lb^{k+1}, ub^{k+1}]$ 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 maximal allowable delay $\delta_i \in [0, \delta_{\text{max}}]$, therefore the domain of $d_{ij} = \delta_j - \delta_i$ is the interval $[-\delta_{\text{max}}, \delta_{\text{max}}]$. We simply discard the conflict whenever $\overline{ub} < -\delta_{\text{max}}$ or $\underline{lb} > \delta_{\text{max}}$.

A pair of flights may conflict several disjoint times over their entire trajectories, 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:

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

or, rewritten as a disjunctive constraint over the decision variables $(-\delta_{\text{max}} \leq d_{ij} < \underline{lb}^1) \vee (\overline{ub}^1 < d_{ij} < \underline{lb}_2) \vee \cdots \vee (\overline{ub}^{\sigma-1} < d_{ij} < \underline{lb}_\sigma) \vee (\overline{ub}^\sigma < d_{ij} \leq \delta_{\text{max}})$, provided that $\underline{lb}^1 > -\delta_{\text{max}}$ and $\overline{ub}^\sigma < \delta_{\text{max}}$, otherwise the first or last part of the disjunction is removed. We note:

$$\mathcal{C}_{ij} = \left[\underline{lb}^1, \overline{ub}^1\right] \cup \cdots \cup \left[\underline{lb}^\sigma, \overline{ub}^\sigma\right] \tag{2}$$

the union of forbidden intervals that represents the conflict between flights i and j.

The cost of a solution is then defined as:

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

IV. Deconfliction with Flight Level Allocation

The flight level allocation is aimed at vertically separating intersecting flows of aircraft, in order to reduce the complexity of the traffic, prior to the takeoff time allocation. This FL allocation is performed according to the following steps:

- aggregation of flights into flows;
- computation of intersections in the horizontal plane between flows;
- allocation of a FL to each flow in such a way that two intersecting flows are given a different FL (if possible).

Of course, this phase cannot solve catch up conflicts within the same flow and doesn't take the climbing or descending parts of trajectories into account, so conflicts will still be detected after the allocation. These remaining conflicts will be solved by the following ground holding stage, which is expected to compute much better solutions once the FL allocation is optimized.

A. Flows and Intersections

Flights that share the same route (i.e. the same sequence of beacons in their flight plans) and RFL are gathered into one flow. We note $\mathcal{F}$ the set of all flights, and we define for each flow $\mathcal{F}_k \subseteq \mathcal{F}$:

- r_k its route;
- RFL_k its requested flight level;
- size_k its number of flights.

Note that the set of all flows $\mathcal{F}_k$ forms a partition of the set of flights $\mathcal{F}$.

Without taking time into account, we could define a (potential) conflict between two flows of aircraft $\mathcal{F}_k$ and $\mathcal{F}_l$ if their routes r_k and r_l intersects in the horizontal plane. However, this first approximation dramatically overestimates the number of conflicts and yields so over-constrained problems that only a very small subset of the constraints may be satisfied with limited flight level resources (i.e. if the distance between the allocated flight level and the RFL is bounded by a small constant, noted dev_{max} in the following section).

In order to reduce the tightness of the corresponding conflicts graph, the temporal dimension of the flights must be considered. Indeed, an aircraft does not occupy its entire route all day long, but intersects other trajectories during short time periods, as explained in the previous section. So two flows are in conflict if at least two flights (one from each flow) are in conflict w.r.t. definition 1 given in section III-B, but with a conflict detection performed in the horizontal plane only. Resulting conflicting intervals between flights i and j will be noted $\mathcal{C}^H_{ij}$ (by analogy with the $\mathcal{C}_{ij}$ in definition 2).

Such a conflict will occur if d_{ij} (i.e. the difference between the delays of flight i and j) belongs to the union of forbidden intervals $\mathcal{C}^H_{ij}$. So if no delay is considered, i.e. $\delta_i = 0$, $\delta_j = 0$ and therefore $d_{ij} = 0$, flights i and j will be in conflict iff $0 \in \mathcal{C}^H_{ij}$. Given two flows $\mathcal{F}_k$ and $\mathcal{F}_l$, we will note $conflict(\mathcal{F}_k, \mathcal{F}_l)$ the relation associated to the former definition. Of course, with this definition of conflicting flights and flows, potential delays allocated during the ground holding phase are not taken into account, and two flows which are not in conflict for the FL allocation model could be potentially conflicting during the slot allocation.

However, it is easy to generalize the conflict definition in order to take delays into account. If we note δ_{max} the maximal delay of the slot allocation phase as in section III-B, a conflict between two flights will be detected iff $[-\delta_{\text{max}}, \delta_{\text{max}}] \cap \mathcal{C}^H_{ij} \neq \emptyset$. Then the $conflict$ relation can be more formally defined:

$$conflict(\mathcal{F}_k, \mathcal{F}_l) \Leftrightarrow \exists i \in \mathcal{F}_k, \exists j \in \mathcal{F}_l, \text{ s.t. } [-\delta_{\text{max}}, \delta_{\text{max}}] \cap \mathcal{C}^H_{ij} \neq \emptyset \quad (3)$$

However, taking even very small delays (e.g. 5-15min) into account tends to generate very dense conflict graphs which are over-constrained for reasonable values of the maximal discrepancy dev_{max} allowed between allocated FLs and RFLs. Many conflicts will then remain for the ground holding phase and the purpose of FL allocation compromised. Furthermore, only a fraction of the flights will be delayed during the second phase, and many postponed flights are allocated very small delays, so it would be a too rough approximation to constrain each pair of intersecting flows as if the maximal delay were allocated to their conflicting flights. Eventually, the detection used to produce the results presented in section V is performed with $d_{ij} = 0$ (i.e. as if the flights were not delayed in the following phase), which allows to obtain solutions without conflicting flows for very low discrepancy bounds between allocated FL and RFL.

B. Constraint Model

The decision variables of the CP model are the flight levels allocated to flow k:

$$FL_k, \forall k \in [1, m]$$

where m is the number of flows. As an aircraft which does not fly at its optimal FL will burn more fuel during the flight, we limit the possible discrepancy between the RFL of a flow and its allocated flight level by a maximal deviation dev_{max}. So each FL_k has finite domain $[RFL_k - \text{dev}_{\text{max}}, RFL_k + \text{dev}_{\text{max}}]$.

The constraints of the model are straightforwardly derived from equivalence 3 of the previous section that defines conflicting flows:

$$\forall k, l \in [1, m]^2, \, k < l, \quad conflict(\mathcal{F}_k, \mathcal{F}_l) \Rightarrow FL_k \neq FL_l \quad (4)$$

As this model still may produce over-constrained instances, the constraint program uses *soft constraints* instead of hard ones to be able to compute partial solutions which do not satisfy all the disequality constraints appearing in 4. To obtain a solution which will ease as much as possible the task of the delay allocation, i.e. a traffic with as few 4D-intersections as possible between flows, the cost of a solution is defined as the number of remaining conflicting flows, i.e. flows which have a common FL with at least one of their intersecting flows.

$$\text{cost}_{FL} = |\{\mathcal{F}_k,\, k \in [1,m] \,\text{s.t.}\, \exists l \in [1,m],\, l \neq k,\\ conflict\,(\mathcal{F}_k, \mathcal{F}_l) \wedge FL_k = FL_l\}| \quad (5)$$

The soft constraint program allows then to produce the best possible solution w.r.t. the number of conflicting flows, and even if there is no solution with $\text{cost}_{FL} = 0$ for low values of dev_{max}, the resulting traffic will be much easier to deconflict during the following slot allocation phase than the raw traffic with its original RFLs.

However, a rough estimation of airlines operational cost of such a flight level allocation scheme should take the number and range of all the discrepancies from RFLs of the set of flights, so we define:

$$\text{op\_cost}_{FL} = \sum_{k=1}^{m} |RFL_k - FL_k| \times \text{size}_k \quad (6)$$

Even if the optimization criterion of our model only takes the former cost_{FL} into account, the search strategy attempts to minimize op_cost_{FL} by allocating flight levels as close as possible to their RFLs.

Note that the induced cost of a discrepancy to the RFL would probably raise more rapidly than a linear growth as proposed in equation 6, were more technical considerations taken into account to estimate a real fuel cost function. But as dev_{max} will be chosen very low during the experiments reported in section V to be acceptable for airlines (a deviation of 3, i.e. an FL of 10, 20 or 30 above or below the RFL, seems to be enough to solve all conflicts of most instances), the ordering of the solutions w.r.t. op_cost_{FL} should be quite similar to the one induced by a realistic operational cost function, the design of which is beyond the scope of this work.

V. Results

These two CP models have been implemented with the FaCiLe library [20] and have produced the following results on various days of traffic within the French airspace in 2008, with up to 8600 flights. After the conflict detection in the horizontal plane, flights are divided in flows and fed to the FL allocation program with a FL30 maximal discrepancy from the RFL and no delay taken into account; then the conflict detection in 3D is performed on the modified traffic and its result is taken as an instance of the ground holding program. About 10% of the flights are non-European flights, so their delays will be fixed to 0 during the ground holding phase as aforementioned.

A. FL Allocation Phase

Table I sums up a few dimensions of the instances like the number of flights, of flows (flights sharing the same route and RFL) and of conflict constraints between flows intersecting in the horizontal plane *and* temporally (with no delay taken into account). The instances typically comprises 3000 to 3500 flows with 160 000 to 200 000 constraints.

The maximal discrepancy dev_{max} is fixed to 3, which allows the FL of a flow to be allocated to FL10, FL20 or FL30 above or below its RFL. Such a low value is enough to obtain solutions without conflicts on all the days of traffic of our data set and should be acceptable for airlines. The typical distribution of discrepancies to RFL after allocation for the various days of traffic is presented in figure 4: 60% of the flights remain untouched, and among deviated flights, 70-80% have a discrepancy of FL10 and 1-5% only the maximal discrepancy of FL30.

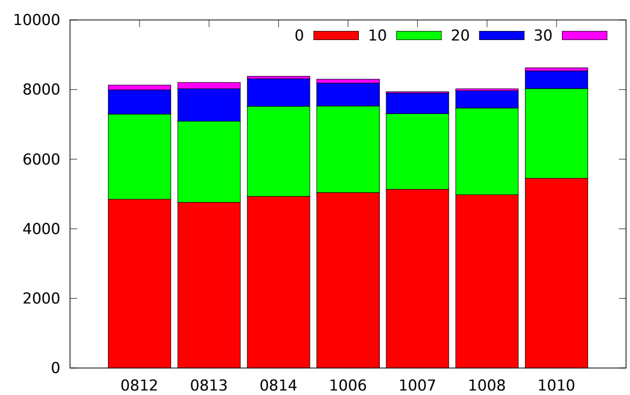

Fig. 4. Distribution of discrepancies from RFL after allocation.

The operational cost op_cost_{FL} defined in equation 6 is not directly optimized by the CP branch & bound algorithm, as mentioned in section IV-B, but the search strategy first assigns flight levels closest to the RFL of each flow to obtain allocations with reasonable discrepancies. The last line of table I shows the operational cost obtained for the days of our data set.

To assess the allocation phase and to produce the new trajectories data for the ground holding phase, the CATS simulator is fed with the flight plans and the new FLs with its conflict detection activated. Figure 5 shows the evolution on the number of actual conflicts detected by CATS, on the 12/08/2008 traffic, with increasing values of dev_{max}: 0 corresponds to the original RFLs, i.e. without FL allocation, and 30 to the maximal discrepancy

TABLE I

INSTANCES DIMENSIONS AND CORRESPONDING OPERATIONAL COST.

day	08/12	08/13	08/14	10/06	10/07	10/08	10/10
flights	8130	8204	8384	8297	7941	8021	8624
flows	2921	3065	3168	3146	3112	3156	3406
conflicts	16579	17775	17641	18671	16613	16515	19717
op_cost_{FL}	42550	47360	43910	41330	34770	36480	38530

value used for the solutions given to the ground holding phase. The conflicts are divided in two main types:

- "Evol" for conflicts where at least one of the two flights is climbing or descending;
- "Cruise" for conflicts where the two flights are in the cruise phase;

then each of the previous categories are further divided in two subtypes:

- "Crossing" for conflicts where flights are on distinct crossing segments;
- "Catch up" for conflicts where aircraft are on the same flight segment between two beacons.

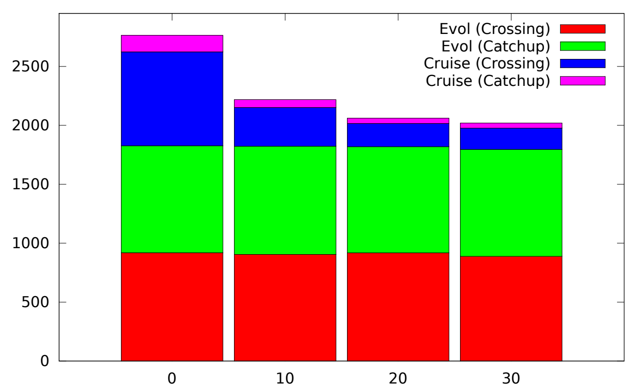

Fig. 5. Maximal discrepancy from RFL dev_{max} vs number of remaining conflicts on 12/08/2008.

As expected, the total number of conflicts decreases as the maximal discrepancy dev_{max} grows, essentially among conflicts occurring during the cruise phase of the flights. Indeed, the FL allocation model approximates trajectories without taking climb or descent phases into account, so almost all conflicts occurring with one vertically evolving aircraft will not be solved. Nevertheless, the total number of conflicts is reduced by 20% for a FL10 maximal discrepancy, and a discrepancy of FL30 was necessary to solve all conflicting flows of the FL allocation model, inducing a 27% reduction of the conflicts.

Of course, the effect is much more sensible if only conflicts occurring during the cruise phase are taken into account, as 60% of them are solved by a FL10 discrepancy, and up to 75% for FL30 (81% if only crossing conflicts are counted). But the ratio of conflicts with evolving aircraft is around 65% for a typical day of traffic like the one of figure 5, and as our FL allocation scheme is unable to solve this heavy share of the conflicts, they will have to be taken care of by the ground holding phase. After FL allocation, the share becomes close to 90%.

B. Ground Holding Phase

The CP ground holding algorithm takes as input the trajectories of one day of traffic modified by the previous FL allocation with a FL30 maximal discrepancy. Figure 6 shows the variation of the cost, i.e. the maximal delay needed to solve all conflicts, before and after the FL allocation phase. The decrease of the cost can reach more than 50% for the first instance, or as bad as 0% for the second – and could possibly raise, even if we did not observe it on our data set.

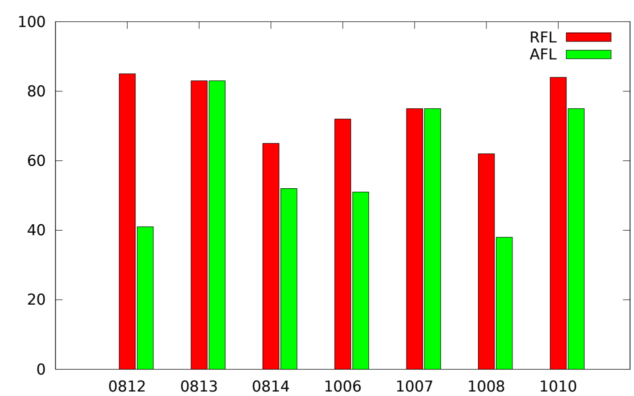

Fig. 6. Optimal cost before and after FL allocation.

However, it was already mentioned in [7] that the maximal cost was rather aimed at proving the feasibility of the delay allocation but was a poor indicator alone of the quality of a solution, and that the sum of the delays and the ratio of delayed flights give much more information. Moreover, it was observed as well that the maximal cost exhibits very steep variations when the algorithm is parameterized to select flights and conflicts above a given flight level only (which allows to tune the size), unrelated to the smooth decrease observed for the sum of delays.

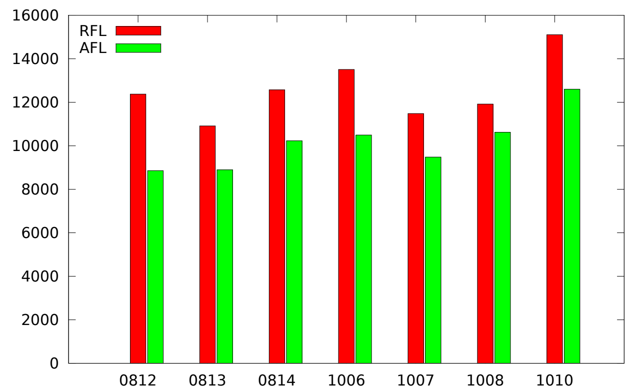

Fig. 7. Sum of delays before and after FL allocation.

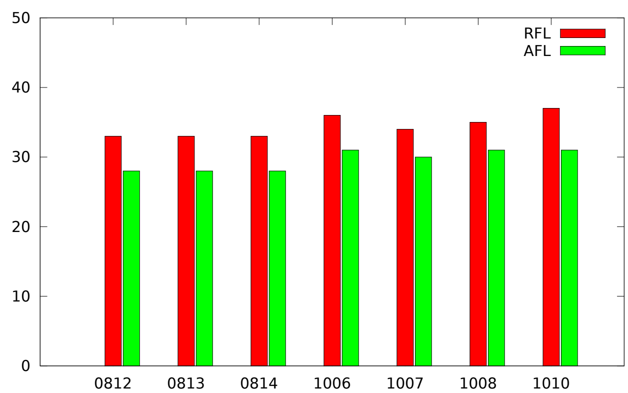

Fig. 8. Percentage of delayed flights before and after FL allocation.

Nevertheless, the decrease in terms of the sum of delays or ratio of delayed flights is much more sensible: 11% to 28% is gained on the sum, with a 20% mean, and a reduction of 12% to 16% of the percentage of delayed flights (i.e. an average improvement of 5% of the total amount of flights).

VI. Further Work

The results of this combination of FL allocation and ground holding are encouraging. However, we have only addressed so far the resolution of conflicts within the French airspace. 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. We plan to experiment with various refinements of our algorithm to be able to address such large scale problems, as well as to handle uncertainties on takeoff times and trajectory management.

A. Handling Uncertainties

To improve the robustness of our solutions towards uncertainty on takeoff times, the conflicting intervals described in section III-B can be extended by a fixed amount of time ext, thus allowing the allocation process to handle a $\pm\frac{ext}{2}$ uncertainty. The resulting constraints would be tightened compared to the original model, so that the cost of our solutions would increase significantly.

This method was tested in [7] without the prior FL allocation, showing a costly increase of the cost with the ext parameter. We plan to experiment this feature with FL allocation taken into account to test whether the cost increase scheme is improved through this combination.

It also possible, as seen in section IV-A, to take some delay into account for the flight level allocation. Handling large delays seems to yield over-constrained instances. However, since most flights are allocated a small delay (typical figures for the problems we considered are: 85% flights with a delay less than 3 min, 90% with a delay less than 5 min), we plan to study the effect on our solutions of taking such small delays into account for the FL allocation.

In order to handle takeoff time uncertainties and other operational hazards such as flight cancellations or bad weather, we also plan to adapt our algorithm to proceed on a sliding time window. Slices of the problem would be iteratively solved on a limited time window T_w, then only the earliest part of the solution would be kept over a small interval λ and then the resolution window would slide by λ. Each time the window is shifted, takeoff times can be updated, cancelled flights can be discarded and other constraints can be taken into account. Parameters T_w and λ must be carefully chosen according to the computation time of the resolution. A similar approach is used for dynamic conflict resolution in the CATS solver as described in [17].

B. European Instances

Our algorithm was not able to address European instances with ground holding only. However, we expect that the combination of ground holding with FL allocation as described in this paper will allow such large instances to be solved with a reasonable amount of delay.

Also, the use of a sliding time window method as described above would help handling such instances by slicing it into smaller ones.

VII. Conclusion

We have presented a novel combination of flight level allocation and ground holding to solve all potential conflicts for a day of traffic in the French airspace which improves the results published in [7]. Rather than trying to respect sector capacity constraints, we model each possibly conflicting situations between any two aircraft and impose adjustments of FL and departure times to keep them separated, with the hypothesis that aircraft could precisely follow their planned 4D-trajectories.

The size of the corresponding combinatorial problem is huge, but our CP algorithm is able to reach optimal solutions for the two phases. The maximal delay, overall delay sum and ratio of delayed flights are comparable to delays allocated by the CFMU, while discrepancies from the RFL are tightly bounded.

However, this preliminary work does not yet take into account uncertainties on the 4D-trajectories accuracy. We plan to add robustness towards uncertainties on the takeoff times to our approach by tightening the conflict graph of the FL allocation with small delays taken into account during conflict detection, adding the so-called "conflict extension" presented in [7] for the ground holding phase and iteratively solving temporal slices of the problem.

References

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

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

[3] Robert Graham and David Young. Preparing an initial assessment of the SESAR concept of operations "EP3: Single european sky implementation support through validation". Technical report, Eurocontrol Experimental Centre, France, 2006.

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

[5] Nicolas Barnier, Pascal Brisset, and Thomas Rivière. Slot allocation with constraint programming: Models and results. In *International Air Traffic Management R&D Seminar ATM2001*, Santa Fe (NM), USA, December 2001.

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

[7] Nicolas Barnier and Cyril Allignol. 4D-Trajectory deconfliction through departure time adjustment. In *International Air Traffic Management R&D Seminar ATM2009*, Napa (CA), USA, July 2009.

[8] Vincent Letrouit. *Optimisation du réseau de routes aériennes en Europe*. PhD thesis, Institut National Polytechnique de Grenoble, 1998.

[9] Nicolas Barnier and Pascal Brisset. Graph coloring for air traffic flow management. In *CPAIOR'02*, pages 133–147, Le Croisic, France, March 2002.

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

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

[12] 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 *USA/Europe Air Traffic Management R&D Seminar ATM2007*, Barcelona, Spain, July 2007.

[13] 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.

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

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

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

[17] Géraud Granger, Nicolas Durand, and Jean-Marc Alliot. Optimal resolution of en route conflicts. In *International Air Traffic Management R&D Seminar ATM2001*, Santa Fe (NM), USA, December 2001.

[18] David Gianazza and Nicolas Durand. Assessment of the 3d-separation of air traffic flows. In *USA/Europe Air Traffic Management R&D Seminar ATM2005*, Baltimore (MD), USA, June 2005.

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

[20] 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.

Authors

Dr Nicolas Barnier is a lecturer and research assistant at ENAC. He graduated from ENAC as an engineer in 1997, and received a PhD in computer science in 2002 from the University of Toulouse. He is one of the authors of FaCiLe, an open source Constraint Programming library for the functional language O'Caml. His research interests focus on Constraint Programming, Local Search and Combinatorial Optimization in general.

Cyril Allignol is currently a PhD student in the Planning, Optimization, and Modelling team of the DSNA/DTI R&D domain. He received his engineer's degrees (IEEAC in 2006) from the french civil aviation academy (ENAC) and his M.Sc. (2006) in computer science from the "Institut National Polytechnique de Toulouse" (INPT).