Slot allocation in Air Traffic Flow Management

Nicolas Barnier, Pascal Brisset

Slot Allocation in Air Traffic Flow Management

Nicolas Barnier and Pascal Brisset
ENAC - CENA, Toulouse, France
barnier,brisset@recherche.enac.fr
http://www.recherche.enac.fr/opti

Abstract

Current European Air Traffic Control (ATC) system is far exceeded by the demand and the resulting delays are a financial and psychological burden for airlines and passengers. The Central Flow Management Unit (CFMU), which is in charge of regulating the flights to respect operational en-route capacity constraints of Air Traffic Control Centres (ATCC), fails to allocate efficiently departure slots and merely satisfy a weak and unrealistic modelisation of the fuzzy-stated workload constraints. Disregarding the minimization of the sum of the delays, this paper presents new models, based on the Constraint Programming paradigm, of the slot allocation problem focused on the controllers workload: an extension of the current model with a standard formulation, and a novel approach involving the *sort* constraint, both able to maintain workload constantly below a given capacity and the latter also providing efficient failure proof on over-constrained instances. The behaviours of the different models are discussed with partial and full instances from real French air traffic data set and we show the potential operational improvement supplied by these *continuous* models.

1 Introduction

Airspace congestion is today the most critical issue European Air Traffic Management (ATM) has to face. French Air Traffic Control Centres (ATCC) capacities are far exceeded by a constant growth in air traffic demand, resulting in ever increasing flight delays. These time and management costs are such a nuisance for all airlines and passengers that the European Commission

has just released a special statement (December 1st, 1999) acknowledging that current ATFM systems are unable to support high traffic loads and unscalable for the predicted growth. The following economic loss is evaluated by the Commission at more than 5 billions Euros for the past year and drastic improvements must be achieved to overcome the problem [IP/99/924, 1999].

The Central Flow Management Unit (CFMU) in Brussels is in charge of reducing these congestion costs by, among many other strategic or tactical measures, delaying departure slots for the flights involved in overloaded sectors. The purpose of delaying is to respect the en-route capacity constraints provided by each ATCC according to their daily schedule. These capacities are expressed as a number of planes per hour and used by the CFMU with 30 min periods on a first-demand first-served basis to allocate the regulated departure slots (i.e. "ground holding"). As the main optimization criterion is the sum of the delays, concerned flights may be very unevenly distributed over the 30 min time period so that controllers have to cope with traffic peaks when the periods start. Therefore capacity constraints are eventually (actually most of the time) not satisfied for time periods which do not begin at multiples of 30 min from the start time of a given sector capacity constraint. Moreover, the regulation for a given flight is computed with respect to the sector inducing the greatest delay, which often leads to capacity constraints violation for the other sectors crossed by the same flight.

As the statements of capacity constraints are rather ambiguous and obviously not very satisfactorily interpreted by the current ATFM system from an operational point of view, a better modelisation of these fuzzy requirements would be to maintain controllers workload below the specified capacities continuously. We provide for this aim several new models endowed with a more realistic, thus harder, satisfaction of capacity constraints:

- sliding discrete windows which allow to smooth the profile of controllers workload from the weakest constrained model (the current CFMU model) to the hardest (the satisfaction of capacity constraints over any time period of a given length);

- "sorting" model which states continuously the constraints over the ranks of the entry time of flights within a given *sector-period*, i.e. a sector for particular period of time and capacity.

Constraint Programming allows a very straightforward formulation of these two models, as well as for the CFMU model which is presented for comparison purpose. The versatility of CP technology eases fast implementation and alternatives testing. An all-purpose constraint library has been

used to experiment with the various models, leading to drastically different allocation schemes.

The paper is divided as follows: we first give a precise description of the problem and some indication about the size and complexity of the instances we are interested in. Then we present several models, starting with standard ones and refining them with a more continuous formulation. The next section is devoted to the results for simplified and real instances where behaviour of the different models are compared. We conclude with an overview and some hints about possible future work.

2 Slot Allocation Problem Description

Air traffic flow management is a daily pre-tactical filter intended to regulate scheduled flights across controlled airspace. Its aim is to limit the number of planes in a given space over a given period. This planning will be precisely scheduled by the air-traffic controllers in real-time.

The ATFM problem is described in terms of

- *Flights*: A flight starts from one airport at a specified time, follows a predefined route at a fixed speed and arrives at another airport.

- *Sectors*: The airspace is divided into sectors crossed by the routes of the flights. A sector is a 3D polyhedra, usually a vertical cylinder endowed with a *capacity* expressed as the maximum number of flights entering the sector during a time period (usually one hour). The partition of the airspace, i.e. the number and the shape of the sectors, varies during the day. The capacity of the same sector may also changes at given times. We call *sector-period* a sector for particular period of time and capacity.

Figure 1 gives a glance at a slice of the French airspace at 24000 ft.

The constraints of the problem are the capacities of the sectors. There are several degrees of freedom to satisfy these constraints: choosing different routes for flights, delaying departures, changing speed of the aircrafts during flight, asking aircrafts to hold their position... In this paper, we focus only on the ground-holding policy: each flight may be delayed at departure. ATFM is in charge of solving the problem on a day-by-day basis.

The difficulty of the problem does not come from the complexity of its constraints but from its size. We have used real data archived by the French civil aviation tool COURAGE and solved the ATFM problem for May 20th, 1999:

- The French airspace is concerned by 7375 flights entering between 0h01 and 23h59.
- 140 sectors are active during the day, some of them with various capacities (up to 6 different).
- More than 700 flights may enter a single sector-period.
- Capacities vary from 19 to 52 flights per hour.

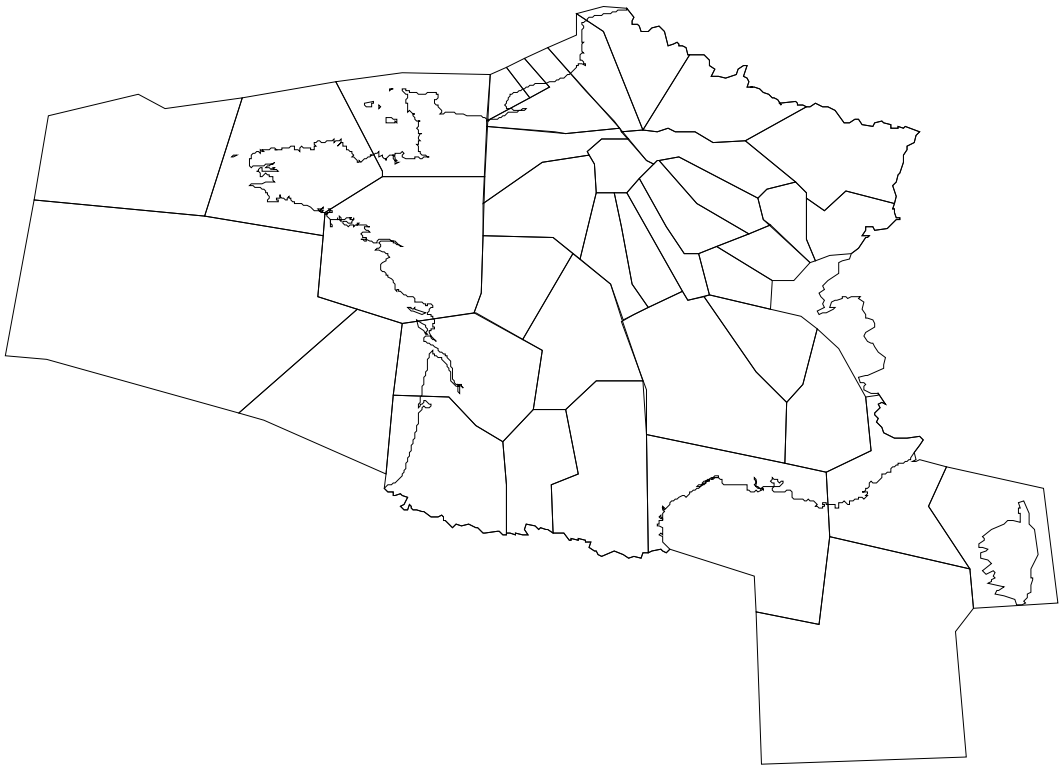


Figure 1: Sectors in the French airspace at flight level 240 (7200 m)

The objective of the slot allocation problem is to reduce the delays. There are several ways to achieve this purpose: reducing the total sum of the delays (utility), the maximum delay (equity), the average delay, etc. [Maugis, 1996] gives an extensive description of what could and should be a cost function and finally concludes with results for the simplest one, the total sum of delays. It is also the choice of [Bertsimas and Stock, 1995] with slightly different model. In this paper, we are more interested in the qualitative properties of the solution than in its numerical cost.

3 Four Models

In this section, we describe four models for the slot allocation problem. These models are not equivalent and differ according to the interpretation of the sector load constraints.

The models are described with the following quantities:

- $\mathcal{S}$: the set of sector-periods, each with a *start* and an *end*;
- $\mathcal{F}$: the set of flights;
- t_i^s: the time the flight i enters the sector s according to the original timetable;
- $capa^s$: the capacity of the sector-period s (flights/hour);
- δ: time base for the capacity constraint (minutes).

All the models use the main decision variables

- D_i: departure delay for flight i.

We present two pairs of equivalent models. The first one is the one commonly used [Maugis, 1996].

3.1 Non Overlapping Windows

We consider the load constraint in successive contiguous periods:

- $\mathcal{P}^s = \{p_0^s, p_1^s, ...\}$: successive periods of length δ, each with a *start* and a *end*. The first period starts with the sector-period: $start(p_0^s) = start(s)$.

In an other way:

$$\mathcal{P}^s = \{[start(s), start(s) + \delta[, [start(s) + \delta, start(s) + 2\delta[, ...\}$$

We present in the next sections two equivalent formulations with non overlapping windows.

3.1.1 Boolean Variables

The `Basic` model is written with auxiliary boolean variables:

- B_{i,p_j^s}: flight i enters the sector s during the period p_j^s.

A first constraint relates the delay variables with the boolean variables. A second one sets the sector load capacity.

$$\begin{array}{ll} \forall s \in \mathcal{S}\ \forall p \in \mathcal{P}^s & B_{i,p_j^s} \text{ iff } start(p) \leq t_i^s + D_i < start(p) + \delta \\ \forall s \in \mathcal{S}\ \forall p_j^s \in \mathcal{P}^s & \sum_{i \in \mathcal{F}} B_{i,p_j^s} \leq capa^s \end{array}$$

The drawback of this model is a huge amount of boolean variables:

$$|\{B_{i,p_j^s}\}| \quad = \quad |\mathcal{F}| \sum_{s \in \mathcal{S}} \frac{end(s) - start(s)}{\delta}$$

The next model avoid this drawback using a global constraint.

3.1.2 Global Cardinality Constraint

A Global Cardinality Constraint (gcc) is specified in terms of

- a set of variables $X = \{X_1, ...X_n\}$;
- a set of values $V = \{v_1, ...v_d\}$;
- a set of cardinals $C = \{C_1, ...C_d\}$

and constrains the number of times the variables in X take values in V. Precisely, the constraint $gcc(C, V, X)$ says that $\forall i\ |\{x \in X | x = v_i\}| = C_i$. [Régin, 1996] has proposed an efficient filtering algorithm for this constraint. It is implemented in the `IlcDistribute` global constraint of ILOG Solver [Solver, 1999].

The gcc provides an easy way to reformulate the basic model for slot allocation. The trick is to be able to compute the period in which a flight enters the sector according to its delay. Because in the basic model the periods constitute a regular partition of the time, the period number can be computed with a simple arithmetic operation: if successive periods of length δ are numbered $0, 1, ..., j, ...$, the period of a date t is $\lfloor t/\delta \rfloor$. Then, setting $V_j = j$, it is enough to constrain C_j to be less than the capacity of the period j.

Using the gcc, we get a model with integer auxiliary variables and values:

- X_i^s : period index of entry time in sector-period s for flight i;
- $V_j^s = j$: index of the jth period in sector-period s.
- $C_j^s = 0..capa^s$: number of flights entering the sector-period s during the jth period.

For each sector-period, we need constraints to define auxiliary variables and one gcc.

$$\begin{aligned} \forall s \in \mathcal{S}\ \forall i \in \mathcal{F} \quad & X_i^s = (t_i^s + D_i)/\delta \\ \forall s \in \mathcal{S} \quad & gcc(C^s, V^s, X^s) \end{aligned}$$

Using a global constraint has several advantages compared to a hard-coded solution :

- the model is simpler to write and easier to understand;
- the number of constraints and the number of variables are reduced;
- efficient propagation can be done.

However, a global constraint may be too specific and hard to customize for a slightly different model.

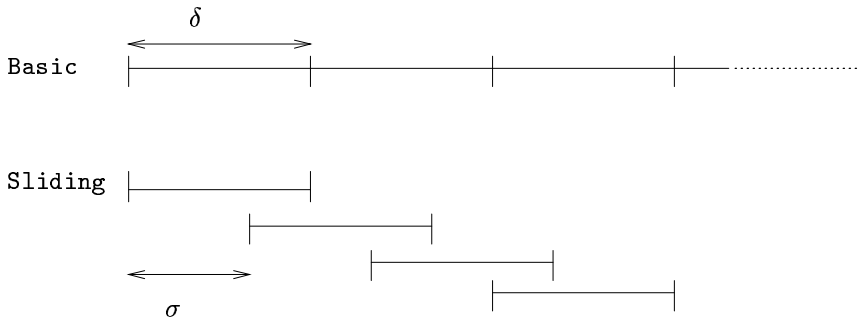
3.2 Continuous Models

The models previously described suffer from strong discontinuity. They only state that the sector load limit must be satisfied at the beginning of each period, usually 24 times a day. Because the main objective of the cost function is to reduce the delays, the expected side-effect of this model is a concentration of flights entering at the beginning of the periods. This is confirmed by the experiments (see section 4.2.1).

We propose two new formulations. The first one comes straightforwardly from the `Basic` one. The second one relies on a sorting global constraint.

3.2.1 Sliding Periods

Keeping the same idea of the `Basic` model, it is possible to get a more continuous model with overlapping periods.



Here we use another parameter (σ) which is the time step between periods:

$$\mathcal{P}^s = \{[start(s), start(s) + \delta[, [start(s) + \sigma, start(s) + \sigma + \delta[, ...\}$$

Of course this model supersedes the `Basic` one and is equivalent for $\sigma = \delta$. It can be implemented with the same kind of boolean variables. Unfortunately, the global gcc constraint cannot be used because the trick does not work: a flight may appear in several periods.

3.2.2 The Sorting Constraint

Let D be a totally ordered set. The sort constraint is the relation associated with the standard functional sort function. A sort function takes a sequence of length n containing elements in D and returns another sequence containing the same elements ordered. The sort constraint "relates" two sequences containing finite domain variables taking values in D.

Example: let $S_1 = \{[0-13]; [6-10]; [10-11]; [4-16]; [4-6]\}$ and $S_2 = \{[1-3]; [5-10]; [6-9]; [11-17]; [10-15]\}$ be two sequences of interval variables. Setting the constraint

$$sort(S_1, S_2)$$

must lead to the following refinements: $S_1 = \{[1-3]; [6-9]; 11; [11-15]; [5-6]\}$ and $S_2 = \{[1-3]; [5-6]; [6-9]; 11; [11-15]\}$.

[Guernalec and Colmerauer, 1997] proposed an efficient filtering algorithm for this constraint. Surprisingly enough, the complexity of this complete (the propagation refines the domains as much as possible) narrowing algorithm has an optimal complexity of $\mathcal{O}(n \log n)$.

The algorithm is described in six steps involving quite complex components like sorts and balanced binary tree data structures. The narrowing is done only on the bounds of the variables.

For our problem, we use one sorting constraint per sector-period. The constraint is set on auxiliary variables: the entry times and the sorted entry times:

- T_i^s: actual entry time of flight i in sector-period s;
- S_j^s: entry time of the jth flight entering the sector-period s.

The constraints relate the auxiliary variables to the main delay variables and set the sector load limit: two flights distant of *capa* in the ordered sequence must be distant of at least δ minutes:

$$\begin{aligned} \forall s \forall i \in \mathcal{F} \quad & T_i^s = t_i^s + D_i \\ \forall s \quad & sort(T^s, S^s) \\ \forall s \forall j \in \{1, .., |\mathcal{F}| - capa^s\} \quad & S_j^s + \delta \leq S_{j+capa^s}^s \end{aligned}$$

The previous constraint for the load limit is not totally correct because it is set even if one of the two concerned flights enters the sector before or after the period of the sector. A solution is to relax the constraint using auxiliary boolean variables:

$$\begin{aligned} \forall s \in \mathcal{C} \; \forall j \quad & B_j^s \text{ iff } start(s) \leq S_j^s < end(s) \\ \forall s \in \mathcal{C} \; \forall j \in \{1, .., |\mathcal{F}| - capa^s\} \quad & S_j^s + \delta \leq S_{j+capa^s}^s + (2 - B_j^s - B_{j+capa^s}^s) * \delta \end{aligned}$$

The boolean variable B_j^s states that the sector-period s is concerned by the flight j. If one of the two flights is out of the period, the constraint is relaxed thanks to the second term of the right hand side.

4 Experiments

4.1 Implementation

Three of the four models have been implemented using a finite domain library we have written using the Objective Caml system [Leroy, 1999]. This strongly typed functional language provides well documented libraries for various data-structures. A fast compiler produces efficient native code.

Our constraint library includes standard finite domain variables, arithmetic constraints [Harvey and Stuckey, 1998], efficient arc-consistency propagation on binary constraints (AC6) [Bessiere and Cordier, 1993] and some global constraints: difference with filtering based on a bipartite graph matching algorithm, sorting, ... The search is controlled in a Prolog way with goals (success and failure continuations) and cuts.

The `Gcc` model has been implemented with ILOG solver using the `IlcDistribute` constraint.

4.2 Results

We give in this section some results obtained for the different models applied on real data. All the data concern May 20th, 1999.

The same labeling is used for all the models : standard labeling on the delay variables (D_i's) sorted with departure date.

All the experiments are done with a precision ϵ (time unit) of 5 min and a maximum delay of 60 min.

4.2.1 A Single Sector

In order to analyse the solutions of the different models, we first looked at a very simplified problem with a single sector with no variation of capacity. We chose the sector of May 20th, 1999 concerned by the maximum number of flights (644). The hourly capacity is 40. The flights are expected to enter in the sector between 0h52 and 23h39.

It was first possible to check the equivalence and order between models (`M1` $\leq$ `M2` means that `M2` is more constrained than `M1`, i.e. that a solution for `M2` is a solution for `M1`):

- `Basic` $\equiv$ `Gcc`;
- `Basic` $\leq$ `Sorting`;
- `Sliding` $\leq$ `Sorting`;
- `Sliding` with $\sigma = \epsilon \equiv$ `Sorting`.

Table 1 reveals the corresponding numerical results: a more constrained model get a higher cost. We also see in this table that most of the flights are not delayed flights or get a small delay (compare to 644 concerned flights). Note that the average delay for all the solutions is smaller than the time precision (5 minutes).

Model	δ	σ	$\sum$Delays	Delay=0	Delay$\leq$15
Basic, Gcc	60		690	602	624
Basic, Gcc	30		1960	532	595
Sliding	60	30	1760	549	597
Sliding	60	15	2480	504	565
Sorting, Sliding	60	ϵ	3660	467	553

Table 1: Delays for different models and parameters

Numerical cost is a poor indicator about a solution. The expected differences between the four models are qualitative. In figures 2 and 3 is plotted the instantaneous load of the sector, i.e. the number of flights which will arrive during the next δ period. Dots correspond to the unregulated flow (the initial data) while lines show the solutions.

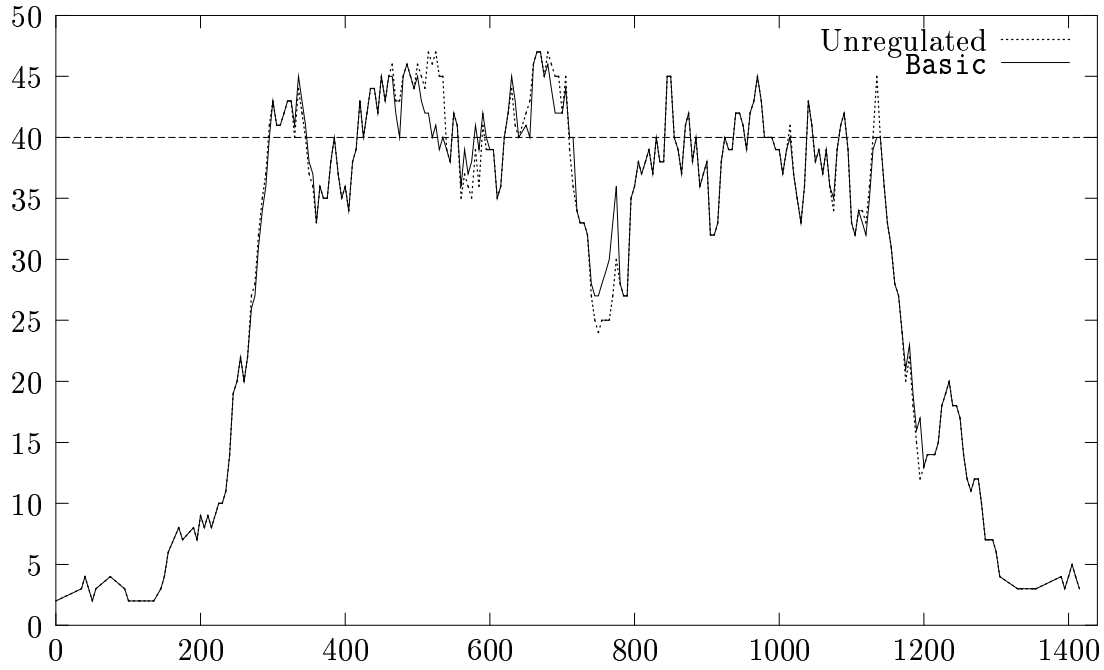


Figure 2: Regulation with the `Basic` model. Flights entering during the next δ min.

At a first glance the `Basic` model does not seem to regulate anything. In fact, it only insures that the curve goes under the capacity line every δ minutes (every hour starting from 0).

The figure 3 for `Sorting` shows more interesting and expected results. The solution gives an almost constant instantaneous load from 5h00 to 20h00.

Figure 4 shows the influence of the σ parameter on the `Sliding` model. With $\sigma = \delta = 60$, we get the solution of the `Basic` model. With a smaller σ (15), we observe that the load goes under the capacity every σ minutes but has still "time" to go up 10% over the capacity in between.

Figure 5 shows the side-effects of the `Basic` model. In this experiment, we reduced the capacity (10%) and augmented the max delay (120 min) to force more flights to be delayed. We plot here the instant number of flights inside the sector (one can notice that a plane does not stay for a long time

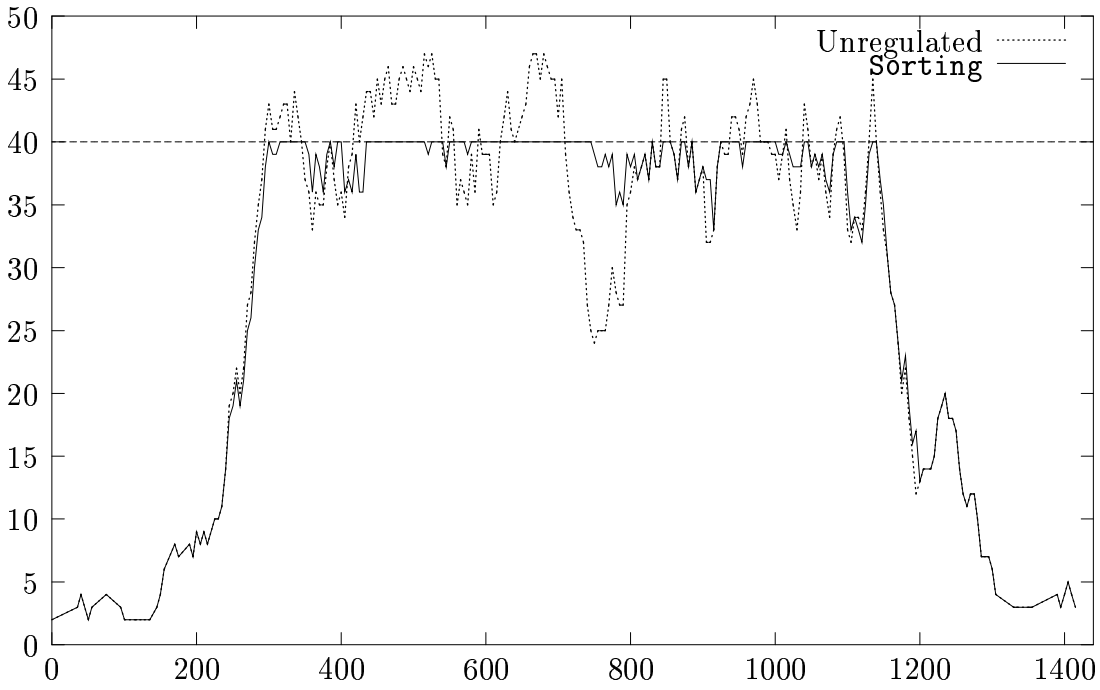


Figure 3: Regulation with the `Sorting` model. Flights entering during the next δ min.

in a sector). Peaks occur at the beginning of this periods: a delayed flight has to be scheduled during the next non-full period; then a minimum delay corresponds to the beginning of the period. The `Sorting` model does not suffer from this side-effect[1] and ensures an even load.

Table 2 gives some indication of the ability of the models to prove that a problem has no solution. With a reduced capacity of 36, the failure could not be proved in a "finite time" (∞ in the table) except using the `Sorting` model. Note the problem seems to have a phase transition since a solution is easily found with a slightly greater capacity (37).

4.2.2 Full Problem

We were able to prove that there is no solution for the `Sorting` model with the given capacities. But solutions can be obtained with allowing an *overload* of the capacity of each sector. We found a solution with a capacity overload of 25%.

The figure 6 gives an overview of the traffic activity over the Brest control

[1]Today, air traffic controllers confirm this periodicity of the current regulation.

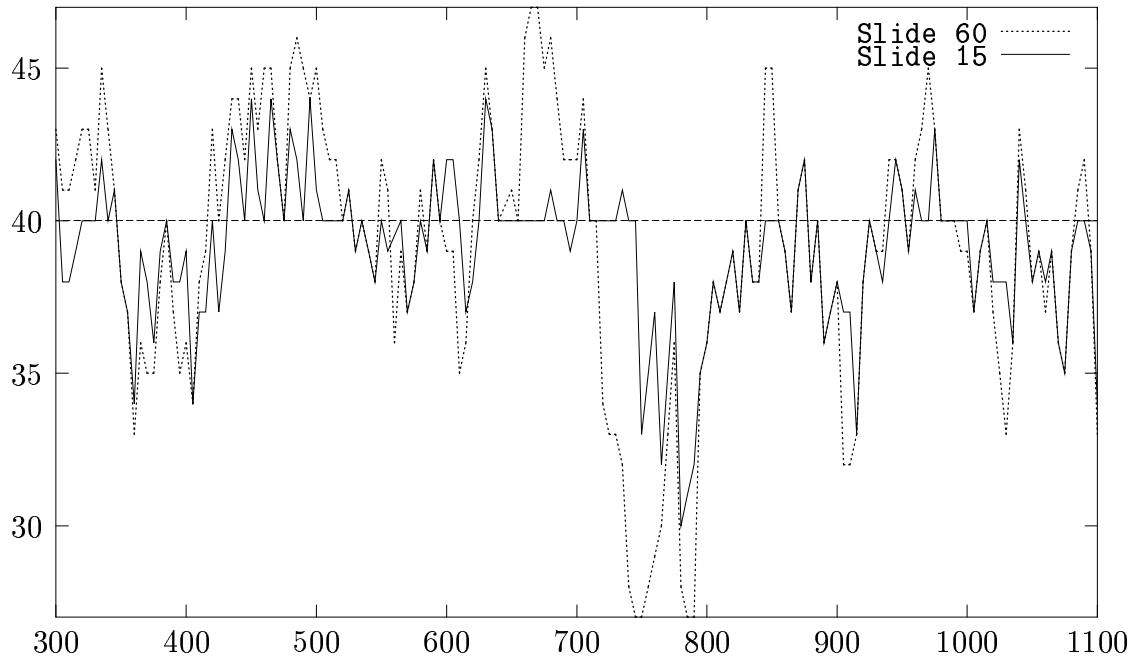


Figure 4: Regulation with the `Sliding` model: Influence of σ (zoom between 5h and 19h20). Flights entering during the next δ min.

Model	Capacity	Result
`Basic`	36	∞
`Gcc`	36	∞
`Sliding` ϵ	36	∞
`Sorting`	36	Failure proved
`Gcc`	37	7390
`Sorting`	37	19775

Table 2: Proof of impossibility

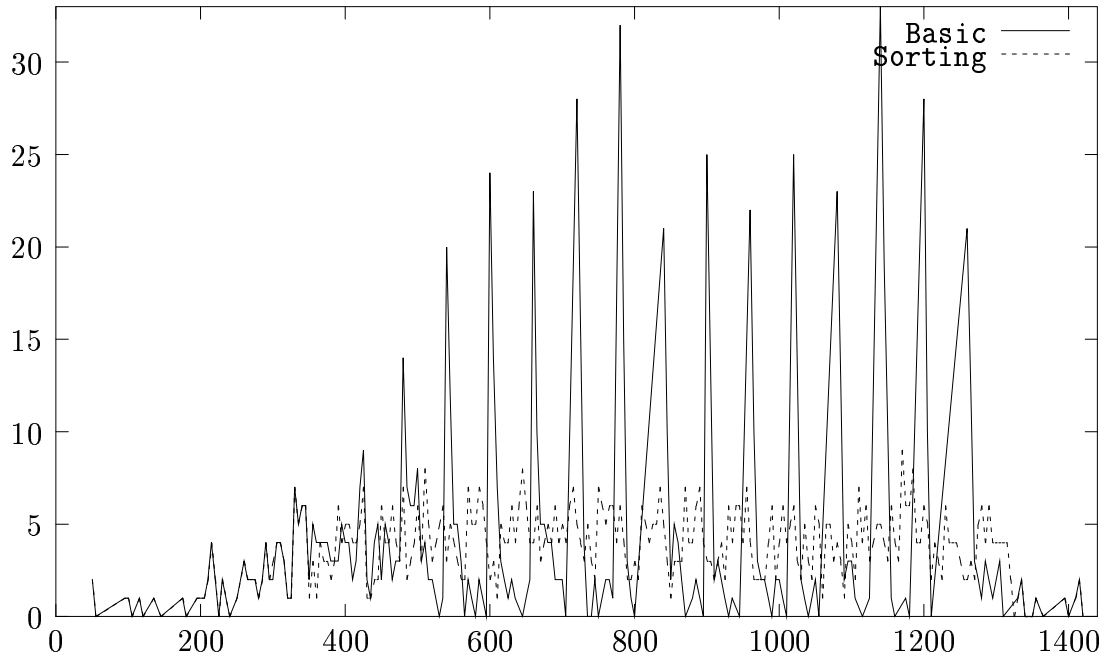


Figure 5: Number of flights in the sector: Traffic peaks at the beginning of each period for the `Basic` model (capacity = 36; delay max = 120).

centre during the day. Figure is split vertically into sectors, horizontally in time. Boxes in the figure are sector-periods. In each box is plotted the number of flights planned to enter during the next 60 minutes. For example, only 3 sectors are open in the the morning (upper left). They are later closed and replaced by 5 others, themselves replaced soon by 8 others, then 9, etc. The regulation has no spectacular qualitative effect like in the simplified case; the whole problem is too much complex to have simple local properties (full load of a sector during a long period for example, maybe except sector `J` around noon which get a constant load). The Brest control centre is one of the simplest among the 6 French ones.

5 Conclusion

Time slot allocation in ATFM is a hard combinatorial problem yet poorly solved by current CFMU systems. Capacity constraints are subject to interpretations and Constraint Programming provides an efficient and straightforward way to implement various models addressing control workload with different operational points of view. Non overlapping windows hardly cope

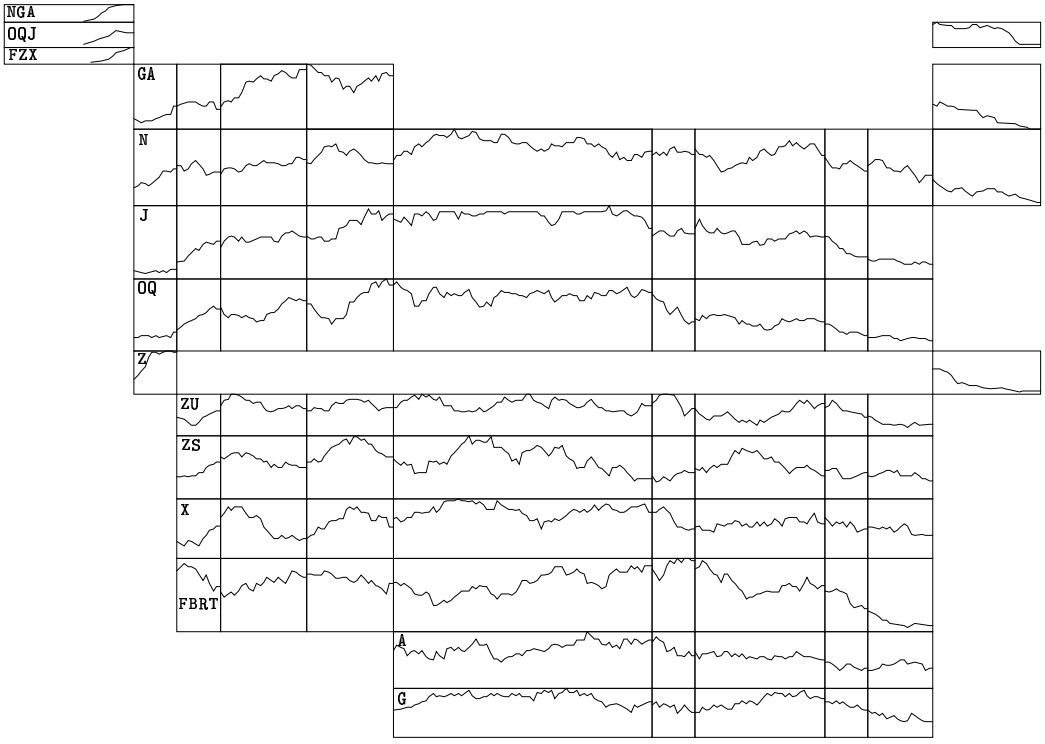


Figure 6: Brest Airspace Control Center - May 20th, 1999

with a realistic semantic of capacity constraints and probably lead Control Centres to underestimate their capacities because of periodic traffic peaks; whereas ϵ-sliding windows and sort models ensure an even regulation inducing a much more constrained CSP with higher delay costs. The sort model however needs fewer control parameters than its time windows counterpart. It is also far more efficient on fail proofs because of very powerful constraint propagation while offering similar computation time performances.

Large size and high dimensionality of air traffic input data make the slot allocation problem hard to optimize with respect to the cumulated delays criterion and future work should address this issue more efficiently. But the integration in the objective function of other factors (which may also be stated at the constraint modeling stage) like the regularity of workload distribution seems to be an important operational requirement and would penalize the basic models.

References

[Bertsimas and Stock, 1995] Bertsimas, D. and Stock, S. (1995). The air traffic flow management problem with enroute capacities. Technical report, MIT.

[Bessiere and Cordier, 1993] Bessiere, C. and Cordier, M. (1993). Arc-consistency and arc-consistency again. In *AAAI*.

[Guernalec and Colmerauer, 1997] Guernalec, N. B. and Colmerauer, A. (1997). Narrowig a $2n$-block of sorting in $O(n \log n)$. In *Principles and Practice of Constraint Programming*. Springer-Verlag.

[Harvey and Stuckey, 1998] Harvey, W. and Stuckey, P. J. (1998). Constraint representation for propagation. In Maher, M. and Puget, J.-F., editors, *Principles and Practice of Constraint Programming*, pages 235–249. Springer-Verlag.

[IP/99/924, 1999] IP/99/924 (1999). European commission: Fifteen countries, a single European sky, (`http://europa.eu.int/comm/pr_en.htm`).

[Leroy, 1999] Leroy, X. (1999). The Objective Caml System: User's and reference manual (`http://caml.inria.fr`).

[Maugis, 1996] Maugis, L. (1996). Mathematical programming for the air traffic flow management problem with en-route capacities. In *in Proceedings of the 14th Triennial World Conference of the International Federation of Operational Research Societies.*

[Régin, 1996] Régin, J.-C. (1996). Generalized arc consistency for global cardinality constraint. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence.*

[Solver, 1999] Solver (1999). ILOG Solver 4.4 user's manual (`http://www.ilog.fr`).