Air-Traffic Complexity Resolution

Pierre Flener, Justin Pearson, and Magnus Ågren
Department of Information Technology
Uppsala University
Box 337, SE – 751 05 Uppsala, Sweden
{pierreF,justin,agren}@it.uu.se

Carlos Garcia Avello, Mete Çeliktin, and Søren Dissing
Directorate of ATM Strategies, Air Traffic Services Division
European Organisation for the Safety of Air Navigation (Eurocontrol)
96 rue de la Fusée, BE – 1130 Brussels, Belgium
firstname.surname@eurocontrol.int

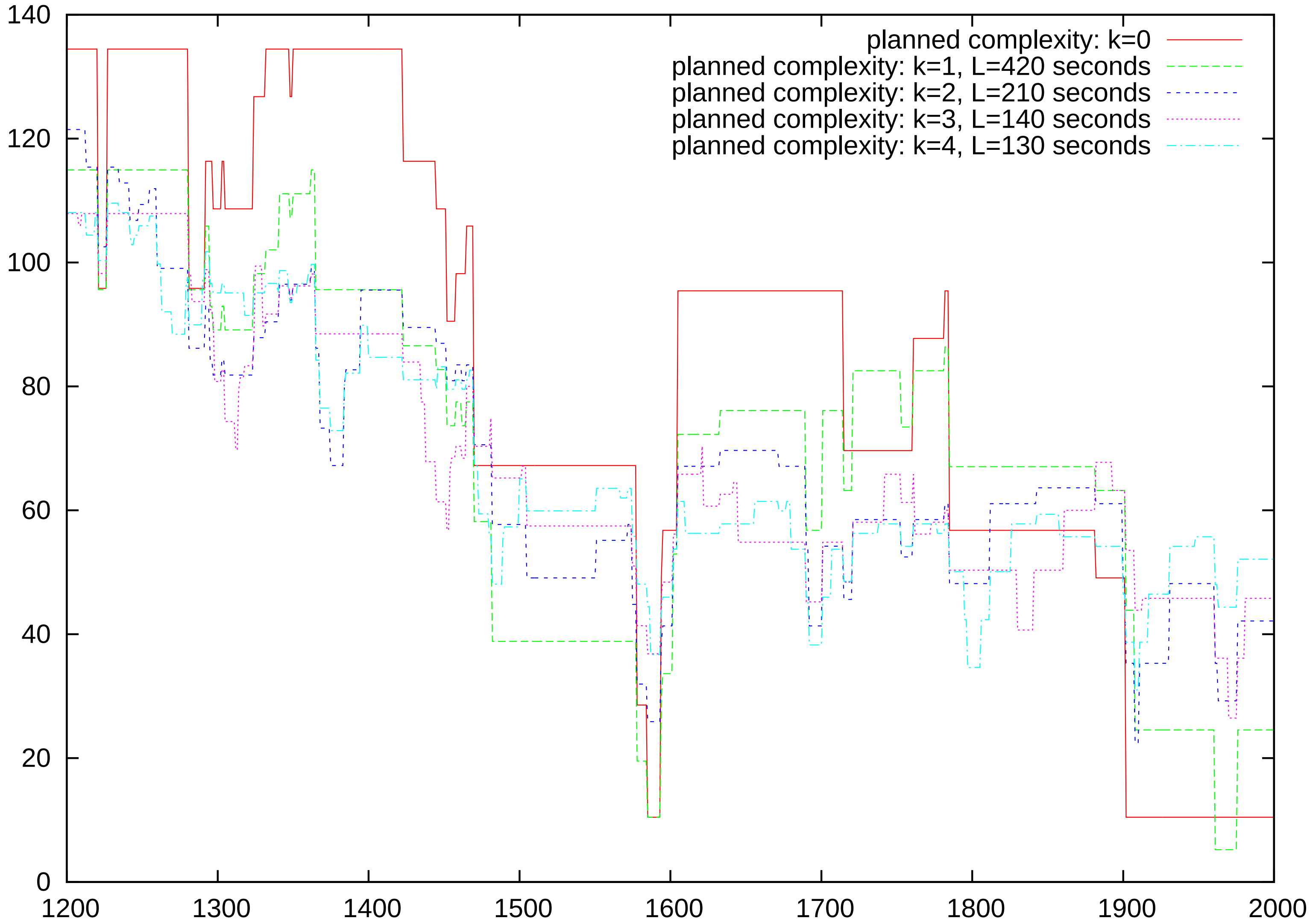
Moment Complexity

The complexity of a given sector s at a given moment m is based on the following terms:

- *Traffic volume*: Let N_{sec} be the number of flights in s at m.
- *Vertical state*: Let N_{cd} be the number of non-level (climbing or descending) flights in s at m.
- *Proximity to sector boundary*: Let N_{nsb} be the number of flights that are at most 15 nm horizontally or 40 FL vertically beyond their entry to s, or before their exit from s, at m.

The *moment complexity* of sector s at moment m is a normalised weighted sum of these terms:

$$C(s, m) = (a_{sec} \cdot N_{sec} + a_{cd} \cdot N_{cd} + a_{nsb} \cdot N_{nsb}) \cdot S_{norm}$$



Planned complexity after 11:10 on 23 June 2004 in sector *EBMALNL*

Interval Complexity

The *interval complexity* of a given sector s over a given time interval $[m, \ldots, m + k \cdot L]$ is the average of the moment complexities of s at the $k + 1$ sampled moments $m + i \cdot L$, for $0 \leq i \leq k$:

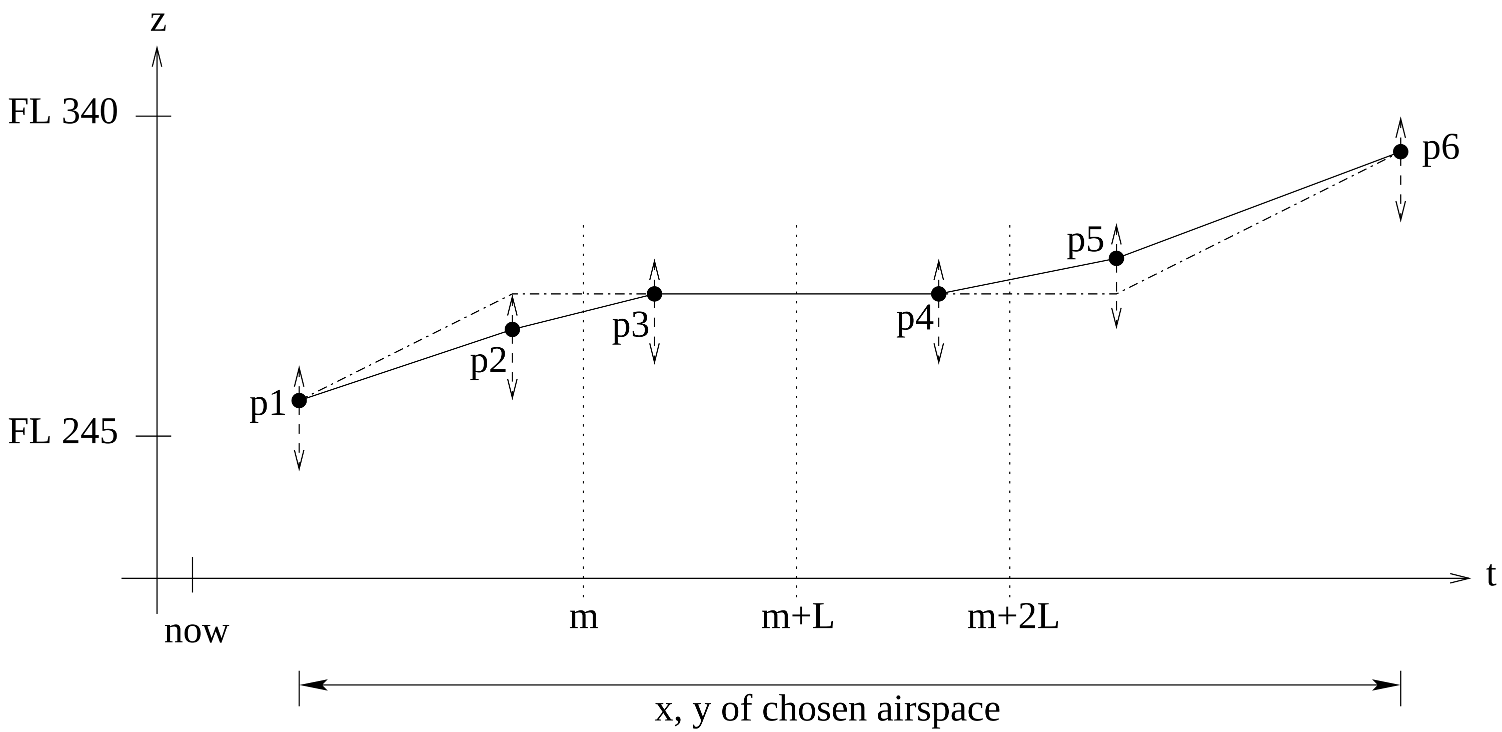
$$C(s, m, k, L) = \frac{\sum_{i=0}^{k} C(s, m + i \cdot L)}{k + 1}$$

where k is called the *smoothing degree*,
and L is the *time step* between the sampled moments.

Allowed Forms of Complexity Resolution

- Changing the take-off time of a not yet airborne flight by an integer amount of minutes, within the range $[-5, \ldots, 10]$.
- Changing the remaining approach time into the chosen airspace of an already airborne flight by an integer amount of minutes, but only within the two layers of feeder sectors around that airspace, at a speed-up (resp. slow-down) rate of maximum 1 (resp. 2) min per 20 min of approach time.
- Changing the altitude of passage over a way-point in the chosen airspace by an integer amount of FL (hundreds of feet), within the range $[-30, \ldots, 10]$, such that the flight climbs no more than 10 FL/min, or descends no more than 30 FL/min if it is a jet, and 10 FL/min if it is a turbo-prop.

Vertical Re-Profiling



Planned profile (plain line) and resolved profile (dot-dashed line) that minimises the number of climbing segments for the considered flight at the three sampled moments m, m+L, and m+2L

Pareto Optimisation

- Multi-objective minimisation problem: given a number of sectors $s_1, s_2, \ldots, s_n$, minimise the vector of their complexities with respect to a resolution R:

$$\langle C_R(s_1, m, k, L), \ldots, C_R(s_n, m, k, L) \rangle$$

- A vector of complexities is *Pareto optimal* if no element can be reduced without increasing some other element.
- Standard technique: Combine the multiple objectives into a single objective using a weighted sum:

$$\sum_{i=1}^{n} \alpha_i \cdot C_R(s_i, m, k, L)$$

for some weights $\alpha_i > 0$. In practice, one often takes $\alpha_i = 1$.

Some Parameters

- *lookahead* is a non-negative-integer amount of minutes; a typical value is a multiple of 10 in the range $[20, \ldots, 90]$.
- *now* is the time, given as an (hour, minute) pair, at which a resolved scenario is wanted with a forecast of *lookahead* minutes.
- Let $m = now + lookahead$ be the start moment of the time interval $[m, \ldots, m + k \cdot L]$ for complexity resolution.
- k is the smoothing degree; a good value is 2.
- L is the time step, in seconds; a good value is 200.
- *ff* is the minimum fraction of the number of flights planned to be in the chosen multi-sector airspace at the sampled moments $m + i \cdot L$ that have to be there in the resolved flight profile as well.

Some Decision Variables

- δT is a 1d array of integer variables for entry-time changes in $[-5, \ldots, 10]$, indexed by *Flights*.
- δH is a 1d array of integer variables for flight-level changes in $[-30, \ldots, 10]$, indexed by *FlightPoints*.
- b_{sec} is a 3d array, indexed by $Index = [0, \ldots, k]$, *OurSectors*, and *Flights*, of 0/1 variables, such that $b_{sec}[i, s, f] = 1$ if flight f is in sector s at moment $m + i \cdot L$ when its entry-time change is $\delta T[f]$.
- b_{cd} is a 3d array of 0/1 variables, such that $b_{cd}[i, s, f] = 1$ if flight f is on a non-level segment in sector s at moment $m + i \cdot L$ when its entry-time change is $\delta T[f]$ and the FL changes are as in δH.
- b_{nsb} is a 3d array of 0/1 variables: $b_{nsb}[i, s, f] = 1$ if f is near the boundary of s at $m + i \cdot L$ when its entry-time change is $\delta T[f]$.

Some Constraints

- All relevant flights planned to take off at or before *now* have taken off exactly according to their profile, but their approach times (within the feeder sectors) can be modified. All other relevant flights can necessarily only be changed to take off after *now*. Formally:

$$\forall f \in \mathit{Flights}\ .$$

$$\text{if } f.\mathit{timeTakeOff} \leq \mathit{now}$$

$$\text{then if } \mathit{now} < t_C[f]$$

$$\qquad \text{then } -\mathit{maxSpeedUp} \cdot \frac{a[f]}{20} \leq \delta T[f] \leq \mathit{maxSlowDown} \cdot \frac{a[f]}{20}$$

$$\qquad \text{else } \delta T[f] = 0$$

$$\text{else } \delta T[f] \in \Delta T \wedge f.\mathit{timeTakeOff} + \delta T[f] \cdot 60 > \mathit{now}$$

- Flight-point pairs flown over at or before *now* cannot have their flight levels changed:

$$\forall p \in \mathit{FlightPoints} : p.\mathit{timeOver} \leq \mathit{now} \ . \ \delta H[p] = 0$$

- Changed flight levels stay within the bounds of the sector, as flights cannot be re-routed through a lower or higher sector:

$$\forall s \in \mathit{OurSectors} \ . \ \forall f \in \mathit{SectorFlights}[s] \ . \ \forall p \in \mathit{Profile}[s,f] \ .$$
$$\mathit{Sector}[s].\mathit{bottomFL} \leq p.\mathit{level} + \delta H[p] \leq \mathit{Sector}[s].\mathit{topFL}$$

- Define the $b_{sec}[i,s,f]$ variables:

$$\begin{array}{l} \forall i \in \mathit{Index} \ . \ \forall s \in \mathit{OurSectors} \ . \ \forall f \in \mathit{Flights} \ . \\ \text{if } f \in \mathit{SectorFlights}[s] \\ \text{then } b_{sec}[i,s,f] = 1 \leftrightarrow \left(\begin{array}{c} \mathit{first}(\mathit{Profile}[s,f]).\mathit{timeOver} \leq m + i \cdot L - \delta T[f] \cdot 60 \\ < \mathit{last}(\mathit{Profile}[s,f]).\mathit{timeOver} \end{array} \right) \\ \text{else } b_{sec}[i,s,f] = 0 \end{array}$$

- Define the $b_{cd}[i,s,f]$ variables:

$$\forall i \in \mathit{Index}\ .\ \forall s \in \mathit{OurSectors}\ .\ \forall f \in \mathit{Flights}\ .$$
$$\text{if } f \in \mathit{SectorFlights}[s]$$
$$\text{then } b_{cd}[i,s,f] = 1 \leftrightarrow \left(\begin{array}{c} \exists p \in \mathit{Profile}[s,f] : p \neq \mathit{last}(\mathit{Profile}[s,f])\ . \\ p.\mathit{timeOver} \leq m + i \cdot L - \delta T[f] \cdot 60 < p'.\mathit{timeOver} \wedge \\ p.\mathit{level} + \delta H[p] \neq p'.\mathit{level} + \delta H[p'] \end{array} \right)$$
$$\text{else } b_{cd}[i,s,f] = 0$$

- Define the $b_{nsb}[i,s,f]$ variables:

$$\forall i \in \mathit{Index}\ .\ \forall s \in \mathit{OurSectors}\ .\ \forall f \in \mathit{Flights}\ .$$
$$\text{if } f \in \mathit{SectorFlights}[s]$$
$$\text{then } b_{nsb}[i,s,f] = 1 \leftrightarrow \left(\begin{array}{c} 0 \leq m + i \cdot L - (\mathit{first}(\mathit{Profile}[s,f]).\mathit{timeOver} + \delta T[f] \cdot 60) \leq hv_{nsb} \\ \wedge\ m + i \cdot L < \mathit{last}(\mathit{Profile}[s,f]).\mathit{timeOver} + \delta T[f] \cdot 60 \\ \vee \\ 0 < \mathit{last}(\mathit{Profile}[s,f]).\mathit{timeOver} + \delta T[f] \cdot 60 - (m + i \cdot L) \leq hv_{nsb} \\ \wedge\ \mathit{first}(\mathit{Profile}[s,f]).\mathit{timeOver} + \delta T[f] \cdot 60 \leq m + i \cdot L \end{array} \right)$$
$$\text{else } b_{nsb}[i,s,f] = 0$$

- No flight has to climb more than *maxUpJet* or *maxUpTurbo* levels per minute or descend more than *maxDownJet* or *maxDownTurbo* levels per minute:

$$\begin{array}{c}\forall s \in \mathit{OurSectors} \,.\, \forall f \in \mathit{SectorFlights}[s] \,.\\ \forall p \in \mathit{Profile}[s,f] : f.engineType = jet \wedge p \neq \mathit{last}(\mathit{Profile}[s,f]) \,.\\ -(p'.\mathit{timeOver} - p.\mathit{timeOver}) \cdot \mathit{maxDownJet} \leq ((p'.\mathit{level} + \delta H[p']) - (p.\mathit{level} + \delta H[p])) \cdot 60\\ \leq (p'.\mathit{timeOver} - p.\mathit{timeOver}) \cdot \mathit{maxUpJet}\\ \wedge\\ \forall s \in \mathit{OurSectors} \,.\, \forall f \in \mathit{SectorFlights}[s] \,.\\ \forall p \in \mathit{Profile}[s,f] : f.engineType = turbo \wedge p \neq \mathit{last}(\mathit{Profile}[s,f]) \,.\\ -(p'.\mathit{timeOver} - p.\mathit{timeOver}) \cdot \mathit{maxDownTurbo} \leq ((p'.\mathit{level} + \delta H[p']) - (p.\mathit{level} + \delta H[p])) \cdot 60\\ \leq (p'.\mathit{timeOver} - p.\mathit{timeOver}) \cdot \mathit{maxUpTurbo}\end{array}$$

- A climbing or descending flight in a sector is necessarily in that sector:

$$\forall i \in \mathit{Index} \,.\, \forall s \in \mathit{OurSectors} \,.\, \forall f \in \mathit{SectorFlights}[s] \,.\, b_{cd}[i,s,f] = 1 \rightarrow b_{sec}[i,s,f] = 1$$

- A flight near the boundary of a sector is necessarily in that sector:

$$\forall i \in \mathit{Index} \,.\, \forall s \in \mathit{OurSectors} \,.\, \forall f \in \mathit{SectorFlights}[s] \,.\, b_{nsb}[i,s,f] = 1 \rightarrow b_{sec}[i,s,f] = 1$$

- A flight in a sector cannot simultaneously be in any other sector:

$$\forall i \in Index\ .\ \forall f \in Flights\ .\ \sum_{s \in OurSectors} b_{sec}[i, s, f] \leq 1$$

- A fraction of minimum *ff* of the sum N of the numbers of flights that are planned to be in one of the chosen sectors at the sampled moments $m + i \cdot L$ must remain in one of the chosen sectors:

$$\sum_{i \in Index} \sum_{s \in OurSectors} N_{sec}[i, s] \geq \lceil N \cdot \mathit{ff} \rceil$$

The Search Procedure and Heuristics

1. Label the 0/1 variables $b_{sec}[i, s, f]$, $b_{cd}[i, s, f]$, and $b_{nsb}[i, s, f]$: Priority goes to placing a flight f within a sector s at time $m + i \cdot L$, but neither on a non-level segment nor near the boundary of s. Consider the sectors s by decreasing size of $\mathit{SectorFlights}[s]$, so as to begin with the sectors that are planned to be the most busy.

2. Label the δT variables, using a value ordering by increasing absolute value within $[-10, \ldots, 5]$.

3. Label the δH variables, using a value ordering by increasing absolute value within $[-30, \ldots, 10]$.

This guarantees revised flight profiles that deviate as little as possible from the original ones.

Implementation

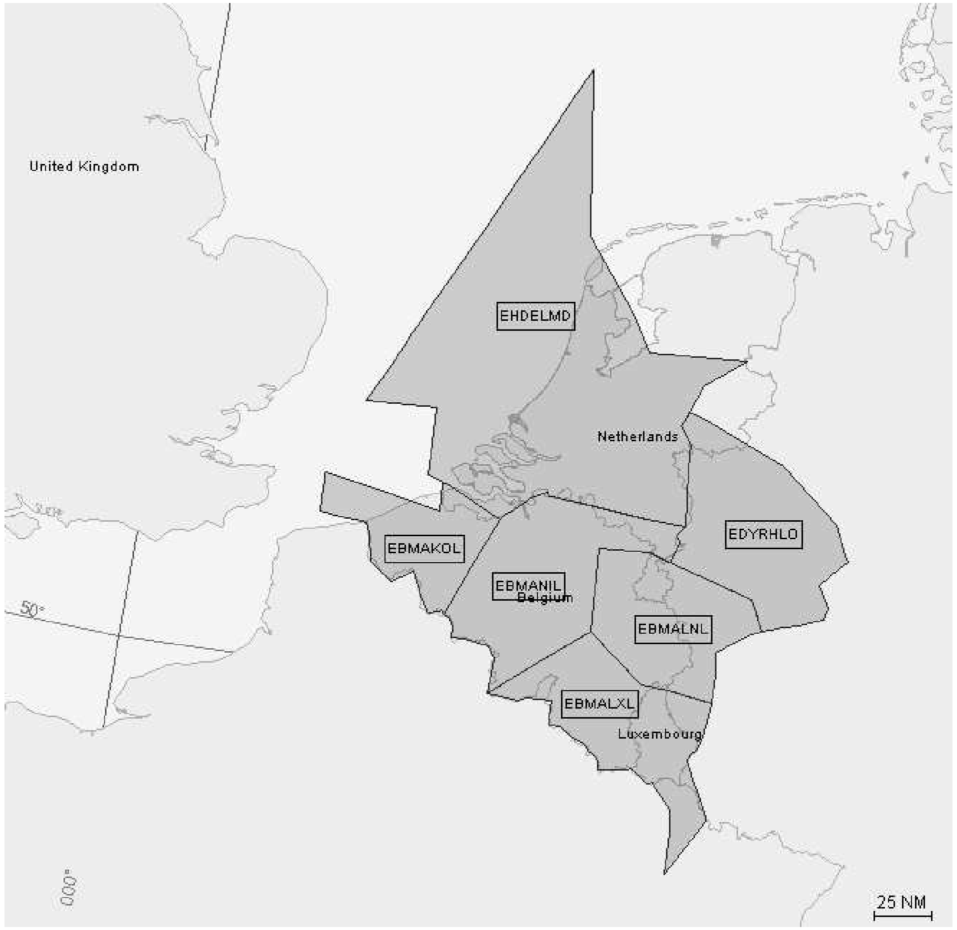
- The constraint program was implemented in OPL 3.7.
- Translating the designed constraint program into an OPL model is merely a matter of slight syntax changes, despite the absence of an existential quantifier in OPL.
- The resulting OPL model has non-linear higher-order constraints and non-linear channelling constraints, hence the OPL compiler translates the model into code for ILOG Solver, rather than for CPLEX, and constraint solving takes place at runtime.

Experimental Setup

- The chosen air-traffic control centre is Maastricht (Netherlands).
- The chosen multi-sector airspace within the Maastricht airspace consists of five high-density, en-route, upper airspace sectors:

sectorId	*bottomFL*	*topFL*	a_{sec}	a_{cd}	a_{nsb}	S_{norm}
EBMALNL	245	340	7.74	15.20	5.69	1.35
EBMALXL	245	340	5.78	5.71	15.84	1.50
EBMAWSL	245	340	6.00	7.91	10.88	1.33
EDYRHLO	245	340	12.07	6.43	9.69	1.00
EHDELMD	245	340	4.42	10.59	14.72	1.11

- There are an additional 34 feeder sectors.



Chosen multi-sector airspace: on the chosen day, the sectors EBMAKOL and EBMANIL were collapsed into EBMAWSL

- The chosen day is 23 June 2004.
- The chosen hours are the peak traffic hours, that is from 07:00 to 22:00 local time.
- The chosen flights follow standard routes (no free flight) and are of the turbo-prop or jet type.

The Central Flow Management Unit (CFMU) provided us with the $1,798$ flight profiles for these choices.

Results

lookahead	k	L	Average planned	Average resolved
20	2	210	87.92	47.69
20	3	180	86.55	50.17
45	2	210	87.20	45.27
45	3	180	85.67	47.81
90	2	210	87.29	44.67
90	3	180	85.64	47.13

Average planned and resolved complexities in the chosen airspace, with $ff = 90\%$ of the flights kept in the chosen airspace, and $timeOut = 120$ seconds on an Intel Pentium 4 CPU with 2.53GHz, a 512 KB cache, and a 1 GB memory

Conclusion

- Success story for constraint programming.
- First usage of a *more sophisticated notion of the air-traffic complexity of a sector* than just its number of flights.
- First attempt at *complexity resolution*, in *multi*-sector planning.
- Strategic use of the model, rather than actual deployment: new definitions of air-traffic complexity can readily be experimented with, and constraints can readily be changed or added.
- In practice, air-traffic complexity resolution will not be a COP, as here, but rather a CSP, simply requiring the resolved air-traffic complexities to be within prescribed intervals.
- No worries about potentially interacting pairs of flights, as their number has a very low correlation with the air-traffic complexity.

Future Work

- Constraints to avoid unacceptable complexity being generated before the start moment m of the time interval $[m, \ldots, m + k \cdot L]$.
- Constraints on sufficiently fast implementability of resolved flight profiles, and that implementing them is still offset by the resulting complexity reductions and redistribution among sectors. For instance, the number of flights affected by the changes may have to be kept under a given threshold.
- Horizontal re-profiling, among static/dynamic list of other routes.
- Cost minimisation: of ground/air holding, . . .
- A notion of airline equity towards a collaborative decision making process between Eurocontrol and the airlines.

Acknowledgements

- This research project has been funded by Eurocontrol grant C/1.246/HQ/JC/04 and its amendments 1/04 and 2/05.
- Many thanks to Bernard Delmée, Jacques Lemaître, and Patrick Tasker at EuroControl DAP/DIA, for pre-processing the CFMU raw data into the extended data and format we needed.