Air Traffic Complexity Resolution in Multi-Sector Planning Using CP

Pierre Flener[1] Justin Pearson[1] Magnus Ågren[1]
Carlos Garcia-Avello[2] Mete Çeliktin[2] Søren Dissing[2]

[1]Department of Information Technology, Uppsala University, Sweden
Firstname.Surname@it.uu.se
http://www.it.uu.se/research/group/astra/

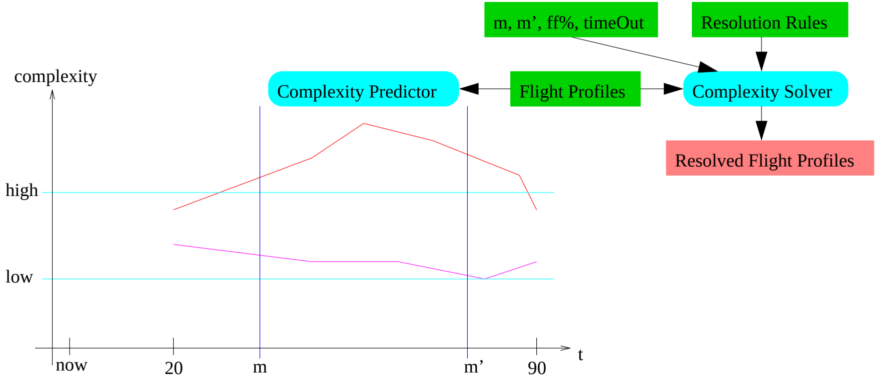
[2]EuroControl, Brussels, Belgium
Firstname.Surname@eurocontrol.int

7th USA / Europe R&D Seminar on Air Traffic Management

Outline

Outline

Target Scenario



Contributions

- Traffic complexity $\neq$ # flights
- Complexity resolution . . .
- . . . in multi-sector planning
- Use of constraint programming (CP)

Outline

Complexity Parameters

The complexity of sector *s* at moment *m* depends here on:

- N_{sec} = # flights in *s* at *m* (traffic volume)
- N_{cd} = # flights in *s* that are non-level at *m* (vertical state)
- N_{nsb} = # flights that are
  - at most 15 nm horizontally, or 40 FL vertically
  - beyond their entry into *s*, or before their exit from *s*

  at *m* (proximity to sector boundary)

Moment Complexity

The moment complexity of sector s at moment m is defined by:

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

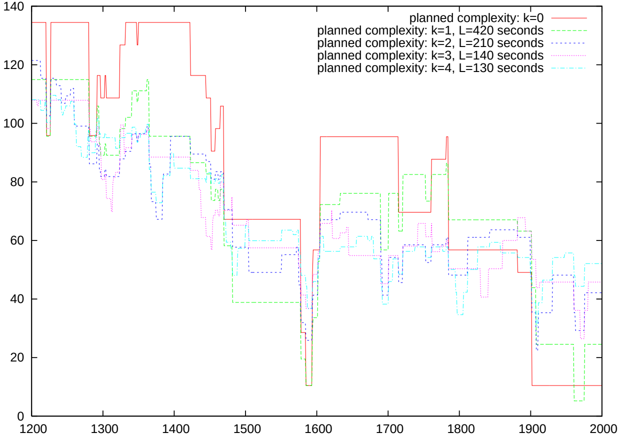
where:

- w_{sec}, w_{cd}, and w_{nsb} are experimentally determined weights
- S_{norm} characterises the structure, equipment used, procedures followed, etc, of s (sector normalisation)

Unused Complexity Parameters

- Data-link equipage, time adjustment, temporary restriction: no data to quantify the w_{sec}, w_{cd}, and w_{nsb} weights.
- Potentially interacting pairs: (surprisingly) weak correlation with the COCA complexity; because traffic volume and vertical state already capture this impact?
- Aircraft type diversity: weak correlation with the COCA complexity; because of the limited amount of data used in the determination of the w_{sec}, w_{cd}, and w_{nsb} weights?

Large Variance of Moment Complexity



Example:
Complexity after 11:10 on 23/6/2004 in EBMALNL

Interval Complexity

The interval complexity of sector s over interval $[m, \ldots, m']$ is the average of its moment complexities at sampled moments:

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

where:

- k = smoothing degree
- L = time step between the sampled moments
- $m' = m + k \cdot L$

In practice, for complexity resolution: $k = 2$ and $L \approx 210$ sec

Outline

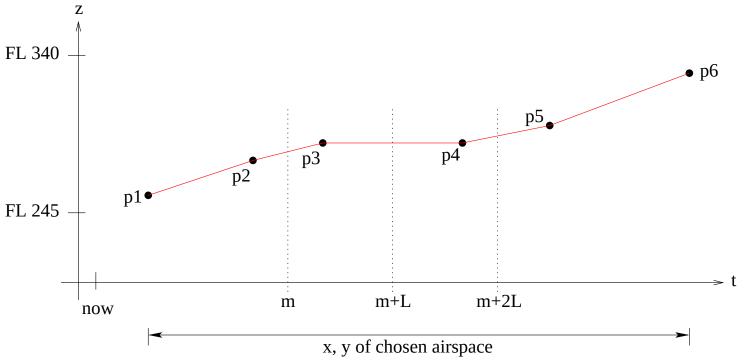
Allowed Forms of Complexity Resolution I

Temporal Re-Profiling:

Change the **entry time** of the flight into the chosen airspace:

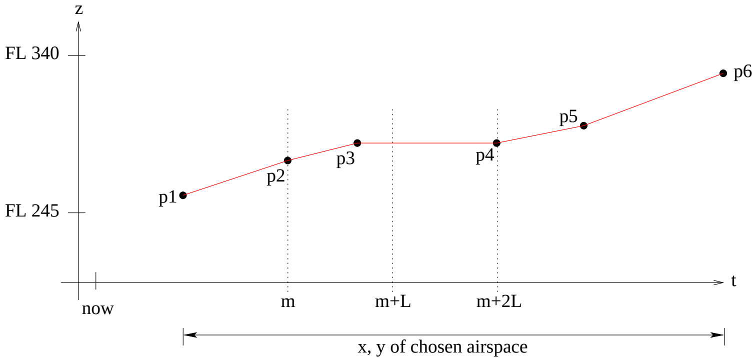
- Grounded: Change the **take-off time** of a not yet airborne flight by an integer amount of minutes within $[-5, \ldots, +10]$
- Airborne: Change the **remaining approach time** into the chosen airspace of an already airborne flight by an integer amount of minutes, but only within the two layers of feeder sectors around the chosen airspace:
  - at a speed-up rate of maximum 1 min per 20 min of flight
  - at a slow-down rate of maximum 2 min per 20 min of flight

Example: Temporal Re-Profiling



Planned profile

Example: Temporal Re-Profiling



Resolved profile

Allowed Forms of Complexity Resolution II

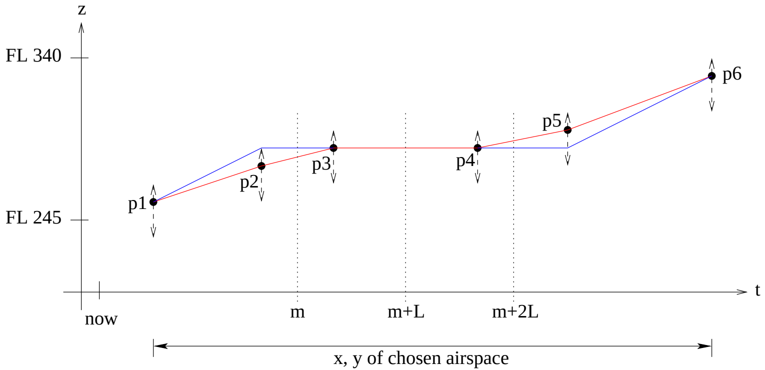
Vertical Re-Profiling:

- Change the **altitude** of passage over a way-point in the chosen airspace by an integer amount of FLs (hundreds of feet), within $[-30, \dots, +10]$, so that the flight
  - climbs no more than 10 FL / min
  - descends no more than 30 FL / min if it is a jet
  - descends no more than 10 FL / min if it is a turbo-prop

Horizontal Re-Profiling:

- Future work?

Example: Vertical Re-Profiling



Planned profile and resolved profile that minimises the number of climbing segments for the considered flight at the sampled moments m, m+L, and m+2L

Assumptions

- Proximity to a sector boundary is approximatable by being at most $hv_{nsb} = 120$ sec of flight beyond the entry to, or before the exit from, the considered sector.
  This approximation only holds for en-route airspace.
- Times can be controlled with an accuracy of one minute: the profiles are just *shifted* in time.
- Flight time along a segment does not change if we restrict the FL changes over its endpoints to be “small”.
  Otherwise, many more time variables will be needed, leading to combinatorial explosion.

Constraint Programming (CP)

New technology for modelling & solving constraint problems:

- Origins: Computer science, AI, computational logic, . . .
- Modelling: Encapsulate solving algorithms in *constraints* capturing common combinatorial structures of problems. **Example:** In a Sudoku puzzle, there are *allDifferent* constraints on each row, column, and 3 by 3 block.
- Solving: Iteratively pick a value for a variable, propagate this choice, and backtrack when necessary; use domain knowledge to guide search with heuristics so that exponential run-time behaviour is a rarer occurrence. **Example:** Just like we humans solve Sudoku puzzles!
- Explaining why a particular solution, or none, was found.

Some Decision Variables

- $\delta T[f]$ = entry-time change in $[-5, \ldots, +10]$ of flight f
- $\delta H[p]$ = level change in $[-30, \ldots, +10]$ of flight-point p
- $N_{sec}[i, s]$ = # flights in sector s at sampled moment $m + i \cdot L$
- $N_{cd}[i, s]$ = # flights on a non-level segment in s at $m + i \cdot L$
- $N_{nsb}[i, s]$ = # flights near the boundary of s at $m + i \cdot L$

Some Constraints I

- All flights planned to take off until *now* have taken off exactly according to their profile.
- All other flights take off after *now*.
- Points flown over until *now* cannot have their FLs changed:

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

- Changed FLs stay within the bounds of the sector, as (currently) no re-routing through a lower or higher sector:

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

Some Constraints II

- Define the $N_{sec}[i, s]$ decision variables:

$$\forall i \in [0, \ldots, k] \,.\, \forall s \in OurSectors \,.$$
$$N_{sec}[i,s] = \left| \left\{ f \in Flights[s] \;\middle|\; \begin{array}{c} first(Profile[s,f]).timeOver \leq m + i \cdot L - \delta T[f] \\ < last(Profile[s,f]).timeOver \end{array} \right\} \right|$$

- Define the $N_{cd}[i, s]$ decision variables:

$$\forall i \in [0, \ldots, k] \,.\, \forall s \in OurSectors \,.$$
$$N_{cd}[i,s] = \left| \left\{ f \in Flights[s] \;\middle|\; \begin{array}{c} \exists p \in Profile[s,f] : p \neq last(Profile[s,f]) \,. \\ p.timeOver \leq m + i \cdot L - \delta T[f] < p'.timeOver \wedge \\ p.level + \delta H[p] \neq p'.level + \delta H[p'] \end{array} \right\} \right|$$

- Define the $N_{nsb}[i, s]$ decision variables:

$$\forall i \in [0, \ldots, k] \,.\, \forall s \in OurSectors \,.$$
$$N_{nsb}[i,s] = \left| \left\{ f \in Flights[s] \;\middle|\; \begin{array}{c} 0 \leq m + i \cdot L - (first(Profile[s,f]).timeOver + \delta T[f]) \leq hv_{nsb} \\ \wedge\, m + i \cdot L < last(Profile[s,f]).timeOver + \delta T[f] \\ \vee \\ 0 < last(Profile[s,f]).timeOver + \delta T[f] - (m + i \cdot L) \leq hv_{nsb} \\ \wedge\, first(Profile[s,f]).timeOver + \delta T[f] \leq m + i \cdot L \end{array} \right\} \right|$$

Some Constraints III

- No climbing $> maxUpJet = 10 = maxUpTurbo$ FL / min, no descending $> maxDownJet = 30$ FL / min, no descending $> maxDownTurbo = 10$ FL / min:

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

Some Constraints IV

- Minimum *ff* of the sum *N* of the numbers of flights 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:

$$\sum_{i \in [0,\ldots,k]} \sum_{s \in OurSectors} N_{sec}[i,s] \geq \lceil ff \cdot N \rceil$$

- Define the $MC[i,s]$ moment complexities:

$$\forall i \in [0,\ldots,k] \,.\, \forall s \in OurSectors \,.$$
$$MC[i,s] = (w_{sec}[s] \cdot N_{sec}[i,s] + w_{cd}[s] \cdot N_{cd}[i,s] + w_{nsb}[s] \cdot N_{nsb}[i,s]) \cdot S_{norm}[s]$$

- Define the $IC[s]$ interval complexities:

$$\forall s \in OurSectors \,.\, IC[s] = \frac{\sum_{i \in [0,\ldots,k]} MC[i,s]}{k+1}$$

The Objective Function

- Multi-objective optimisation problem: minimise the vector $\langle IC[s_1], \ldots, IC[s_n] \rangle$ of the interval complexities of n sectors.
- A vector of values 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 $\sum_{j=1}^{n} \alpha_j \cdot IC[s_j]$ for some weights $\alpha_j > 0$.
- In practice, and as often done, we take $\alpha_j = 1$:

$$\text{minimise} \sum_{s \in OurSectors} IC[s]$$

The Search Procedure and Heuristics

1. Assign the $N_{sec}[i,s]$, $N_{cd}[i,s]$, and $N_{nsb}[i,s]$ variables:
   Try placing a flight within s at the i^{th} sampled moment, but neither on a non-level segment nor near the boundary of s. Begin with the sectors planned to be the busiest.
2. Assign the $\delta T[f]$ variables.
   Try by increasing absolute values in $[-10,\ldots,+5]$.
3. Assign the $\delta H[p]$ variables.
   Try by increasing absolute values in $[-30,\ldots,+10]$.

The given orderings guarantee resolved flight profiles that deviate as little as possible from the planned ones.

Implementation

- The constraints were implemented in the Optimisation Programming Language (OPL), marketed by ILOG SA.
- Merely a matter of slight syntax changes!
- The resulting OPL model has non-linear and higher-order constraints, hence the OPL compiler translates the model into code for ILOG Solver, rather than for ILOG CPLEX, and constraint processing takes place at runtime.

Outline

Experimental Setup I

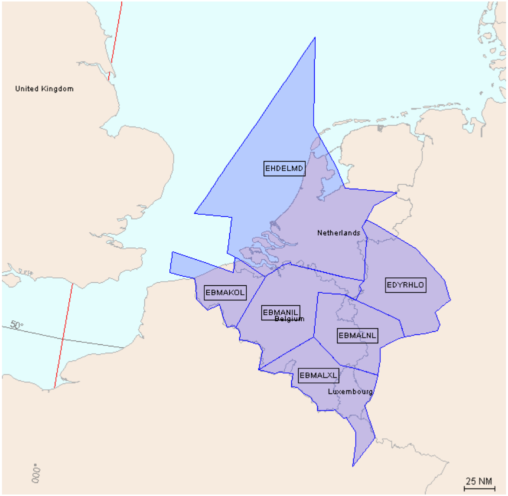
- ATC centre = Maastricht, the Netherlands
- Multi-sector airspace =
  five high-density, en-route, upper-airspace sectors:

sectorId	*bottomFL*	*topFL*	w_{sec}	w_{cd}	w_{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

- Time = peak traffic hours, from 7 to 22, on 23/6/2004
- Flights = turbo-props and jets, on standard routes

Central Flow Management Unit (CFMU): 1,798 flight profiles

Experimental Setup II



Chosen multi-sector airspace, surrounded by an additional 34 feeder sectors (on the chosen day, the sectors EBMAKOL and EBMANIL were collapsed into EBMAWSL)

Results

Significant complexity reductions and re-balancing:

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 chosen airspace, with $ff = 90\%$ of the flights kept in the chosen airspace, and $timeOut = 120$ seconds

Outline

Contributions

- Traffic complexity $\neq$ # flights
- Complexity resolution . . .
- . . . in multi-sector planning
- Use of constraint programming (CP)

Future Work

- Strategic use of the model, rather than actual deployment: new definitions of complexity can readily be experimented with, and constraints can readily be changed or added.
- In practice, complexity resolution is not an optimisation problem, but a satisfaction problem:
  Constraints on *interval* for resolved complexities.
- Constraints on *fast* implementability of resolved profiles.
  Example: Keep # affected flights under a given threshold.
- Horizontal re-profiling: among static / dynamic list of routes
- Cost minimisation: of ground / air holding, . . .
- Airline equity: towards a collaborative decision making process between EuroControl and the airlines.

Acknowledgements

- This research project was 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 we needed.

UPPSALA UNIVERSITET

EUROCONTROL

Sabancı Üniversitesi

Bibliography I

Baptiste, Philippe; Le Pape, Claude; and Nuijten, Wim.
Constraint-Based Scheduling. Kluwer, 2001.

Darby-Dowman, Ken and Little, James.
Properties of some combinatorial optimization problems and their effect on the performance of integer programming and constraint logic programming.
INFORMS Journal on Computing, 10(3):276–286, 1998.

EuroControl, Directorate of ATM Strategies, Air Traffic Services division. *Complexity algorithm development:*
- *Literature survey & parameter identification*. 9 Feb 2004.
- *The algorithm*. Edition 1.0, 7 April 2004.
- *Validation exercise*. Edition 0.3, 10 September 2004.

Bibliography II

- EuroControl Experimental Centre (EEC). *Pessimistic sector capacity estimation*. EEC Note Number 21/03, 2003.
- Flener, P.; Pearson, J.; Ågren, M.; Garcia Avello, C.; and Çeliktin, M. Technical report at `http://www.it.uu.se/research/publications/reports/2007-003/`.
- Milano, M. (editor). *Constraint and Integer Programming: Toward a Unified Methodology*. Kluwer, 2004.
- Rossi, Francesca; van Beek, Peter; and Walsh, Toby (eds). *Handbook of Constraint Programming*. Elsevier, 2006.
- Van Hentenryck, P. Constraint and integer programming in OPL. *INFORMS J. on Computing*, 14(4):345–372, 2002.