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]ASTRA Research Group
Department of Information Technology
Uppsala University, Sweden
Firstname.Lastname@it.uu.se
www.it.uu.se/research/group/astra

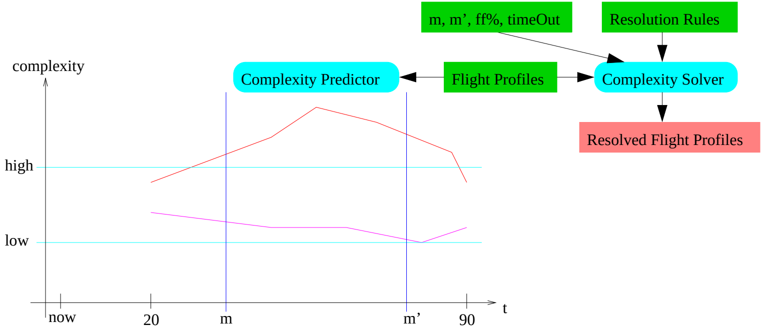
[2]EuroControl, Brussels, Belgium
Firstname.Lastname@eurocontrol.int

Int'l Workshop on Constraint Programming
for ATC & ATM, Brétigny (France), 2 December 2008

Outline

Outline

Target Scenario



Contributions

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

UPPSALA UNIVERSITET

Contributions

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

Contributions

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

Contributions

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

Outline

UPPSALA UNIVERSITET

Objective

Air Traffic Complexity

Complexity Resolution

A CP Model

Experiments

Conclusion

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

  at moment *m* (proximity to sector boundary)

NB: The complexity of sector *s* at moment *m* does **not** depend here on potentially interacting pairs of aircraft: surprisingly weak correlation with the COCA complexity; do traffic volume & vertical state already capture this impact?

UPPSALA UNIVERSITET

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

  at moment m (proximity to sector boundary)

NB: The complexity of sector s at moment m does **not** depend here on potentially interacting pairs of aircraft: surprisingly weak correlation with the COCA complexity; do traffic volume & vertical state already capture this impact?

Moment Complexity

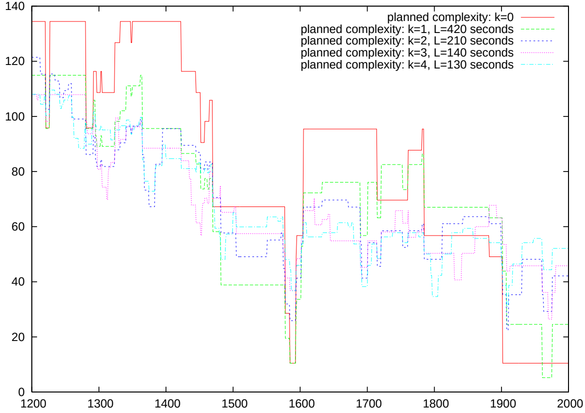
The moment complexity of sector s at moment m is here:

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

where:

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

Large Variance of Moment Complexity



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

UPPSALA UNIVERSITET

Interval Complexity

The interval complexity of sector s over interval $[m, \ldots, m']$ is the average of its moment complexities at the $k + 1$ sampled moments m, $m + L$, $m + 2L$, ..., $m + k \cdot L = m'$:

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

where:

- k = smoothing degree
- L = time step between the sampled moments

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

NB: This definition of complexity can be changed **without** compromising the whole work!

UPPSALA UNIVERSITET

Objective

Air Traffic Complexity

Complexity Resolution

A CP Model

Experiments

Conclusion

Interval Complexity

The interval complexity of sector s over interval $[m, \ldots, m']$ is the average of its moment complexities at the $k+1$ sampled moments m, $m+L$, $m+2L$, ..., $m+k \cdot L = m'$:

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

where:

- k = smoothing degree
- L = time step between the sampled moments

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

NB: This definition of complexity can be changed **without** compromising the whole work!

Outline

UPPSALA UNIVERSITET

Allowed Forms of Complexity Resolution I

Objective
Air Traffic Complexity
Complexity Resolution
A CP Model
Experiments
Conclusion

Temporal Re-Profiling:

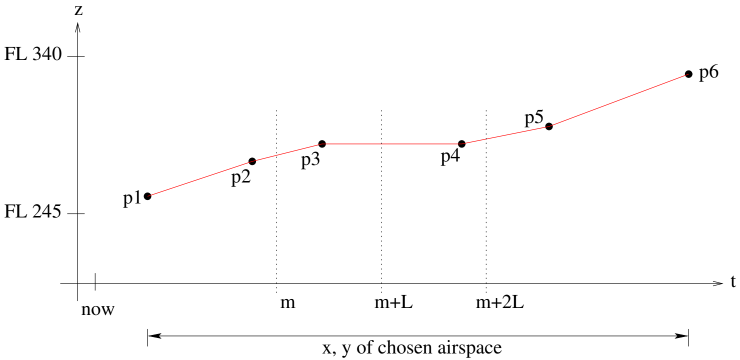
Change the entry time of a flight into the chosen airspace:

- 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 5%
  - at a slow-down rate of maximum 10%

UPPSALA UNIVERSITET

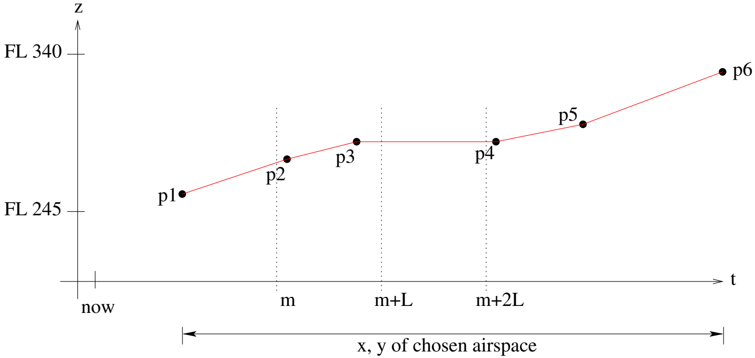
- Objective
- Air Traffic Complexity
- **Complexity Resolution**
- A CP Model
- Experiments
- Conclusion

Example: Temporal Re-Profiling



Planned profile

Example: Temporal Re-Profiling



Resolved profile

UPPSALA UNIVERSITET

Allowed Forms of Complexity Resolution II

Vertical Re-Profiling:

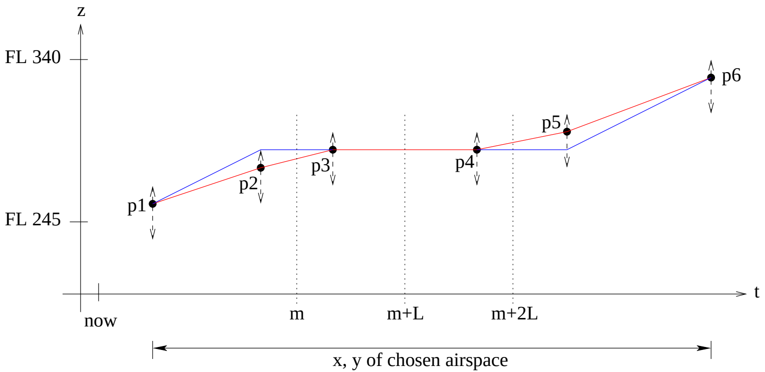
- Change the altitude of passage over a way-point in the chosen airspace by an integer amount of FLs within $[-30, \ldots, +10]$, so that the flight
  - climbs no more than 10 FL / min
  - descends no more than 30 FL / min if it is a jet
  - descends no more than 10 FL / min if it is a turbo-prop

2D Re-Profiling:

- Future work?

UPPSALA UNIVERSITET

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 1 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.

Outline

Some Parameters

- *now* is the time at which a resolved scenario is wanted with a forecast of *lookahead* minutes
- *lookahead* is typically a multiple of 10 in $[20, \ldots, 90]$
- $m = now + lookahead$ is the start moment of the time interval $[m, \ldots, m + k \cdot L]$ for complexity resolution
- *ff* = minimum fraction of flights planned to be in chosen airspace that must stay there at the sampled moments
- *timeOut* = amount of CPU seconds after which the currently best feasible solution is to be returned

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 i
- $N_{cd}\ [i, s]$ = # flights on a non-level segment in s at i
- $N_{nsb}[i, s]$ = # flights near the boundary of s at i

Some Constraints

- 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 get changed FLs:

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

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

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

Some Constraints

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

UPPSALA UNIVERSITET

Some Constraints

- 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 get changed FLs:

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

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

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

Some Constraints

- 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 get changed FLs:

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

- Changed FLs stay within the bounds of the sector, as (yet) 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

- 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 get changed FLs:

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

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

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

Some Constraints

- 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 get changed FLs:

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

- Changed FLs stay within the bounds of the sector, as (yet) 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 (cont'd)

- 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:

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

Some Constraints (cont'd)

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

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

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

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

Some Constraints (cont'd)

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

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

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

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

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

Objective
Air Traffic Complexity
Complexity Resolution
A CP Model
Experiments
Conclusion

Some Constraints (cont'd)

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

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

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

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

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

Some Constraints (cont'd)

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

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

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

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

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

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

Some Constraints (cont'd)

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

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

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

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

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

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

Some Constraints (cont'd)

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

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

Some Constraints (cont'd)

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

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

Some Constraints (end)

- Minimum fraction *ff* of the number of flights planned to be in the chosen airspace at the sampled moments *i* must remain then in that chosen airspace:

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

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

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

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

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

Some Constraints (end)

- Minimum fraction *ff* of the number of flights planned to be in the chosen airspace at the sampled moments *i* must remain then in that chosen airspace:

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

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

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

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

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

Some Constraints (end)

- Objective
- Air Traffic Complexity
- Complexity Resolution
- **A CP Model**
- Experiments
- Conclusion

- Minimum fraction *ff* of the number of flights planned to be in the chosen airspace at the sampled moments *i* must remain then in that chosen airspace:

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

- 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:

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

Some Constraints (end)

- Minimum fraction ff of the number of flights planned to be in the chosen airspace at the sampled moments i must remain then in that chosen airspace:

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

- 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:

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

UPPSALA UNIVERSITET

Some Constraints (end)

- Minimum fraction ff of the number of flights planned to be in the chosen airspace at the sampled moments i must remain then in that chosen airspace:

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

- 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:

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

Some Constraints (end)

Objective
Air Traffic Complexity
Complexity Resolution
A CP Model
Experiments
Conclusion

- Minimum fraction *ff* of the number of flights planned to be in the chosen airspace at the sampled moments *i* must remain then in that chosen airspace:

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

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

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

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

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

UPPSALA UNIVERSITET

The Objective Function

- We have a multi-objective optimisation problem: minimise the vector $\langle IC[s_1], \ldots, IC[s_n] \rangle$ of the interval complexities of n sectors s_i.
- A vector of values is Pareto minimal 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 $\alpha_j = 1$ for all j:

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

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 sampled moment i, 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 $[-30, \ldots, +10]$.

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

UPPSALA UNIVERSITET

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 sampled moment i, 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 $[-30,\ldots,+10]$.

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

UPPSALA UNIVERSITET

Implementation

The constraints were implemented in the Optimization Programming Language (OPL), marketed by ILOG. This is merely a matter of slight syntax changes! Prejudice:

> *The contribution of the article should be the reduction of an engineering problem to a known optimization format. [. . .] showcases pseudo code [. . .] submit this work to a journal interested in code semantics [. . .].*
> — Reviewer of this paper at a prestigious OR journal

The resulting OPL model has non-linear and higher-order constraints, hence the OPL compiler translates the model into code for ILOG Solver (now ILOG CP Optimizer), rather than for ILOG CPLEX, and constraint propagation takes place at runtime.

Outline

Experimental Setup I

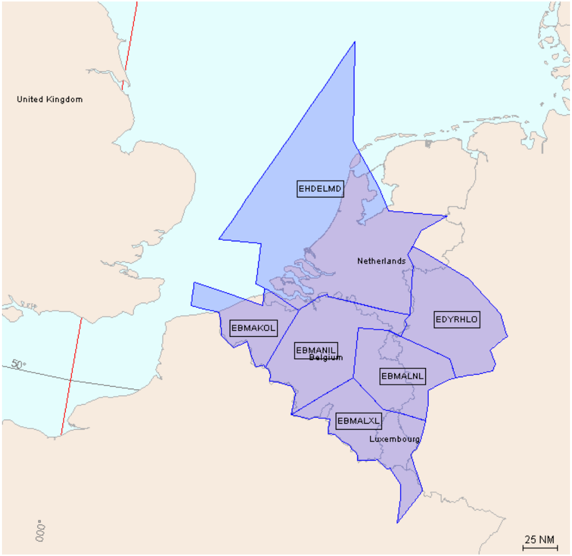
- ATC centre = Maastricht, in the Netherlands
- Multi-sector airspace =
  five high-density, en-route, upper-airspace sectors:

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 flights

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)

UPPSALA UNIVERSITET

- Objective
- Air Traffic Complexity
- Complexity Resolution
- A CP Model
- **Experiments**
- Conclusion

Results

Significant complexity reductions and re-balancing, obtained quickly (though with long proofs of optimality):

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

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

Outline

UPPSALA UNIVERSITET

Summary

Reduction: Complexity can be reduced by combination of:

- Reprofiling flights into less complex sectors
- Reprofiling flights away from sector boundaries
- Reprofiling flights onto level segments

Non-Zero Sum:

- Take-off and speed resolutions do **not** just transfer complexity to adjacent multi-sectors, because a parameter controls the percentage of flights that are to be kept within the considered multi-sector.
- Level and speed resolutions can reduce the complexity of a sector **without** increasing it elsewhere.

Rebalancing: Current flight profiles often yield huge complexity discrepancies among sectors, but complexity resolution also addresses this.

Contributions

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

Future Work

- Strategic use of the model, rather than deployment: new definitions of complexity can readily be tried, and constraints can readily be changed or added.
- In practice, complexity resolution is **not** an optimisation problem, but a satisfaction problem:
  need constraints on *interval* for resolved complexities.
- Constraints on *fast* executability of resolved profiles.
  Example: Keep # affected flights under threshold.
- Horizontal re-profiling: among static / dynamic route list
- 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

Bibliography: Background

- EuroControl, Directorate of ATM Strategies, Air Traffic Services division. *Complexity algorithm development:*
  - *Literature survey & parameter identification*. Feb 2004.
  - *The algorithm*. April 2004.
  - *Validation exercise*. September 2004.
- EuroControl Experimental Centre (EEC). *Pessimistic sector capacity estimation*. EEC Note 21/03, 2003.
- Flener, P.; Pearson, J.; Ågren, M.; Garcia Avello, C.; Çeliktin, M; and Dissing, S. Air-Traffic Complexity Resolution in Multi-Sector Planning. *J. of Air Transport Management*, 13(6):323–328, Nov. 2007. Technical report at `www.it.uu.se/research/reports/2007-003`.
- Van Hentenryck, P. *The OPL Optimization Programming Language*. The MIT Press, 1999.

Bibliography: Related Work

- Bertsimas, D. and Stock Patterson, S.
The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach.
Transportation Science, 34(3):239–255, 2000.
- Dalichampt, M.; Petit, E.; Junker, U.; and Lebreton, J..
Innovative slot allocation.
EEC Report 322, Brétigny, France, 1997.
- Sherali, H. D.; Staats, R. W.; and Trani, A. A.
An airspace planning and collaborative decision-making model: Part I — Probabilistic conflicts, workload, and equity considerations.
Transportation Science, 37(4):434–456, 2003.