# Air-traffic Flow Management with ILOG CP Optimizer

Ulrich Junker
ILOG

# Outline

- Eurocontrol's Air Traffic Flow Management Problem
- How to develop a precise and accurate optimization model?
- How to find good and precise solutions quickly?
- Experimental results with ILOG CP Optimizer

# Air-traffic Flow Management

(as of 1997)

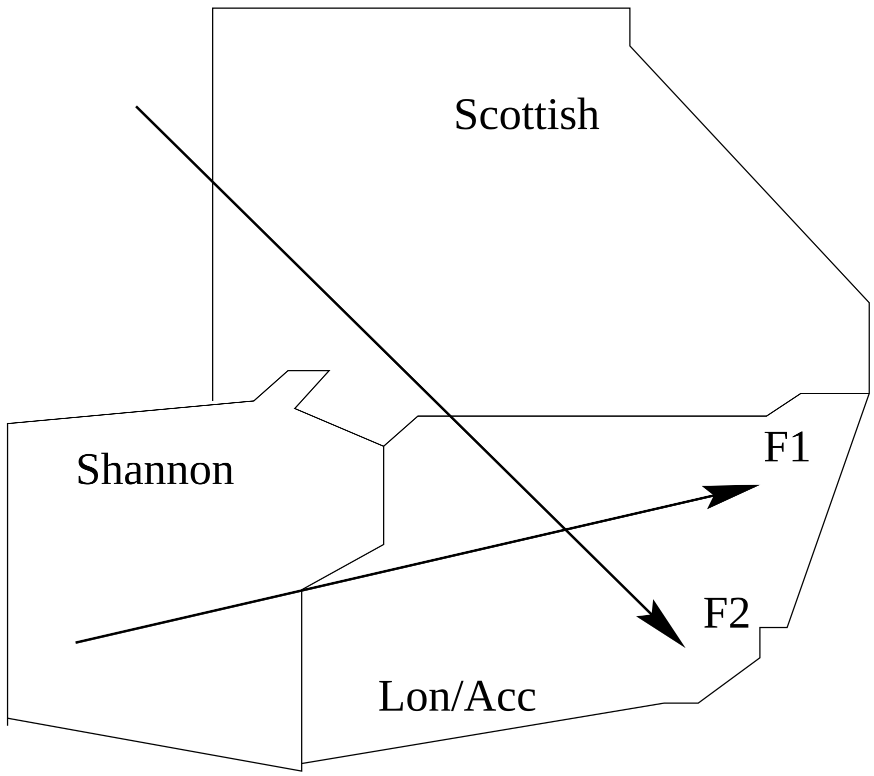
- flight traffic over Europe is increasing rapidly (70 % in 10 years)
  $\Rightarrow$ *congestion of air traffic control sectors*
- central management of European air traffic by Eurocontrol since 1995.
  $\Rightarrow$ *more than 20000 flights to treat each day*

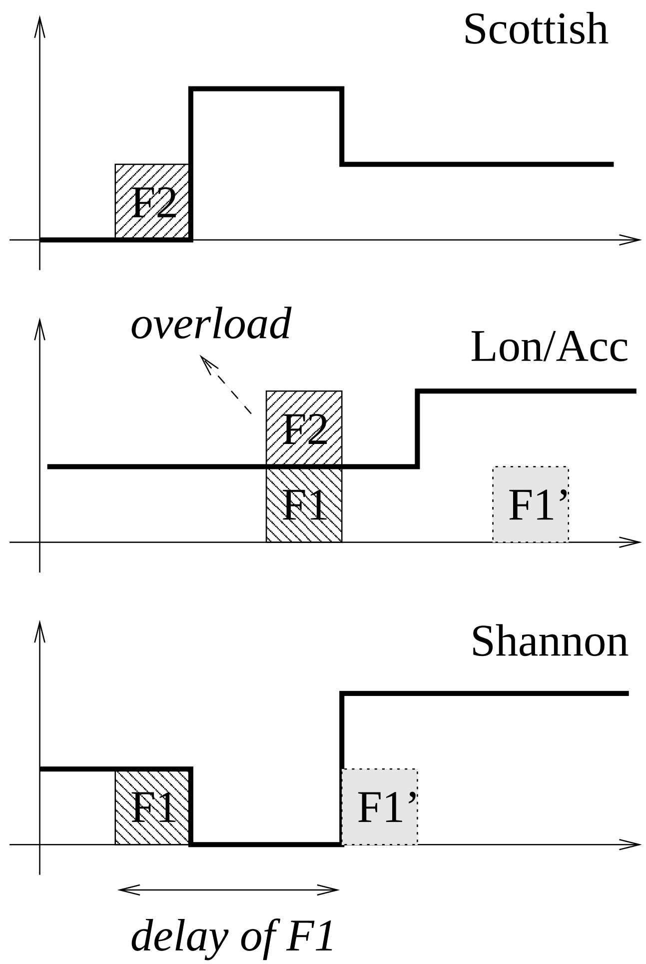
# Objectives of ATFM

assign a take-off delay (slot!) to each flight such that

1. the capacities of sectors are respected
2. the flight delay is minimized
3. security is ensured
4. equity principles are respected

# Reducing Congestion



# Local Equity Principles

- First-come first served (FCFS)
  - the flights will enter a sector in the expected order
  - FCFS achieves minimal delay and optimal fairness if no flight enters multiple congested sectors
  - FCFS is infeasible in the general case
- FCFS for most-penalizing regulation
  - relaxed version that considers only the most-penalizing regulation for each flight
  - the principle is not optimal as delaying an earlier flight may reduce the total delay if it traverses multiple congested sectors
  - meaning of this relaxed principle?

# Questions

- How much can the total flight delay be reduced if the FCFS principle is not applied?
- Can such an allocation be done online as frequent replannings (e.g. each 5 min.) are necessary during the day of operation?

study on innovative slot allocation algorithms (1995-97)

# Outline

- Eurocontrol's Air Traffic Flow Management Problem
- How to develop a precise and accurate optimization model?
- How to find good and precise solutions quickly?
- Experimental results with ILOG CP Optimizer

# European ATFM Problem (1997)

- $m$ sectors and $n$ flights
- flights $F_j$ entering sector $j$
- expected-time over $eto_{i,j}$ for each $i \in F_j$
- capacity $c_{j,k}$ limits the number of flights that enter sector $j$ during $[s_{j,k}, e_{j,k})$
  - contractual constraints: capacity per hour
  - smoothing constraints: capacity per intervals of $5$ or $10$ min.
- maximal delay $d_{max}$

# Constraint Programming for ATFM

- Integer Variables:

  delay $d_i \in [0, d_{max}]$ of flight $i$

- Capacity Constraints:

  a limited number $c_{j,k}$ of flights can enter sector $j$ during $[s_{j,k}, e_{j,k})$:

  $$card\ \{i \in F_j \mid s_{j,k} \leq d_i + eto_{i,j} < e_{j,k}\}\ \leq\ c_{j,k}$$

- Objective:

  minimize the total delay $D = \sum_{i=1}^{n} d_i$

# Integer Programming for ATFM

- approach:

  adapted from [Bertsimas and Stock, 1994]

- time representation:

  – all times $s_{j,k}, e_{j,k}, eto_{j,k}$ are rounded to multiples of a given $\Delta$ (e.g. $5$ minutes)

  – the binary variable $d_{i,t}$ has the value $1$ iff the flight $i$ has at least the delay $t$

- structural constraints

  if the delay of flight $i$ is at least $t + \Delta$ then it is at least $t$

$$d_{i,t} \geq d_{i,t+\Delta}$$

# Integer Programming (cnrd)

- capacity constraints

$$\sum_{i \in F_j} \left(d_{i,round(s_{j,k}-eto_{j,k})} - d_{i,round(e_{j,k}-eto_{j,k})}\right) \; \leq \; c_{j,k}$$

- objective:

  minimize the total delay $D = \sum_{i=1}^{n} \sum_t \Delta \cdot d_{i,t}$

- problems
  - large size (48000 variables for 2000 flights)
  - no exact results: extra delay caused by rounding operations

# Outline

- Eurocontrol’s Air Traffic Flow Management Problem
- How to develop a precise and accurate optimization model?
- How to find good and precise solutions quickly?
- Experimental results with ILOG CP Optimizer

# Problem Solving Methods

- chronological scheduling:
  - achieves first-come, first-served on a global basis
  - first solutions are clearly non-optimal
- decomposition by time periods: not possible
  - decisions about a period influence past and future periods
- heuristic repair: good reduction of delay
  - repair violations of capacity constraints (overloads)
  - minimizes violations by delaying flights that traverse several overloaded sectors.

# Heuristic Repair [Minton]

- Search state
  - variables have a current value
  - constraints have a degree of violation
- Repair action:
  - choose a violated constraint $C$
  - choose a variable $x$ of $C$ and a new value $v'$ of $x$ such that *violations* are minimized by changing the value of $x$ from $v$ to $v'$
  - assign $v'$ to $x$
- Properties:
  - a variable can be repaired several times
  - search can enter cycles

# Heuristic Repair & Tree Search

- Search state
  - variables have a current value and a domain
  - constraints have a degree of violation
- Repair action:
  - choose a violated constraint $C$
  - choose a variable $x$ of $C$ and a new value $v'$ from the domain of $x$ such that *violations* are minimized by changing the value of $x$ from $v$ to $v'$
  - branch: assign $v'$ to $x$ or remove $v'$ from $x$'s domain
- Properties:
  - a variable can be repaired only once on a branch
  - dead-ends can be encountered frequently

# Least-commitment strategy

- Search state
  - variables have a current value and a domain
  - constraints have a degree of violation
- Repair action:
  - choose a violated constraint $C$, a variable $x$ with current value $v$ and new value $v'$ as before
  - left branch: remove $v$ from $x$'s domain and use $v'$ as new current value
  - right branch: assign $v$ to $x$ and keep it as current value
- Properties:
  - a variable can be repaired several times on a left descent

# Heuristic Repair for ATFM

- current values:

  lower bounds $lb(d_i)$ of variables $d_i$

- violations:

  overloads of capacity constraints

$$
\begin{aligned}
L_{j,k} &= \{i \in F_j \mid s_{j,k} \leq lb(d_i) + eto_{i,j} < e_{j,k}\} \\
O_{j,k} &= max(card(L_{j,k}) - c_{j,k}, 0)
\end{aligned}
$$

- repair action:

1. choose $j, k$ with highest $O_{j,k}$
2. choose a flight $i \in L_{j,k}$ s.t. setting $lb(d_i)$ to $e_{j,k} - eto_{i,j}$ leads to the highest reduction of the sum of overloads
3. left branch: set $lb(d_i)$ to $e_{j,k} - eto_{i,j}$
4. right branch: set $ub(d_i)$ to $e_{j,k} - eto_{i,j} - 1$

# Outline

- Eurocontrol's Air Traffic Flow Management Problem
- How to develop a precise and accurate optimization model?
- How to find good and precise solutions quickly?
- Experimental results with ILOG CP Optimizer

# Experimental Results

example with 1989 flights and 16 sectors (= 1 day of traffic over France)

| Constraints | Strategy | Total delay | CPU time |
|---|---|---|---|
| contractual + | chronological | 32401 min | 0.17 sec. |
| 10 min-smoothing | heuristic repair | 21267 min | 0.69 sec. |
| only | chronological | 19441 min | 0.15 sec. |
| contractual | heuristic repair | 12492 min | 0.26 sec. |
| only | chronological | 16887 min | 0.15 sec. |
| 10-min-smoothing | heuristic repair | 11537 min | 0.5 sec. |

heuristic repair reduces the delay by about $30\%$

# Conclusion

- Modelling:

  CP allows ATFM models of precise time granularity and avoids rounding errors of IP models that use time steps of 5 minutes

- Solving:

  Heuristic repair strategy (with least-commitment branching) achieves a good delay minimization for the ATFM problem while allowing online allocation during the day of operation.

# Open Questions

- Explanations:
  - stakeholders need explanations to accept a solution
  - explanation of optimality is a new research topic
    see IJCAI-09 Tutorial on Explanations in Problem Solving
- Decision Theory:
  - what is the theoretically well-founded formalization of objectives such as equity?
  - interesting topic for Algorithmic Decision Theory as studied by European COST Action IC0602 + ADT Conference in Venice, Oct 2009