Large scale airline optimization

Dag Wedelin
Chalmers University of Technology
Gothenburg, SWEDEN

Traditional steps in airline scheduling

1. timetable is determined
2. fleet assignment
3. crew pairing
4. crew rostering

Usually done by different departments.
Traditional steps in airline scheduling

1. timetable is determined
2. fleet assignment
3. crew pairing
4. crew rostering

Usually done by different departments.
## Crew pairing example

<table>
<thead>
<tr>
<th>Leg number</th>
<th>From</th>
<th>To</th>
<th>Departure</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
<td>6.30</td>
<td>13.30</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>A</td>
<td>14.30</td>
<td>21.30</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>C</td>
<td>10.15</td>
<td>11.45</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>B</td>
<td>12.15</td>
<td>13.45</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>C</td>
<td>14.15</td>
<td>15.45</td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>B</td>
<td>16.15</td>
<td>17.45</td>
</tr>
</tbody>
</table>

**Rules:**

- A pairing must start and end in A or B
- Max 2 day pairings
- Max duration 12 hours/day
- Layover min $1.5 \times$ duration of first day
The two main steps of optimization

1) **Model** the problem, i.e. define it and formulate it so that a suitable algorithm can accept it.

2) **Solve** the resulting problem.

Both steps can be very challenging for large and complex real world problems. The steps must be carefully interfaced.

What is an optimization problem?

Example: shortest path problem

Minimize

\( \text{the length} \) over \( \text{all paths from a to b in a graph} \)

Minimize

\( \text{the objective function} \) over \( \text{the set of feasible solutions} \)
Mathematical optimization

Minimize
\[ x_1 + 3 x_2 + x_3 \]

subject to
\[ x_1 + x_2 = 1 \]
\[ x_1 + 2 x_3 \leq 2 \]
\[ x_i \text{ binary} \]

Good standard software available

Standard optimization models in transportation

**Polynomial:**
- Assignment, transportation, transshipment.
- Single depot vehicle scheduling (SDVSP, scheduling= tasks fixed in time).

(can all be translated into network flow)

**More difficult** (polynomial or exponential depending on details):
- Multiple depot vehicle scheduling (translates into multicommodity flow).
- Shortest path with time windows and/or resource constraints (use dynamic programming),
- Vehicle routing (VRP), Vehicle routing with time windows (VRPTW), pickup/delivery, … (try greedy+local search or two step approach),
Very difficult to model the rules mathematically.

Rules are defined in an application specific rule language (RAVE).

Compiles to test functions in C code.

Rules can be changed and tried out by users anytime!

```
RULE max_duty =
  %duty% <= 9:00;
REMARK "Maximal duty";
END
```

```
rule max_duty_a =
  %fdp_length% <= %max_duty_acclimatized%;
remark "Maximum scheduled FDP acclimatized";
end

%fdp_length% = las.'leg, arrival) - first.'leg, departure) + 2:00;

%max_duty_acclimatized% = matrix m_max_duty_acclimatized;

matrix m_max_duty_acclimatized =

%lendingso,%local_checkin% -> %max_duty_acclimatized%;
(0,1),2, 3, 4, 5, 6;
(06:00,07:59); 12:00, 11:45, 11:00, 10:15, 9:00, 0:00;
(08:00,12:59); 12:00, 12:00, 11:45, 10:15, 0:00;
(13:00,17:59); 12:00, 12:00, 11:30, 10:45, 9:15, 0:00;
(18:00,21:59); 12:00, 11:15, 10:30, 9:45, 9:00, 0:00;
(22:00,05:59); 11:00, 10:15, 9:30, 9:00, 9:00, 0:00;
end
```

In this code, the rule \texttt{max\_duty\_a} specifies that the expression \texttt{%fdp\_length%} must be less than the expression \texttt{%max\_duty\_acclimatized%}. These expressions are then defined, \texttt{%fdp\_length%} by accessing some data in the line of work, and \texttt{%max\_duty\_acclimatized%} by the use of a "matrix", whose (scalar) value is determined by table lookup based on the values of \texttt{%lendingso%} and \texttt{%local\_checkin%}.
System Components

Data entry from and to crew members (internet, cell phones, etc).

SSIM files, crew data, external events, etc

Crew Communicator

Graphical User Interface (Manual Scenario Editor)

Data storage

Optimizers

Report Generator

Rave (legality)

External systems (e.g. hotel booking, etc)

Standard methods for solving transport optimization problems

1. Time-space networks and network flow, or special cases thereof (assignment, transportation, transshipment)
2. Construction heuristics with greedy and local search
3. Two-step approach for general task scheduling: generate separate paths first, then optimize how to combine them. Column generation

Attention to small details and special properties of the problem at hand is crucial for best performance.
Crew pairing example

<table>
<thead>
<tr>
<th>Leg number</th>
<th>From</th>
<th>To</th>
<th>Departure</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
<td>6.30</td>
<td>13.30</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>A</td>
<td>14.30</td>
<td>21.30</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>C</td>
<td>10.15</td>
<td>11.45</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>B</td>
<td>12.15</td>
<td>13.45</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>C</td>
<td>14.15</td>
<td>15.45</td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>B</td>
<td>16.15</td>
<td>17.45</td>
</tr>
</tbody>
</table>

Rules:

- A pairing must start and end in A or B
- Max 2 day pairings
- Max duration 12 hours/day
- Layover min $1.5 \times$ duration of first day

Enumeration of all legal pairings

<table>
<thead>
<tr>
<th>P1</th>
<th>1-10-2 (2)</th>
<th>P7</th>
<th>5-6 (1)</th>
<th>P13</th>
<th>3-10-4-5-6 (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td>1-10-3-4-2 (2)</td>
<td>P8</td>
<td>3-4-5-6 (1)</td>
<td>P14</td>
<td>3-4-5-10-4 (2)</td>
</tr>
<tr>
<td>P3</td>
<td>1-5-10-4-2 (2)</td>
<td>P9</td>
<td>2-10-1 (2)</td>
<td>P15</td>
<td>3-4-5-10-4-5-6</td>
</tr>
<tr>
<td>P4</td>
<td>1-5-6-10-2 (2)</td>
<td>P10</td>
<td>2-10-1-5-6 (2)</td>
<td>P16</td>
<td>5-10-4 (2)</td>
</tr>
<tr>
<td>P5</td>
<td>3-4 (1)</td>
<td>P11</td>
<td>3-10-4 (2)</td>
<td>P17</td>
<td>5-10-4-5-6 (2)</td>
</tr>
<tr>
<td>P6</td>
<td>3-6 (1)</td>
<td>P12</td>
<td>3-10-6 (2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

lo: layover
The resulting optimization problem

\[
\begin{align*}
\text{minimize} & \quad 2x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + 2x_9 + 2x_{10} + 2x_{11} + 2x_{12} + 2x_{13} + 2x_{14} + 2x_{15} + 2x_{16} + 2x_{17} \\
\text{subject to} & \quad x_1 + x_2 + x_3 + x_4 + x_9 + x_{10} = 1 \\
& \quad x_1 + x_2 + x_3 + x_4 + x_9 + x_{10} = 1 \\
& \quad x_2 + x_5 + x_6 + x_8 + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1 \\
& \quad x_2 + x_3 + x_5 + x_8 + x_{11} + x_{13} + 2x_{14} + 2x_{15} + x_{16} + x_{17} = 1 \\
& \quad x_3 + x_4 + x_7 + x_8 + x_{10} + x_{13} + x_{14} + 2x_{15} + x_{16} + 2x_{17} = 1 \\
& \quad x_4 + x_6 + x_7 + x_8 + x_{10} + x_{12} + x_{13} + x_{15} + x_{17} = 1 \\
& \quad x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17} \in \{0, 1\}
\end{align*}
\]

Chalmers
E.P. 20.115
6 (27)

Crew scheduling

Main design conflict:
*rule modelling* versus *optimization*

Try to separate them!
Pairing optimization overview

Algorithm design techniques
How solve the large problems?

• Selection of daily subproblems.

• Generator search heuristics. E.g. generate pairings that follow the aircraft.

• Special purpose algorithm for the large optimization problems (10^6 variables, 10^4 constraints).

• Heuristic solution to deadheading

More recent developments

Send feedback from optimizer to generator.

The idea is to get new pairings that may improve current solution (solve constrained shortest path problem in generator).

Nice mathematical theory but tricky in practice.

A challenge to integrate with the separate rule system.
The set covering optimizer

Solves the problem

\[
\begin{align*}
\min \ c^T x \\
A x &\geq 1 \\
C x &\leq d \quad \text{(base capacity constraints)}
\end{align*}
\]

Solved using numerical algorithms:

- CPLEX for small and medium-sized problems
- probl or pays for large problems

developed at Chalmers!

The simple assignment problem

\[
\begin{align*}
\min \sum_{i,j} c_{ij} x_{ij} \\
\text{subject to} & \\
\sum_{j} x_{ij} &= 1 \\
\sum_{i} x_{ij} &= 1 \\
x_{ij} &\in \{0, 1\}
\end{align*}
\]
Some observations

Observation 1: some instances are easy to solve just by inspecting the cost matrix

\[
\begin{array}{cccc}
4 & -3 & 11 & 2 \\
5 & 2 & -8 & 7 \\
-6 & 9 & 2 & 4 \\
3 & 3 & 4 & -1 \\
\end{array}
\]

Observation 2: the problem is invariant with respect to additions or subtractions of any constant to a row or column.

Simple idea: use the invariants to make the problem easy!

Operational description

Example assignment problem

\[
\begin{array}{c|c|c|c}
\text{old } \bar{c} & \text{new } \bar{c} \\
\hline
5 & 2 & 9 & 7 \\
6 & 1 & 1 & 8 \\
1 & 9 & 5 & 3 \\
4 & 5 & 0 & 6 \\
\hline
\end{array}
\]

Look at each constraint separately and choose

\[y_i = \frac{-r^+ + r^-}{2}\]

Strategize!
What happens when we run this simple algorithm?

For simple assignment it works fine!

But for less trivial problems we get lots of zeros in \( C \):

\[ \text{eq. 8-queens with diagonal constraints} \]

No unique solution \( \implies \) useless!

What shall we do?

Coordinate search in dual space
The heuristic

Now it suddenly works!

(weighted 8-queens problem)

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Sweep mechanism

\[ x > 0 \] gives fast convergence and usually a high quality integer solution!

Interplay between different algorithmic design principles!

No exponential blowup \( \Rightarrow \) very large problems ok!

Performance

Same general picture supported by regular benchmarking (2004)
“Carmen is one of the best investments we have ever made”

British Airways

Optimization trends

10 years ago - an overnight run, today - within seconds.

Combined effect of faster algorithms and faster computers.

This makes it possible to:

- improve modelling detail (gives larger problems)
- integrate planning steps (gives larger problems)
- approach day-of-operation planning (requires faster optimizers)