Column Generation Methods for Disrupted Airline Schedules

Niklaus Eggenberg
Dr. Matteo Salani and Prof. Michel Bierlaire

In collaboration with APM Technologies
Funded by CTI Switzerland
Index

- Airline Scheduling
- The Airplane Recovery Problem (ARP)
- The Column Generation (CG) approach
- Solving the pricing problem with Recovery Networks
- Implementation and results
- Future work and conclusions
Airline Scheduling Approach

1. Route Choice
2. Fleet Assignment
3. Tail Assignment
4. Crew Pairing
5. Crew Roistering
6. Passenger Routing (catering)
Maintenances

Maintenances are forced by RESOURCE consumption (eg. flown hours)

Resources are renewed during maintenance
Disrupted Schedule and Recovery

Initial Schedule → Disruption → Disrupted Schedule

Disruption

Back to normal → Recovery

The Airplane Recovery Problem (ARP)

**Input**
- Planes’ States
- Initial Schedule
- Maintenances
- Cancelation Costs
- Delay Cost

**Output**
- \( T \)
- New schedule up to \( T \)
- Recovery cost
Definitions:

**PLANES:**
Initial State: position, initial time, initial resource consumption
Final State: position, expected time, expected resource consumption
Feasible Flight Set: coverable flights
Feasible Final State Set: coverable final states

**AIRPORTS:**
Activity Slots: periods when take-off/landings are permitted
Maintenance Slots: periods when given plane type can perform maintenance
Definitions (2):

**Flights:**

Origin and Destination

Scheduled Departure Time (SDT)

Flight Duration

Flight Cost

Cancelation Cost
Determine a Final State:
Determine a Final State:
Determine a Final State:

The ARP (4)

GVA  AMS  BCN  MIL  BUD

T

TRANSP-OR
Solution to the ARP:

A recovery scheme for each plane:
Multi-objective optimization:

Minimize both $T$ and recovery costs

Strategy: for fixed $T$ find optimal recovery plan

Give several recovery plans for different values of $T$ (decision aid)
Column Generation Approach

Find out optimal solution by combining individual recovery schemes \( r \in R' \) (master problem) on a subset \( R' \subseteq R \) of all feasible recovery schemes.

Generate potentially improving recovery schemes \( r \in R-R' \) dynamically for each plane (pricing problem).
Master Problem: MIP formulation

\[
\begin{align*}
\min \quad & z_{MP} = \sum_{r \in R} c_r x_r + \sum_{f \in F} c_f y_f + \sum_{s \in S} c_s z_s \\
\text{s. c.} \quad & \sum_{r \in R} b^f_r x_r + y_f = 1 \quad \forall f \in F \quad (\lambda_f) \\
\sum_{r \in R} b^s_r x_r + z_s = 1 \quad \forall s \in S \quad (\eta_s) \\
\sum_{r \in R} b^p_r x_r \leq 1 \quad \forall p \in P \quad (\mu_p) \\
\end{align*}
\]

\[x_r \in \{0,1\} \quad \forall r \in R\]
\[y_f \in \{0,1\} \quad \forall f \in F\]
\[z_s \in \{0,1\} \quad \forall s \in S\]
What is a column?

- cost
- vector

\[ b_r = (b_r^f, b_r^s, b_r^p)^T \]

Where

- \( b_r^f = 1 \) if flight \( f \) is covered by column \( r \)
- \( b_r^s = 1 \) if final state \( s \) is covered by \( r \)
- \( b_r^p = 1 \) if column \( r \) is affected to plane \( p \)
The Pricing Problem

Find new columns minimizing the reduced cost $\tilde{c}_r^p$:

$$\min_{r \in R} \tilde{c}_r^p = c_r^p - \sum_{f \in F} b_r^f \lambda_f - \sum_{s \in S} b_r^s \eta_s - b_r^p \mu_p \quad \forall p \in P$$
Recovery Networks (Argüello et al. 97)

1. Generate a recovery network for each plane
2. Update arc costs according to dual variables
3. Solve Resource Constrained Elementary Shortest Path (RCESPP)
4. Add Columns to $R'$
5. Resolve restricted LP until optimality and branch
Time – Space Network with

- source node $n_0 = [t, m, r]$
- node $n = [t, m, r]$
- sink $s = [t, m, r]$

- flight arc $[n, n']$
- maintenance arc $[n, n']$
- termination arc $[n, s]$
- maintenance termination arc $[n, s]$
Recovery Network
Updating arc costs

- flight arcs: \( c = c^f + c^d - \lambda_f \)
- maintenance arcs: \( c = c^f + c^d + c^M - \lambda_f \)
- termination arcs: \( c = -\eta_s \)
- maintenance term. arcs: \( c = -\eta_s + c^M \)

Solve RCESPP on networks returns column minimizing the reduced cost!

Righini & Salani (2006), which is an extension of Desrochers et al. (1988)
Some References

- **Argüello et al.** (1997): recovery without maintenance
  up to 27 planes, 162 flights, 30 airports

- **Desrosiers et al.** (1997): daily scheduling NOT recovery
  up to 91 planes, 383 flights, 33 airports; max delay of 30 minutes

- **Clarke** (1997): maintenances requirements but no decision on them
  up to 177 planes, 612 flights, 37 airports; only 0 or 30 min delay

- **Kohl et al.** (2004): Descartes project, good survey of state of the art
  no instance size mentioned for DAR

- **Barnhart and Bratu** (2006): passenger oriented recovery algorithm
  up to 302 planes, 1032 flights, 74 airports
Implementation Issues

Implementation issues

- Implemented in C++ with COIN-OR BCP framework
- Used interior point methods to solve the LP
- Used linear time and logarithmical resource discretisation
- 2 phase pricing:
  - generation (keep also non optimal columns, heuristic pricing)
  - proving optimality (optimal column only, exact pricing)
Implementation Issues (2)

Linear Time Discretization

Logarithmic Resource Discretization
Real Instances

• Got real schedules from Thomas Cook Airlines (APM’s main customer)

• Solved original schedules up to 250 flights (algorithm validation)

• Generated disruption scenarios
  - delayed planes (initial states)
  - grounded planes (initial states)
  - airport closures (activity slots)
  - forced maintenances (initial resource consumption)
## Solved Instances (2): Problem Sizes

<table>
<thead>
<tr>
<th>Instance</th>
<th>2D_5AC</th>
<th>2D_5AC_1del</th>
<th>2D_10AC</th>
<th>2D_10AC_1del</th>
<th>2D_10AC_2del</th>
</tr>
</thead>
<tbody>
<tr>
<td># planes</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td># flights</td>
<td>38</td>
<td>38</td>
<td>75</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td># delayed planes</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td># cancelled flights</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td># delayed flights</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>total delay [min]</td>
<td>0</td>
<td>969</td>
<td>0</td>
<td>969</td>
<td>989</td>
</tr>
<tr>
<td>max delay [min]</td>
<td>0</td>
<td>370</td>
<td>0</td>
<td>370</td>
<td>370</td>
</tr>
<tr>
<td>cost</td>
<td>380(*)</td>
<td>21175(*)</td>
<td>750(*)</td>
<td>21545(*)</td>
<td>21745(*)</td>
</tr>
<tr>
<td>tree size</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>run time [s]</td>
<td>&lt; 0.1</td>
<td>&lt; 0.1</td>
<td>0.7</td>
<td>0.7</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>3D_10AC</th>
<th>4D_10AC</th>
<th>5D_5AC</th>
<th>5D_10AC</th>
<th>7D_16AC</th>
</tr>
</thead>
<tbody>
<tr>
<td># planes</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td># flights</td>
<td>113</td>
<td>147</td>
<td>93</td>
<td>184</td>
<td>242</td>
</tr>
<tr>
<td># delayed planes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td># cancelled flights</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td># delayed flights</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>total delay [min]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>310</td>
</tr>
<tr>
<td>max delay [min]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>cost</td>
<td>1130(*)</td>
<td>1470(*)</td>
<td>930(*)</td>
<td>1840(*)</td>
<td>5600</td>
</tr>
<tr>
<td>tree size</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2033</td>
</tr>
<tr>
<td>run time [s]</td>
<td>3.0</td>
<td>6.5</td>
<td>1.0</td>
<td>29.1</td>
<td>3603</td>
</tr>
</tbody>
</table>
Solved Instances (3): Added value of maintenances

Average results of 10 randomly generated instances

<table>
<thead>
<tr>
<th>Instance</th>
<th>No maint. + 5%</th>
<th>No maint. + 10%</th>
<th>No maint. + 20%</th>
</tr>
</thead>
<tbody>
<tr>
<td># cancelled flts</td>
<td>52.7</td>
<td>46.7</td>
<td>33.2</td>
</tr>
<tr>
<td># delayed flts</td>
<td>5</td>
<td>4.7</td>
<td>5.5</td>
</tr>
<tr>
<td># uncovered final states</td>
<td>1.2</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>total delay [min]</td>
<td>851.3</td>
<td>635.7</td>
<td>712.5</td>
</tr>
<tr>
<td>max delay [min]</td>
<td>271.3</td>
<td>251.5</td>
<td>218.2</td>
</tr>
<tr>
<td>cost</td>
<td>289462</td>
<td>272067</td>
<td>144388</td>
</tr>
<tr>
<td>optimality gap [%]</td>
<td>0.61</td>
<td>0.54</td>
<td>1.27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>Greedy maint.</th>
<th>Maint. Opt</th>
</tr>
</thead>
<tbody>
<tr>
<td># cancelled flts</td>
<td>2.2</td>
<td>2</td>
</tr>
<tr>
<td># delayed flts</td>
<td>2.7</td>
<td>1.5</td>
</tr>
<tr>
<td># uncovered final states</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>total delay [min]</td>
<td>89.6</td>
<td>52.3</td>
</tr>
<tr>
<td>max delay [min]</td>
<td>37.7</td>
<td>37.1</td>
</tr>
<tr>
<td>cost</td>
<td>15881</td>
<td>14683</td>
</tr>
<tr>
<td>optimality gap [%]</td>
<td>0.73</td>
<td>0</td>
</tr>
</tbody>
</table>

Considering maintenances is crucial!!!
Solved Instances (4): Pareto Optimality

Pareto behavior for increasing $T$
Future Work

• Benchmark solutions against practitioners
• Allow repositioning flights and early departures
• Extend Pricing Solver for acceleration
• Include in APM solutions
Conclusions

• Developed a flexible and fast algorithm

• Solutions are very promising

• Maintenance planning is an added value
THANKS for your attention!

Any Questions?