Robust scheduling and disruption recovery for airlines

Niklaus Eggenberg, Matteo Salani

Transport and mobility laboratory
ENAC - EPFL

IDSIA
July 03, 2009
Outline

1 Introduction
2 Aircraft Recovery
3 Passenger Recovery
4 Alternative Robust Scheduling
5 Recoverable robustness
6 Conclusions
Outline

1 Introduction
2 Aircraft Recovery
3 Passenger Recovery
4 Alternative Robust Scheduling
5 Recoverable robustness
6 Conclusions
Airlines $=\text{Complexity}$

- International / Intercontinental
- Network of flights
- Aircraft types (heterogeneous fleet)
- Airport capacities (gates, slots)
- Air traffic control
- Security/Environmental regulations
- Strict safety requirements (maintenance)
- Infinite workforce rules
- Complicated cost structure/pricing
- High competition/Uncertain demand

This is a **nightmare** for practitioners and **fun** for researchers.
Low margins (around 2-3%)
Airline planning process

Decision Level

Strategic
- Flight Schedule
  - OD, departure frequency
- Forecast Demand

Tactical
- Fleet assignment
  - Aircraft Routing
  - Crew rotations
    - Crew pairing
  - Passenger pricing

Operative
- Disruption recovery
  - Fleet, Crews, Passengers

Resources

Long term
- Strategic
- Long term

Short term
- Tactical
- Day Ops
- Short term
Disruptions are unpredicted events which significantly modify the assumptions (data) used for decision making.

For example, in 2007, 21.1% of departures and 22.3% of arrivals in Europe were delayed by more than 15 minutes. The average cancellation rate is about 1.5% for short haul and 0.6% for long haul.

Robust decision making accounts for noise in the data to obtain a more stable system.
The most robust plan

No service is not acceptable!
Recovery strategies

When the system is in a disrupted state (data is revealed or a disruption happens), we refer to a *recovery strategy* as the sequence of actions to restore an operational state of the system.

Often used as an *alternative* to robust planning to cope with noisy data.

Recovery algorithms tend to minimize the total cost of additional operations to restore operational state. Different baseline solutions may incur in different recovery costs given the same disruption.
Outline

1. Introduction
2. Aircraft Recovery
3. Passenger Recovery
4. Alternative Robust Scheduling
5. Recoverable robustness
6. Conclusions
Aircraft recovery problem (ARP)

Restore an operational state minimizing *recovery time* and *costs*.

The problem:

- Given a **baseline** schedule.
- Recover within a given time horizon an airline schedule in a disrupted state minimizing the recovery costs
- Known: ACs’ position, ACs’ expected position, airports, passengers itineraries
- Recovery cost structure used by a major European airline: linear combination of flight and itinerary cancellation and delay, swapping, up-down grading, . . .
Data

After data preprocessing, the relevant informations are:

- $F$: a set of scheduled flights, together with an estimation of cancellation cost $c_f$
- $P$: a set of aircrafts
- $R$: a set of passengers (itineraries)
- $I_p, I_r$: a set of initial positions for both aircrafts and passengers
- $S_p, S_r$: a set of required final positions for both aircrafts and passengers
- $T$: a time horizon
- $L$: a set of airport slots
- $q_i^{Dep}, q_i^{Arr}$: slot capacities for take off and landings
Master problem

We model the recovery problem for aircrafts as:

\[
\begin{align*}
\min z_{MP} &= \sum_{r \in \Omega} c_r x_r + \sum_{f \in F} c_f y_f \\
\sum_{r \in \Omega} b_{r}^{f} x_r + y_f &= 1 \quad \forall f \in F \\
\sum_{r \in \Omega} b_{r}^{s} x_r &= 1 \quad \forall s \in S_p \\
\sum_{r \in \Omega} b_{r}^{p} x_r &\le 1 \quad \forall p \in P \\
\sum_{r \in \Omega} b_{r}^{Dep,l} x_r &\le q_{l}^{Dep} \quad \forall l \in L \\
\sum_{r \in \Omega} b_{r}^{Arr,l} x_r &\le q_{l}^{Arr} \quad \forall l \in L \\
x_r &\in \{0, 1\} \forall r \in \Omega, \ y_f &\in \{0, 1\} \forall f \in F
\end{align*}
\]

Recovery Network

Given $T$, $I_p$ and $S_p$ the R.N. encodes all possible recovery schemes for plane $p$.

Scheduled flights, acyclic, polynomial size
Recovery Network

Given $T$, $I_p$ and $S_p$ the R.N. encodes all possible recovery schemes for plane $p$.

Delay modeling, acyclic network but no more acyclic in terms of flights, exponential size
Recovery Network

Given $T$, $I_p$ and $S_p$ the R.N. encodes all possible recovery schemes for plane $p$.

Time band discretization pseudo-polynomial size but unfeasible recovery schemes are encoded
Generating recovery schemes

Given $\Omega'$, $(x_r^*, y_f^*)$, $(\lambda_f^*, \eta_s^*, \mu_p^*, \nu_l^*, \rho_l^*)$, new profitable schemes for plane $p$ are computed by solving an ERCSP on the Recovery Network, minimizing:

$$\tilde{c}_r^p = c_r^p - \sum_{f \in F} b_r^f \lambda_f^* - \sum_{s \in S} b_r^s \eta_s^* - \mu_p^* - \sum_{l \in L} \left( b_{r \text{Dep}, l}^p \nu_l^* + b_{r \text{Arr}, l}^p \rho_l^* \right) \quad \forall p \in P$$

Bi-directional bounded dynamic programming with DSSR. Righini and S. (2008).
Implementation issues

The algorithm is implemented with BCP framework by COIN-OR.

Speed up, to comply with restricted time limitations:

- Network size is reduced by some parameters: permitted delay, permitted plane swaps
- Pricing problem is solved heuristically with relaxed domination criteria and label elimination
- Heuristic search tree exploration
- Primal heuristics
Real world instances

Mid-size airline (500 flights/week) - synthetic instances (delays, cancellations, apt closures and maintenance disruptions)

<table>
<thead>
<tr>
<th></th>
<th>+ 5%</th>
<th>+ 10%</th>
<th>+ 20%</th>
<th>Heur</th>
<th>Opt</th>
</tr>
</thead>
<tbody>
<tr>
<td># canceled flts</td>
<td>5</td>
<td>4.7</td>
<td>5.5</td>
<td>2.7</td>
<td>1.5</td>
</tr>
<tr>
<td># delayed flts</td>
<td>52.7</td>
<td>46.7</td>
<td>33.2</td>
<td>2.2</td>
<td>2</td>
</tr>
<tr>
<td># uncovered final states</td>
<td>1.2</td>
<td>0.7</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>total delay [min]</td>
<td>851.3</td>
<td>635.7</td>
<td>712.5</td>
<td>89.6</td>
<td>52.3</td>
</tr>
<tr>
<td>max delay [min]</td>
<td>271.3</td>
<td>251.5</td>
<td>218.2</td>
<td>37.7</td>
<td>37.1</td>
</tr>
</tbody>
</table>

Eggenberg, S. And Bierlaire (2008a).
Outline

1. Introduction
2. Aircraft Recovery
3. Passenger Recovery
4. Alternative Robust Scheduling
5. Recoverable robustness
6. Conclusions
Passenger recovery

An integer solution to $z_{MP}$ gives the aircraft assignment and the flight re-timing or cancellation. From that solution we build a unique connection network which comply with connectivity constraints:

- Arc capacities represent available seats

Passenger itineraries are sorted according to deletion cost and for each itinerary:

- Dummy source and sink connections are the only updated
- Cost of arcs connecting the sink represent the delay cost
- A min-cost flow is solved and decomposed into paths
- Each path is a new itinerary
## Results for passenger recovery

<table>
<thead>
<tr>
<th>Instance</th>
<th>A01</th>
<th>A02</th>
<th>A03</th>
<th>A04</th>
</tr>
</thead>
<tbody>
<tr>
<td># canceled psg</td>
<td>41</td>
<td>33</td>
<td>196</td>
<td>79</td>
</tr>
<tr>
<td># rerouted psg</td>
<td>235</td>
<td>2848</td>
<td>587</td>
<td>2468</td>
</tr>
<tr>
<td># delayed psg</td>
<td>8664</td>
<td>7852</td>
<td>10430</td>
<td>9969</td>
</tr>
<tr>
<td>total delay [min]</td>
<td>280312</td>
<td>259133</td>
<td>581312</td>
<td>557593</td>
</tr>
<tr>
<td>average delay [min]</td>
<td>32.3</td>
<td>33.0</td>
<td>55.7</td>
<td>55.9</td>
</tr>
<tr>
<td>recovery costs</td>
<td>89477</td>
<td>11351</td>
<td>342267</td>
<td>219789</td>
</tr>
<tr>
<td>run time [s]</td>
<td>0.66</td>
<td>3155</td>
<td>0.95</td>
<td>1806</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>A05</th>
<th>A06</th>
<th>A07</th>
<th>A08</th>
<th>A09</th>
</tr>
</thead>
<tbody>
<tr>
<td># canceled psg</td>
<td>44</td>
<td>10</td>
<td>441</td>
<td>148</td>
<td></td>
</tr>
<tr>
<td># rerouted psg</td>
<td>243</td>
<td>2779</td>
<td>445</td>
<td>2462</td>
<td></td>
</tr>
<tr>
<td># delayed psg</td>
<td>11293</td>
<td>10469</td>
<td>13007</td>
<td>12997</td>
<td></td>
</tr>
<tr>
<td>total delay [min]</td>
<td>350257</td>
<td>332786</td>
<td>700504</td>
<td>715910</td>
<td></td>
</tr>
<tr>
<td>average delay [min]</td>
<td>31.0</td>
<td>31.8</td>
<td>53.9</td>
<td>55.1</td>
<td></td>
</tr>
<tr>
<td>recovery costs</td>
<td>99893</td>
<td>64859</td>
<td>632736</td>
<td>268534</td>
<td></td>
</tr>
<tr>
<td>run time [s]</td>
<td>0.47</td>
<td>3781</td>
<td>0.72</td>
<td>2562</td>
<td></td>
</tr>
</tbody>
</table>

Table: Results for the PRP - ROADEF dataset.
Outline

1. Introduction
2. Aircraft Recovery
3. Passenger Recovery
4. Alternative Robust Scheduling
5. Recoverable robustness
6. Conclusions
Approaches toward robustness

(Airline) schedule disruptions occur because of unpredicted events (noise in the nominal data) which are of stochastic nature.

**Reactive and proactive approaches**

- Online optimization (Albers (2003))
- Stochastic optimization (with recourse) (Kall and Wallace (1994))
- Worst-case (robust) optimization (Bertsimas and Sim (2004))
- Risk-management/Light robustness (Kall and Mayer (2005), Fischetti and Monaci (2008))
Uncertainty set

Often uncertainty sets are **difficult** to estimate. Wrong estimation of uncertainty set may lead to even more **unstable** solutions.

Example on cargo loading value maximisation (multi-dimension knapsack), simulation over 16200 scenarios.

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad (A_i + \varepsilon_i)x \leq b_i \quad \forall i \in I \\
x & \in \{0, 1\}
\end{align*}
\]

Robust solutions has an average optimality gap of 10%.
When simulated coefficient realization deviates significantly (50-70%) from estimated we obtain more unfeasible solution for the robust approach than the deterministic.
An alternative approach

We aim to design an optimization framework which:

- **simple**, has the same complexity as the deterministic problem
- provides solutions with guaranteed deviation from optimum
- does not need for probabilistic uncertainty sets
- accounts for reactive strategies

We search a robust *recoverable* solution.
Robustness features

Given a deterministic optimization problem:

$$\min f(x)$$
$$s.t. \ x \in X$$

Identify **structural properties** $\mu(x)$ of a solution which are exploited by the reactive strategy. Solve a multi-objective optimization problem:

$$\min f(x), \max \mu(x)$$
$$s.t. \ x \in X$$

Relax original objective in a (budget) constraint:

$$\max \mu(x)$$
$$s.t. \ f(x) \leq (1 + \rho)f(x^*)$$
$$x \in X$$

**Remark:** stochastic and robust optimization can be obtained for specific $\mu(x)$ and $\rho$. 
Stochastic optimization, Birge and Louveaux (1997)

Stochastic optimization:

\[
\mu_{\text{Stoc}}(x) = - \mathbb{E}_U(f(x)) \\
z_{\text{Stoc}} = \min \mathbb{E}_U(f(x)) \\
\alpha(x) \leq b \\
f(x) \leq (1 + \rho)f^* \\
x \in X
\]

Stochastic optimization with recourse:

\[
\mu_{\text{Rec}}(x) = - [f(x) + \mathbb{E}_U(g(x, \xi))] \\
z_{\text{Rec}} = \min f(x) + \mathbb{E}_U(g(x, \xi)) \\
\alpha(x) \leq b \\
f(x) \leq (1 + \rho)f^* \\
x \in X
\]
Robust optimization, Bertsimas and Sim (2004)

\[
(F) \quad z^*_F = \min_{x \in X} \{ f(x) \} \\
= \min_{x \in X} \{ \max_{i=1, \ldots, n} (f_i(x)) \} \\
= \min_{x \in X} \{ \max_{i=1, \ldots, n} (\sum_{j=1}^m a_{ij} x_j + \beta_i(x, J_i) - b_i) \}
\]

\[
\rho = \max_{i=1, \ldots, n} \left\{ \frac{\rho_i f_i(x^*)}{z^*_F} - 1 \right\},
\]

where \( \rho_i \) is defined as the ratio:

\[
\rho_i = \begin{cases} 
\frac{\bar{\beta}_i(x, \Gamma_i)}{f_i(x^*)} & \text{if } f_i(x^*) > 0 \\
0 & \text{otherwise.}
\end{cases}
\]

\[
\mu_{\text{Rob}}(x) = -c^T x
\]
Robust recoverable aircraft scheduling

**Tactical** planning: Re-timing of flights is permitted in the definition of $r \in \Omega$ within a range of 60 minutes.

\[
\begin{align*}
\text{max } z_{RF} = &\mu(x) \\
& (14) - (15) \\
& (17) - (21) \\
\sum_{r \in \Omega} c_r x_r + c_f y_f &\leq (1 + \rho) z_D^* \\
\forall r &\in \Omega \\
x_r &\in \{0, 1\} \\
\forall f &\in F \\
y_f &\in \{0, 1\}
\end{align*}
\]
Robust recoverable aircraft scheduling

**Tactical** planning: Re-timing of flights is permitted in the definition of $r \in \Omega$ within a range of 60 minutes.

$$\max z_{RF} = \mu(x)$$

(11)

$$\text{(14)} - \text{(15)}$$

$$\text{(17)} - \text{(21)}$$

(12)

(13)

$$\sum_{r \in \Omega} d_r x_r \leq C$$

(14)

$$x_r \in \{0, 1\} \quad \forall r \in \Omega$$

(15)
Outline

1. Introduction
2. Aircraft Recovery
3. Passenger Recovery
4. Alternative Robust Scheduling
5. Recoverable robustness
6. Conclusions
Robust recoverable aircraft scheduling

The recovery algorithm perform better in presence of slack time between flights and effective possibilities of swapping planes.

Increase the minimal idle time of schedule $r$

$$\mu_{IT}(x) = \sum_{r \in \Omega} \delta_r^{\text{min}} x_r$$

Quadratic formulation

$$\mu_{CROSS}(x) = \sum_{r \in \Omega} \sum_{p \in \Omega} b_{rp} x_r x_p$$
Robust recoverable aircraft scheduling

The recovery algorithm performs better in presence of slack time between flights and effective possibilities of swapping planes.

Increase the minimal idle time of schedule $r$

$$
\mu_{IT}(x) = \sum_{r \in \Omega} \delta_r^{\min} x_r
$$

We define meeting points $m$

$$
\sum_{r \in \Omega} b^m_r x_r - y_m \geq 0 \quad \forall m \in M
$$

$$
\mu_{CROSS}(x) = \sum_{m \in M} (y_m - 1)
$$
Robust results

Results on ROADEF09 set A instances (average)

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>CROSS</th>
<th>CROSS</th>
<th>IT</th>
<th>IT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUDGET [min]</td>
<td>0</td>
<td>5000</td>
<td>10000</td>
<td>5000</td>
<td>10000</td>
</tr>
<tr>
<td>RECOVERY COST</td>
<td>788775.1</td>
<td>633395.6</td>
<td>555400.3</td>
<td>488701.9</td>
<td>493521.8</td>
</tr>
<tr>
<td># Canceled Flts</td>
<td>6.9</td>
<td>6.9</td>
<td>5.3</td>
<td>5.8</td>
<td>5.9</td>
</tr>
<tr>
<td>Total Delay [min]</td>
<td>2142.9</td>
<td>2083.0</td>
<td>2421.8</td>
<td>2214.9</td>
<td>1895.6</td>
</tr>
<tr>
<td>Avg Delay[min]</td>
<td>41.0</td>
<td>37.9</td>
<td>42.0</td>
<td>36.9</td>
<td>36.5</td>
</tr>
<tr>
<td># Cancelled Psg</td>
<td>582.8</td>
<td>499.3</td>
<td>420.0</td>
<td>384.5</td>
<td>385.3</td>
</tr>
<tr>
<td># Delayed Psg</td>
<td>553.5</td>
<td>511.1</td>
<td>454.1</td>
<td>501.1</td>
<td>448.1</td>
</tr>
<tr>
<td>Avg Psg Delay [min]</td>
<td>34.6</td>
<td>38.7</td>
<td>24.6</td>
<td>29.5</td>
<td>29.8</td>
</tr>
</tbody>
</table>

Eggenberg And S. (2009).
Outline

1. Introduction
2. Aircraft Recovery
3. Passenger Recovery
4. Alternative Robust Scheduling
5. Recoverable robustness
6. Conclusions
Conclusions

- Noisy data and primary objectives can lead to unstable solutions.
- Computational tractability of hard combinatorial problems represent an issue for stochastic or robust optimization.
- In several cases, uncertainty set is hardly identifiable.
- Knowledge of reactive strategies can help.
- Robustness features are structural properties of a solution which are computationally tractable.
- Simultaneous robustness and recoverability is a promising approach for reliable operations planning.

Outlook:
- Under final validation on airline scheduling (robust passenger connections).
- To be validated on container terminal optimization.
- Applications to network planning with restoring to be explored.
- Comparison with stochastic optimization with recourse is still weak.
Thanks

Thanks for your attention

Any question?
References