Airline Disruptions: Aircraft Recovery with Maintenance Constraints

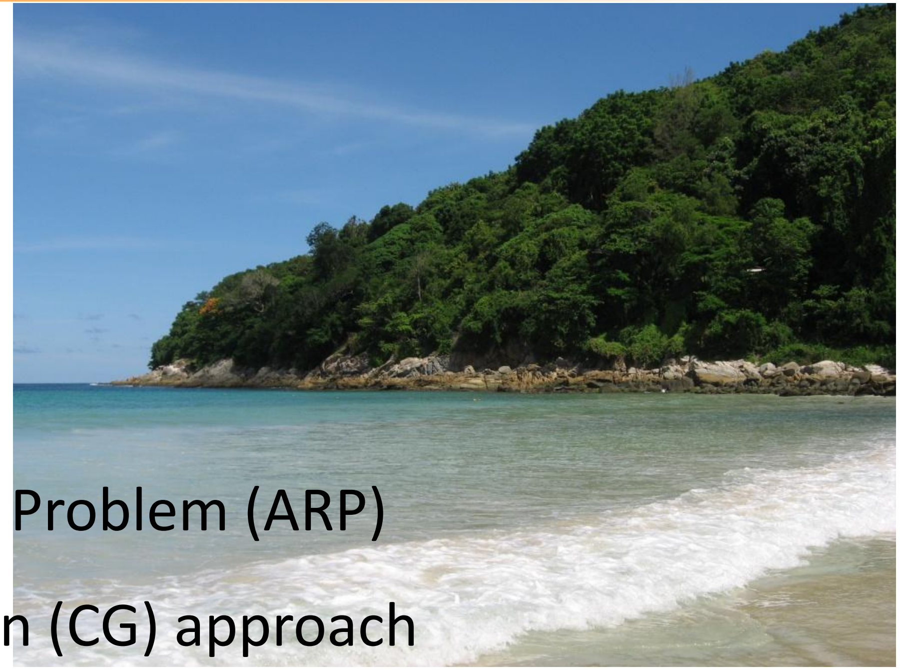
Niklaus Eggenberg
Dr. Matteo Salani and Prof. Michel Bierlaire

In collaboration with *APM Technologies*
Funded by *CTI Switzerland*


TRISTAN VI
JUNE 10-15 2007
Phuket Island-Thailand

Dr. Matteo Salani

Prof. Michel Bierlaire



Index

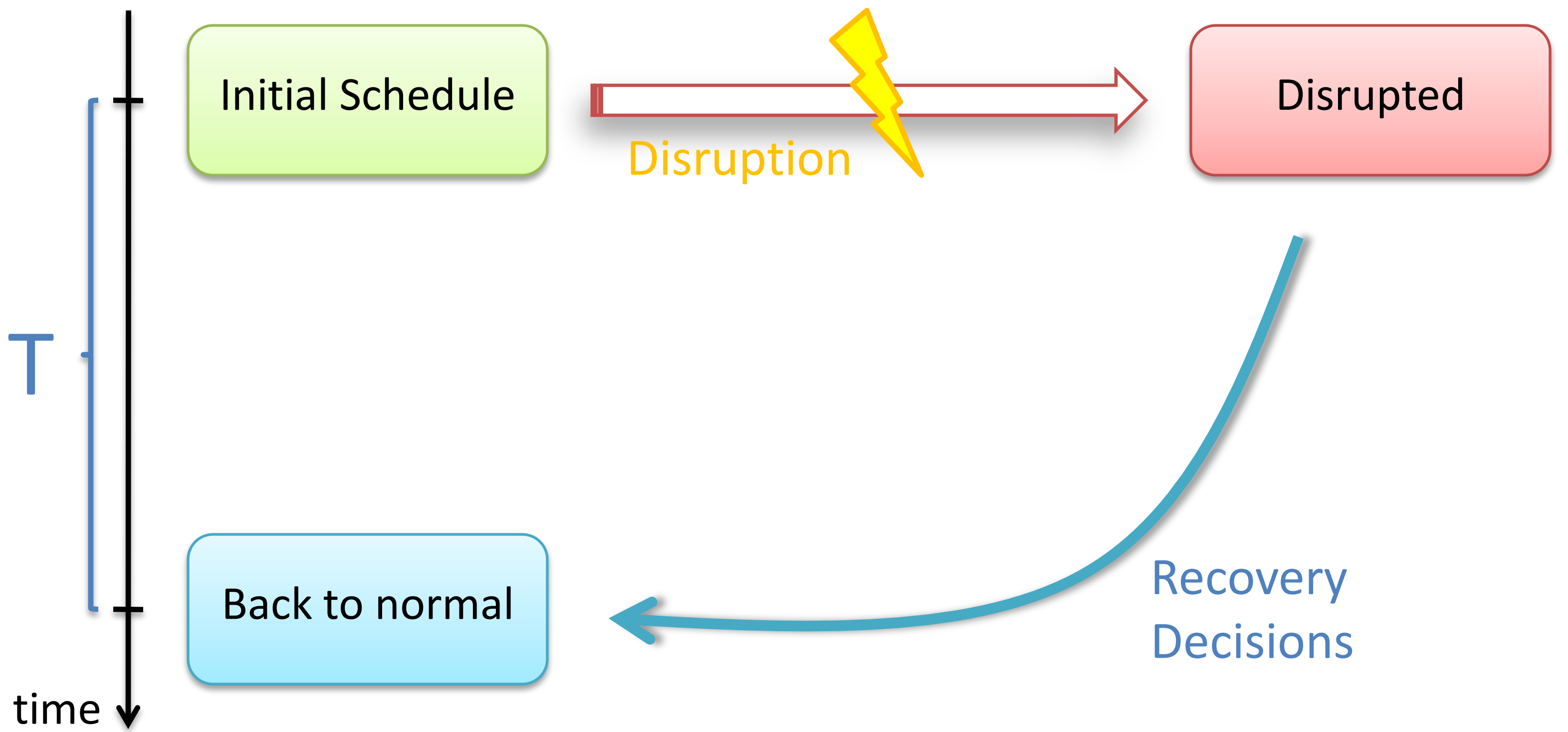
- Airline Scheduling
- The Airplane Recovery Problem (ARP)
- The Column Generation (CG) approach
- Column Description
- Solving the pricing problem with Recovery Networks
- Implementation and results
- Future work and conclusions

Airline Scheduling Approach

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

Fleet Assignment and Tail Assignment: Technical Schedule

Disrupted Schedule and Recovery

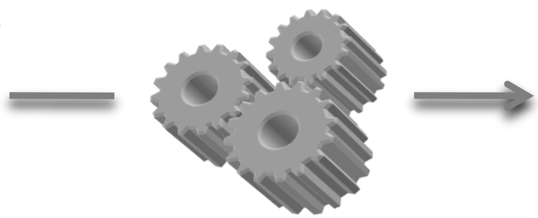


Survey: Kohl (2004)

The Airplane Recovery Problem (ARP)

Input

- Planes' States
- Initial Schedule
- Maintenances
- Cancelation Costs
- Delay Cost



Output

- T
- New schedule up to T
- Recovery cost

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)

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)

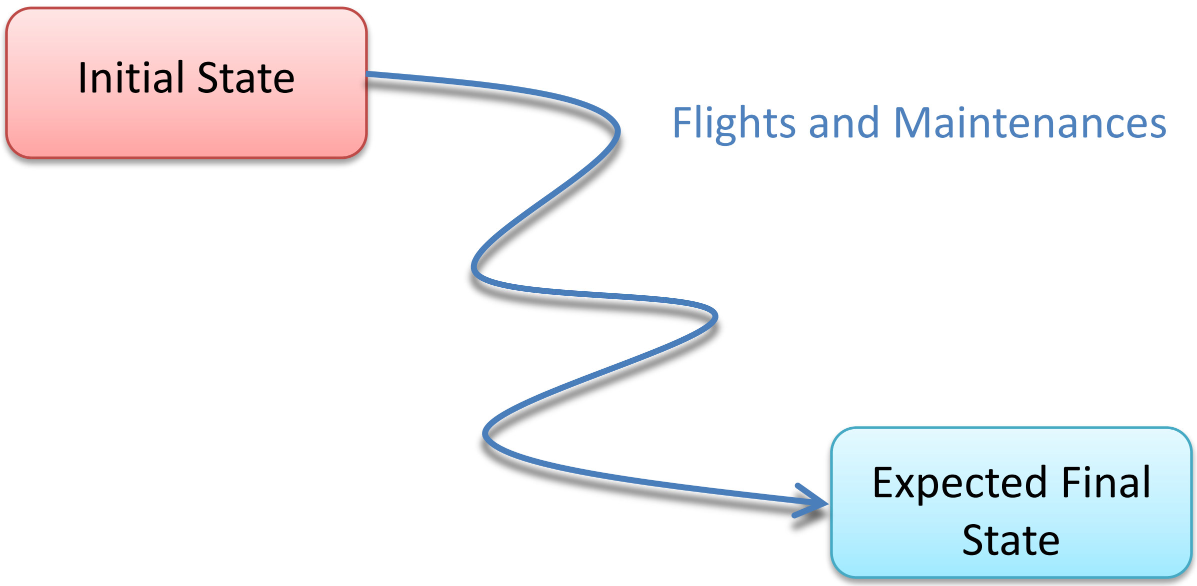
Flight Duration

Flight Cost

Cancelation Cost

Solution to the ARP:

A recovery scheme for each plane:



Example

f_1 GVA to AMS

f_2 AMS to BCN

f_3 BCN to GVA

f_4 MIL to BUD

f_5 BUD to MIL

f_6 BCN to MIL

Example (2)

- flights: $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$
- final states: $S = \{s^{GVA}, s^{MIL}\}$
- planes: $P = \{p_1, p_2\}$
- initial states: p_1 (*GVA*, 0, 0)

  p_2 (*MIL, 0, 0)*

Feasible Solution

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 $r \in R\text{-}R'$ dynamically for each plane (pricing problem)

Master Problem: MIP formulation

$$\min \quad z_{MP} = \sum_{r \in R} c_r x_r + \sum_{f \in F} c_f y_f + \sum_{s \in S} c_s z_s$$

$$s.c. \quad \sum_{r \in R} \boldsymbol{b}_r^f x_r + y_f = 1 \qquad \forall f \in F \quad (\lambda_f)$$

$$\sum_{r \in R} \boldsymbol{b}_r^s x_r + z_s = 1 \qquad \forall s \in S \quad (\eta_s)$$

$$\sum_{r \in R} \boldsymbol{b}_r^p x_r \leq 1 \qquad \forall p \in P \quad (\mu_p)$$

$$x_r \in \{0,1\} \qquad \forall r \in R$$

$$y_f \in \{0,1\} \qquad \forall f \in F$$

$$z_s \in \{0,1\} \qquad \forall s \in S$$

What is a column ?

- cost c_r
- vector $\boldsymbol{b}_r = \left(b_r^f, b_r^s, b_r^p \right)^T$

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*

Column examples

$$\boldsymbol{b}_1 = (0,0,0,0,0,0,1,0,1,0)^T$$

$$\boldsymbol{b}_2 = (1,1,1,0,0,0,1,0,1,0)^T$$

$$\boldsymbol{b}_3 = (0,0,0,1,1,0,0,1,0,1)^T$$

The Pricing Problem

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

$$\min_{r \in R} \ \tilde{c}_r^p = c_r^p - \sum_{f \in F} \boldsymbol{b}_r^f \lambda_f - \sum_{s \in S} \boldsymbol{b}_r^s \eta_s - \boldsymbol{b}_r^p \mu_p \qquad \forall\, p \in P$$

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

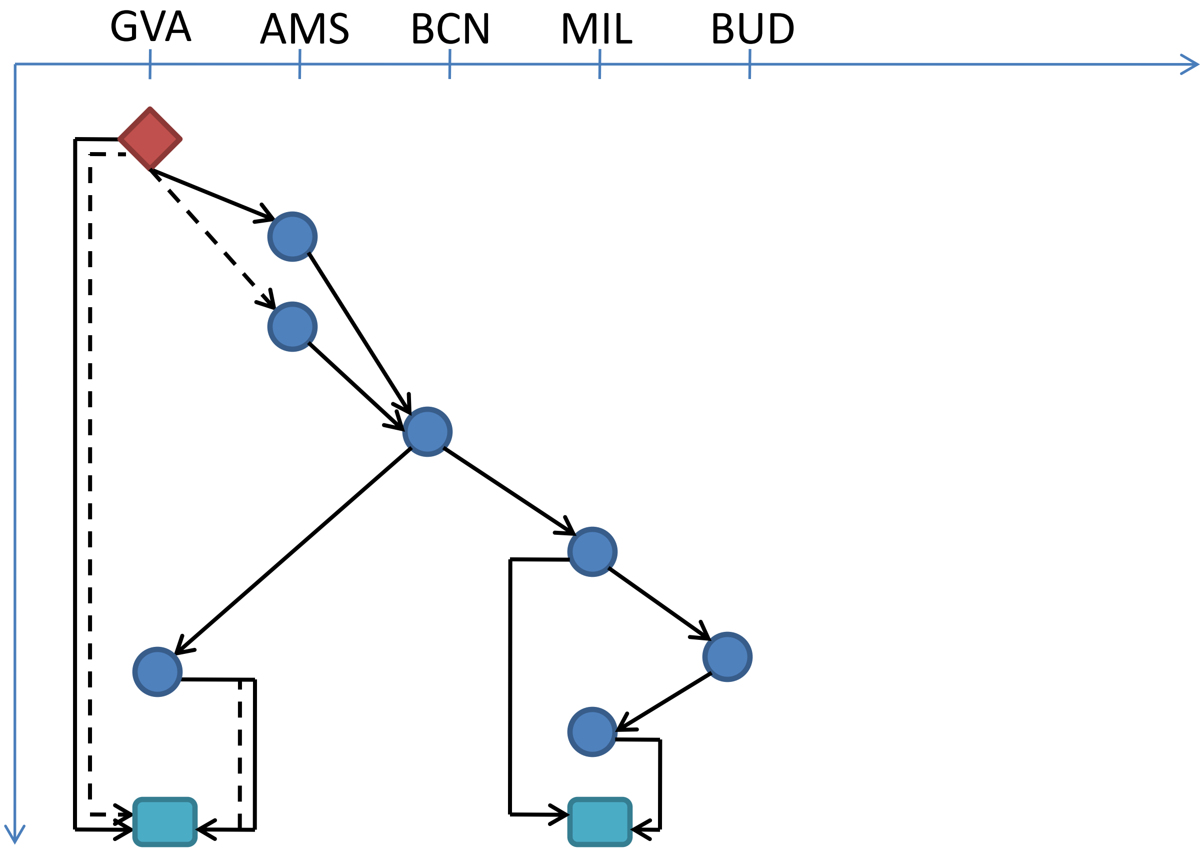
- source node n_0 = [t, m, $\boldsymbol{r}$]
- node n = [t, m, $\boldsymbol{r}$]
- sink s = [t,m,$\boldsymbol{r}$]

- flight arc [n, n']
- maintenance arc [n, n']
- termination arc [n,s]
- maintenance termination arc [n,s]

Example (continued)

- flights: $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$
- final states: $S = \{s^{GVA}, s^{MIL}\}$
- planes: $P = \{p_1, p_2\}$
- initial states: p_1 (*GVA*, 0, 0)
  p_2 grounded for the day
- only maintenances at GVA

Recovery Network of p_1



Generating Recovery Networks

- Create Source node n_0 (initial time, location, resource cons.)
- $S = \{n_0\}$
- While $S \neq \emptyset$:
  - Select $n \in S$, $S \leftarrow S - \{n\}$
  - For all feasible flights:
    - create flight and maintenance arcs
    - create destinations node n_f and n_m
    - $S = S \cup \{n_f, n_m\}$
- Clean 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)

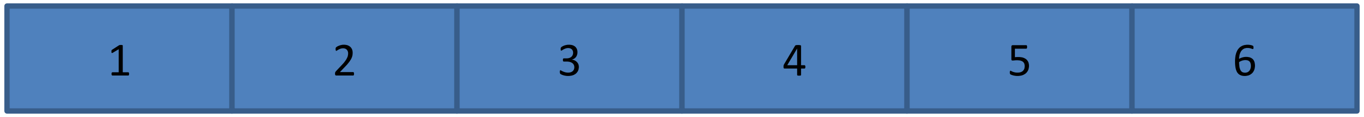
Some References

- 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

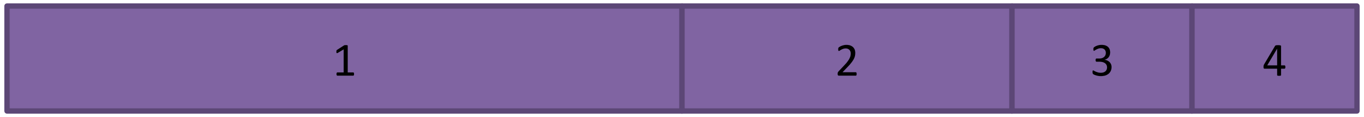
Implementation issues

- 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)

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

Instance	2D_5AC	2D_5AC_1del	2D_10AC	2D_10AC_1del	2D_10AC_2del
# planes	5	5	10	10	10
# flights	38	38	75	75	75
# delayed planes	0	1	0	1	2
# cancelled flts	0	2	0	2	2
# delayed flts	0	4	0	4	5
total delay [min]	0	969	0	969	989
max delay [min]	0	370	0	370	370
cost	380(*)	21175(*)	750(*)	21545(*)	21745(*)
tree size	1	1	1	1	1
run time [s]	< 0.1	< 0.1	0.7	0.7	1.0

Instance	3D_10AC	4D_10AC	5D_5AC	5D_10AC	7D_16AC
# planes	10	10	5	10	16
# flights	113	147	93	184	242
# delayed planes	0	0	0	0	0
# cancelled flts	0	0	0	0	0
# delayed flts	0	0	0	0	11
total delay [min]	0	0	0	0	310
max delay [min]	0	0	0	0	45
cost	1130(*)	1470(*)	930(*)	1840(*)	5600
tree size	1	1	1	5	2033
run time [s]	3.0	6.5	1.0	29.1	3603

Solved Instances (3): Behavior against disruptions

Instance	Den2del	Den2grd	Den4del	Den4grd	Den2del2grd	Den6del	Den6grd
# delayed planes	2	0	4	0	2	6	0
# grounded planes	0	2	0	4	2	0	6
# affected flights	1	4	3	8	5	5	16
# cancelled flts	0	2	0	8	4	0	16
# delayed flts	1	4	7	2	7	13	2
total delay	10	920	230	380	490	640	380
max delayed flight	10	275	85	200	200	100	200
cost	36100(*)	83200(*)	38300(*)	163800(*)	84900(*)	42400(*)	251800(*)
tree size	1	1	1	1	1	41	1
run time	0.7	0.5	0.6	0.3	0.5	1.6	0.2

Instance	Den3del3grd	Den_3x100	Den_1x300	Den_Storm1	Den_Storm2
# delayed planes	3	0	0	0	0
# grounded planes	3	0	0	0	0
# affected flights	9	11	7	3	6
# cancelled flts	6	0	4	0	0
# delayed flts	12	11	11	6	6
total delay	950	675	2560	350	1550
max delayed flight	200	90	385	140	340
cost	127500(*)	42750(*)	125600(*)	39500(*)	51500(*)
tree size	1	1	35	1	3
run time	0.4	0.3	0.8	0.5	0.5

Average results of 10 randomly generated instances

Instance	No maintenance	Dummy maintenance	Maintenance optimization
# cancelled flts	63.3	5.4	4.8
# delayed flts	4.3	3.1	1.1
# uncovered final states	2.2	0.5	0.3
total delay [min]	508	103.3	36.6
max delay [min]	222.2	35.7	31.6
cost	397214.5	36581.5	33074
optimality gap [%]	0.35	0.28	1.01
tree size	29.2	23	12
run time [s]	20.3	57.9	41.8

Considering maintenances is crucial!!!

Example of instance

Instance	No maintenance	Dummy maintenance	Maintenance optimization
# cancelled flts	57	2	0
# delayed flts	9	2	2
total delay [min]	546	61	79
max delay [min]	191	34	50
cost	339195	13310(*)	5760(*)
tree size	5	1	1
run time [s]	8.8	30.5	47.0

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?