An algorithm for the recovery of disrupted airline schedules

Niklaus Eggenberg
Dr. Matteo Salani and Prof. Michel Bierlaire

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

TRANSP-OR

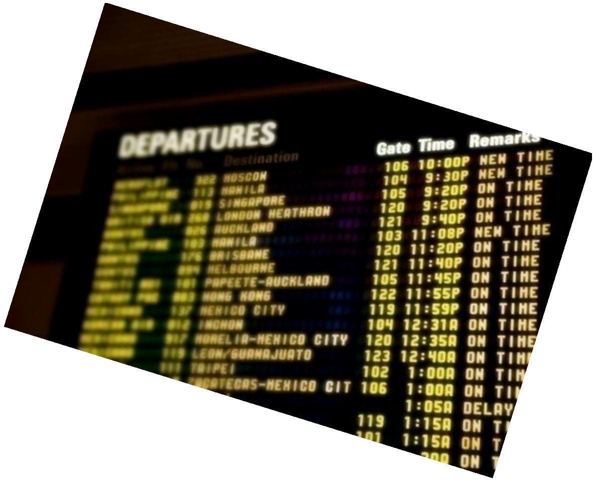
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Index

Airline Scheduling Approach

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

} Technical Schedule

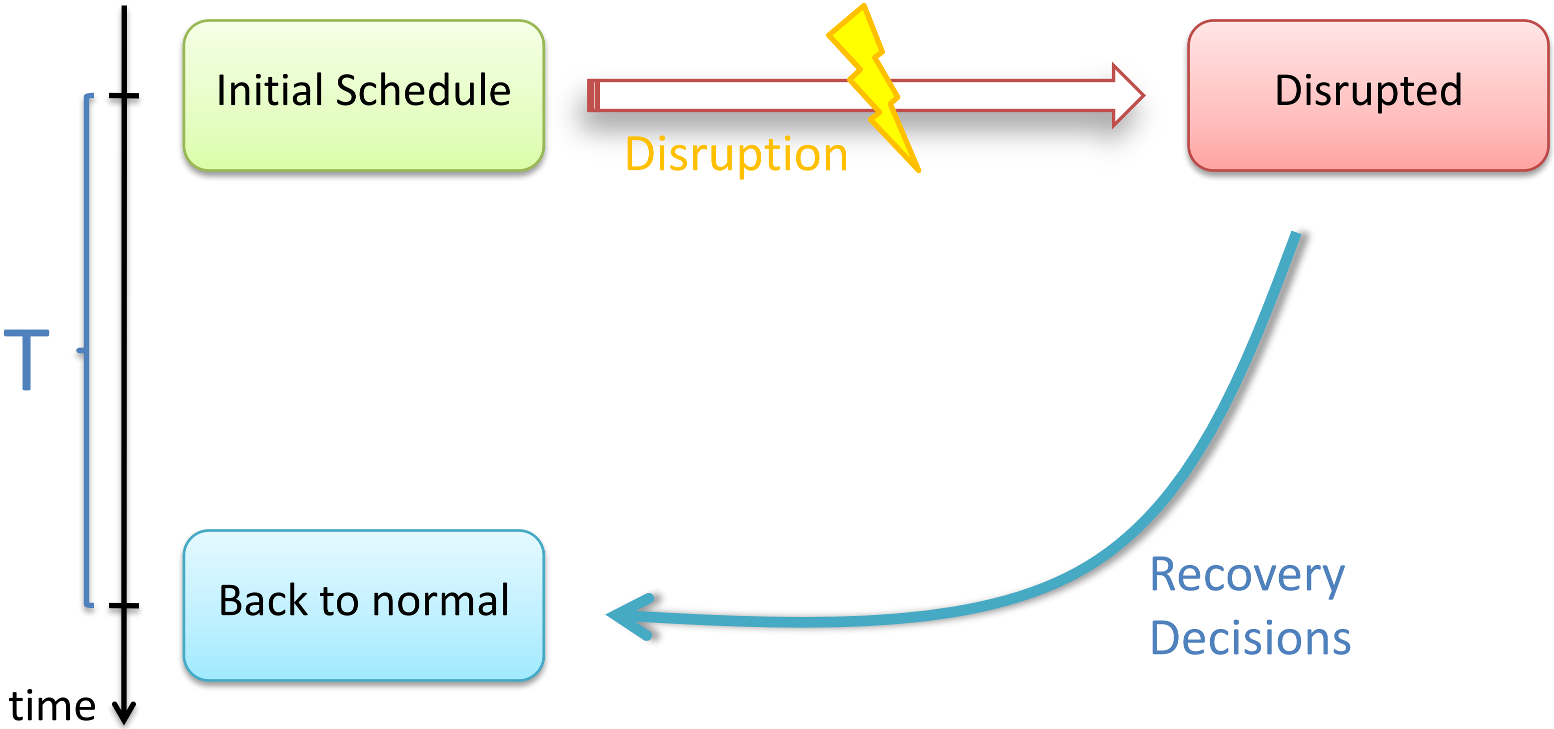


Maintenances

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

Resources are renewed during maintenance

Disrupted Schedule and Recovery



Initial Schedule

Disruption

Disrupted

Recovery Decisions

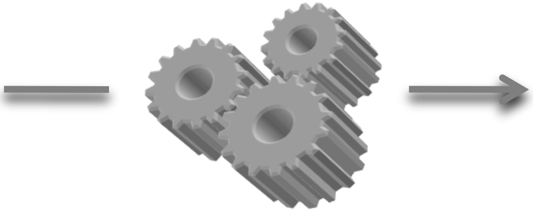
Back to normal

T

time

Survey: Kohl (2004)

The Airplane Recovery Problem (ARP)



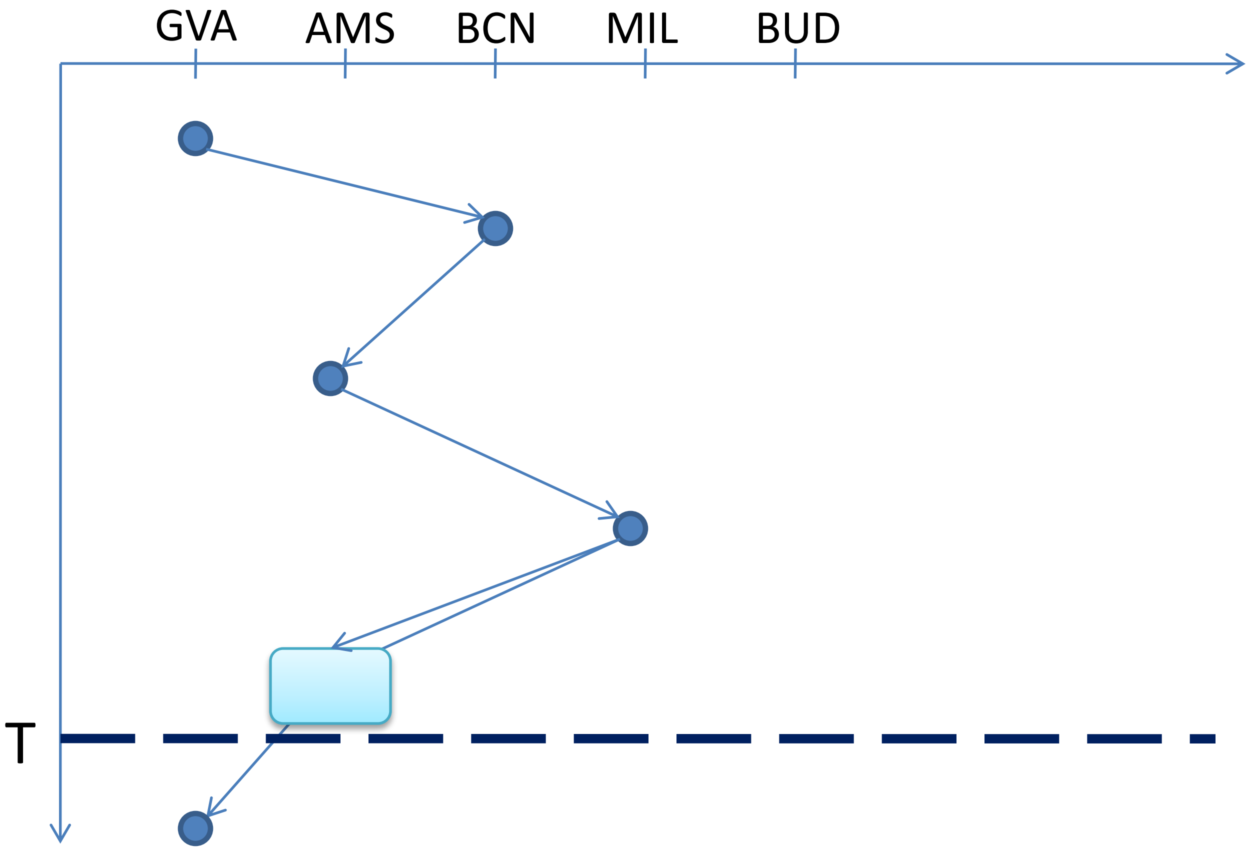
Input

- Planes’ Position
- Initial Schedule
- Maintenances
- Cancelation Costs
- Delay Cost

Output

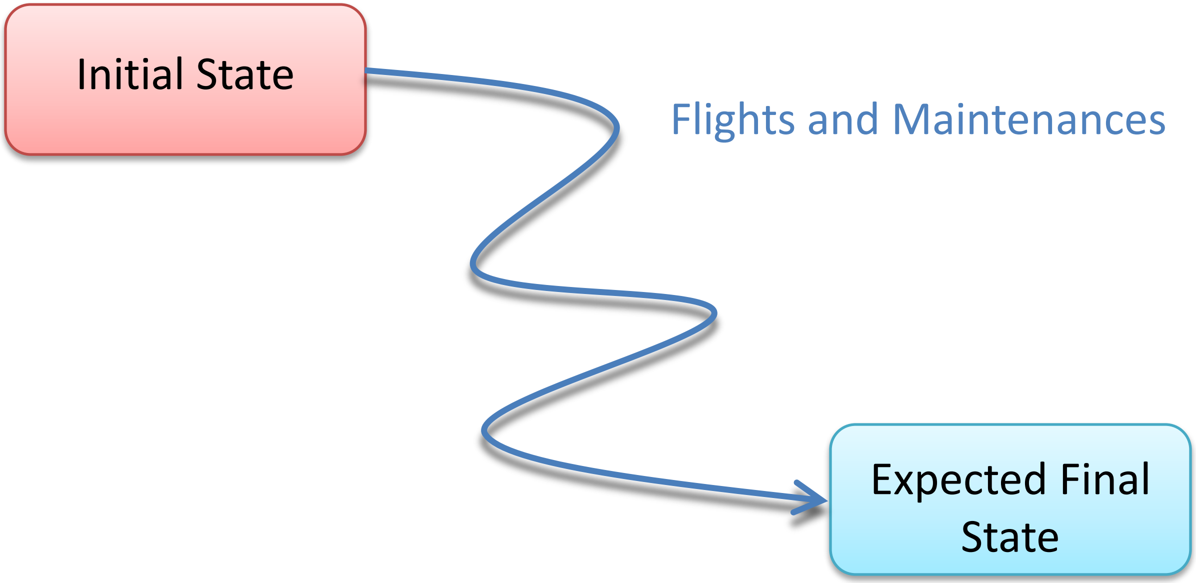
- T (Recovery Period)
- New schedule up to T
- Recovery cost

Determine a Final State:



Solution to the ARP:

A recovery scheme for each plane:



Initial State

Flights and Maintenances

Expected Final State

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)

Variables:

- x_r = 1 if route r is in the solution, 0 otherwise
- y_f = 1 if flight f is cancelled, 0 otherwise
- z_s = 1 if final state s is uncovered, 0 otherwise

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*

Master Problem: MIP formulation

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

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

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

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

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

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

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

The Pricing Problem

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

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

Solve Pricing

⇔

Solve Elementary Shortest Path Problem in Recovery Networks

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

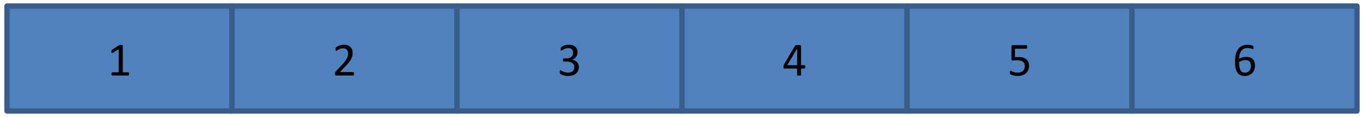
Originality in this work

- Considering Maintenances
- Apply Column Generation technique
- Improved Acceleration Techniques :
  - 2 phase pricing (heuristic and exact)
  - logarithmical discretization
- Network based pricing
- RCESPP algorithm (Righini & Salani, 2006)

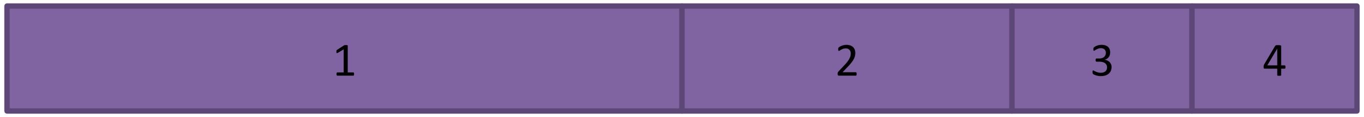
Implementation issues

- 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 (1): Solvable Problem Sizes (with small disruptions)

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

The Denver Instance: Hub and Spoke

- 10 planes
- 36 flights
- Initial Cost (without any disruption) : 36000
- *ndel* : number of delayed planes
- ngrd : number of grounded planes (whole period)
- 3x100 and 1x300 : hub closure for time periods and time lengths
- Storm : four local spoke airports closed for 300 (1) and 500 (2) min

The Denver Instance

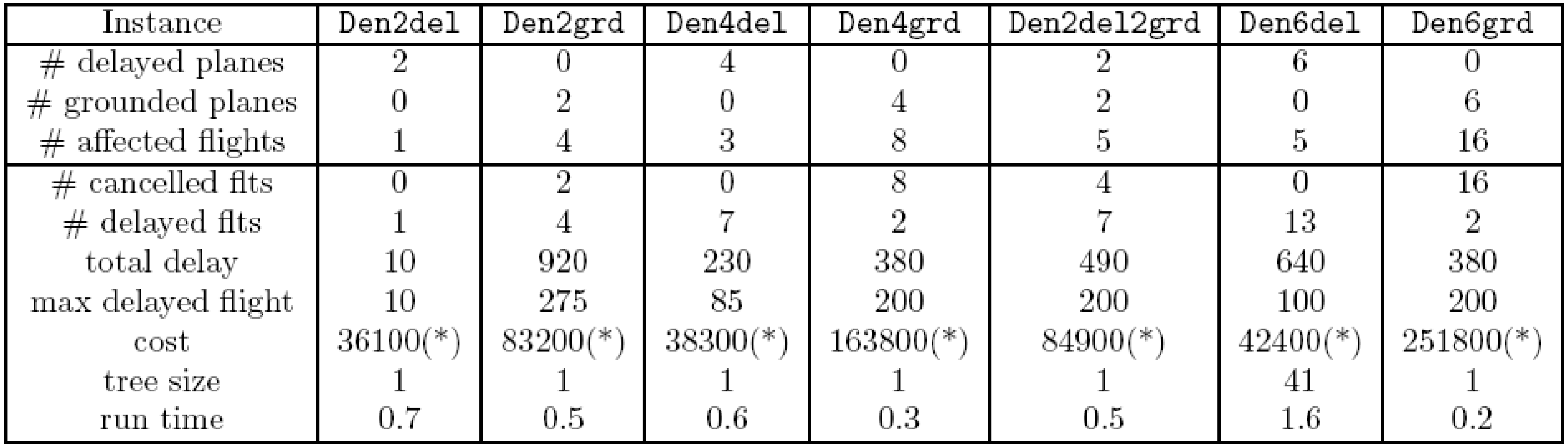
- Affected Planes = number of DIRECTLY affected planes (without considering propagation)

- Cancellation cost = 12000 to 22000 cost units

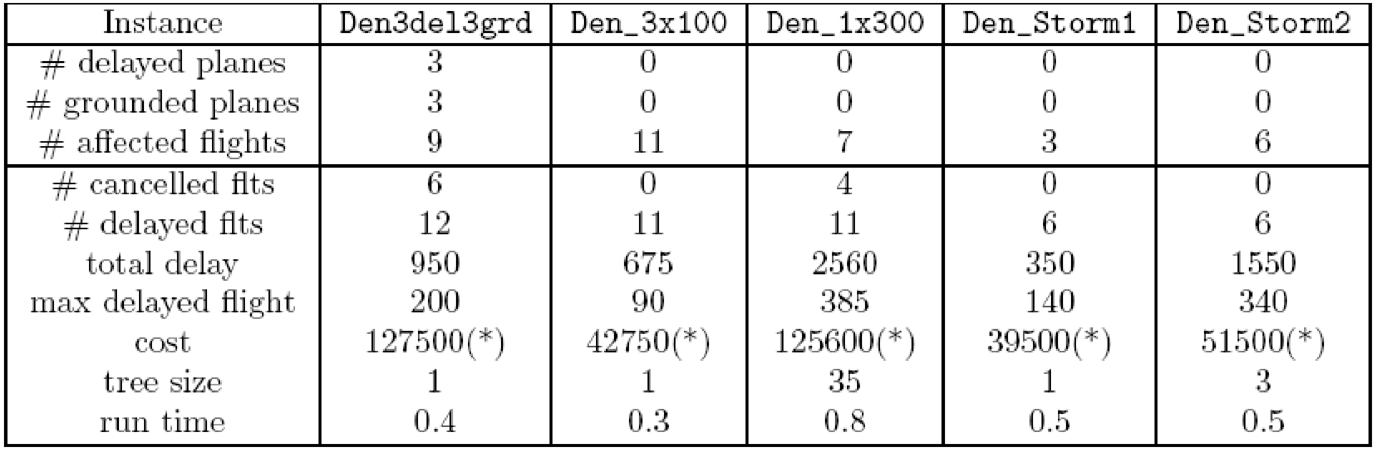
- Delay cost = 10 cost units per minute

Solved Instances (2): Behavior against disruptions

Initial Cost = 36000



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



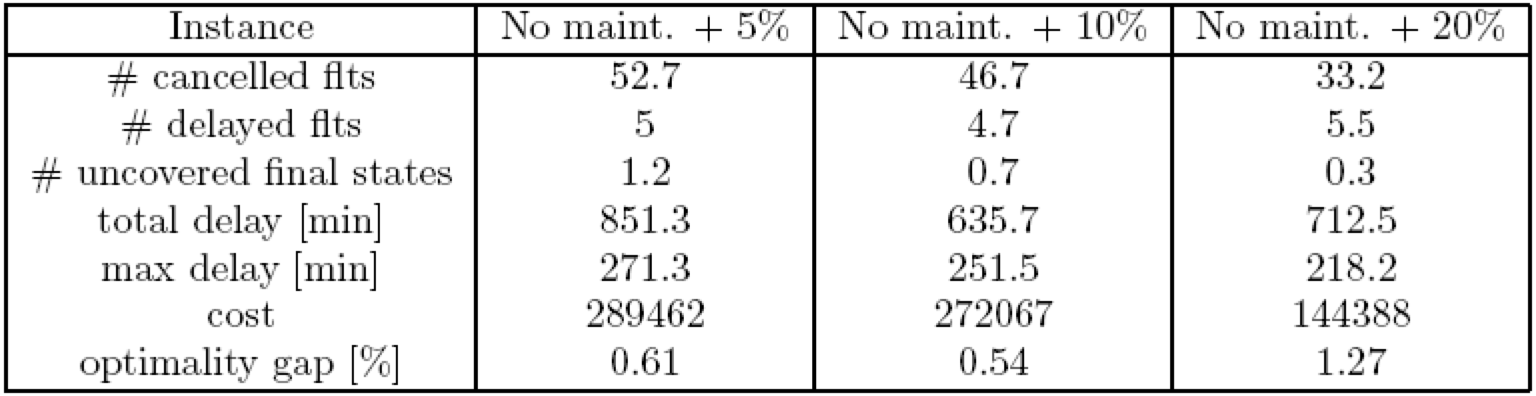
Instance	Den3del3grd	Den_3x100	Den_1x300	Den_Storm1	Den_Storm2
# delayed planes	3	0	0	0	0
# grounded planes	3	0	0	0	0
# affected flights	9	11	7	3	6
# cancelled flts	6	0	4	0	0
# delayed flts	12	11	11	6	6
total delay	950	675	2560	350	1550
max delayed flight	200	90	385	140	340
cost	127500(*)	42750(*)	125600(*)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

Maintenance Scheduling:

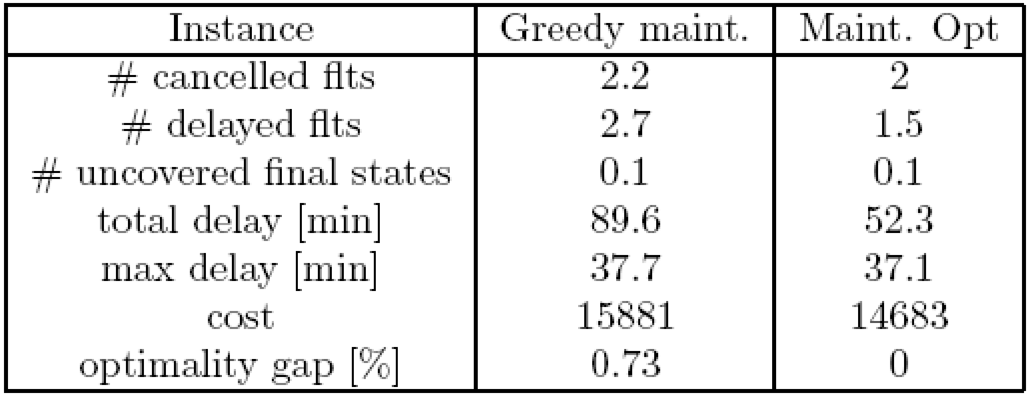
- 10 planes, 147 flights
- Compare 3 approaches :
  - Neglect maintenances: allow resource excess (5, 10 and 20 %)
  - Dummy maintenance: perform maintenance when at least 90% of the resource is consumed
  - Optimize Maintenance using the proposed algorithm

Average Results for 10 randomly generated instances

Solved Instances (3): Added value of maintenances



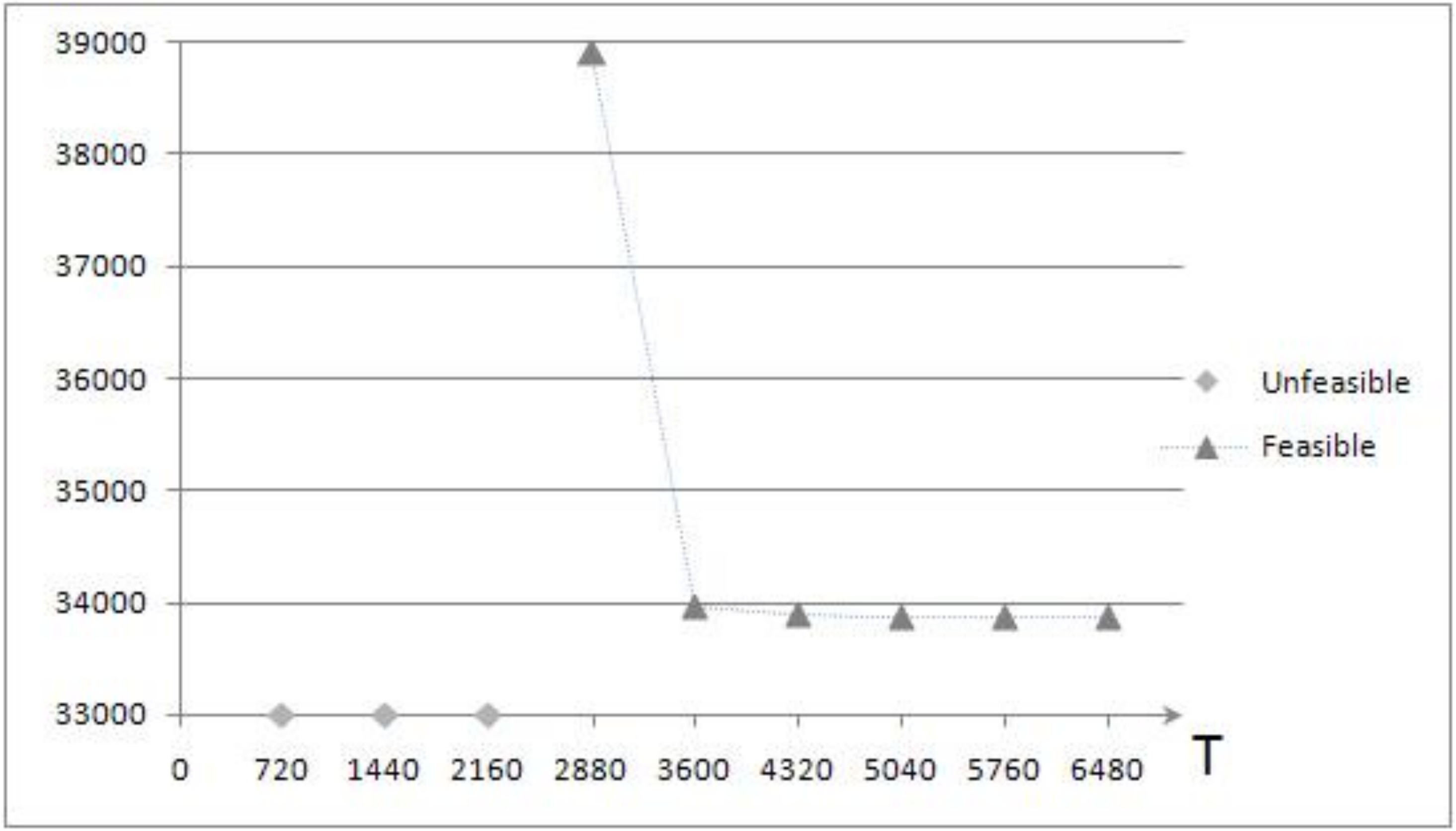
Instance	No maint. + 5%	No maint. + 10%	No maint. + 20%
# cancelled flts	52.7	46.7	33.2
# delayed flts	5	4.7	5.5
# uncovered final states	1.2	0.7	0.3
total delay [min]	851.3	635.7	712.5
max delay [min]	271.3	251.5	218.2
cost	289462	272067	144388
optimality gap [%]	0.61	0.54	1.27



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

Considering maintenances is crucial!!!

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?

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