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*

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

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)

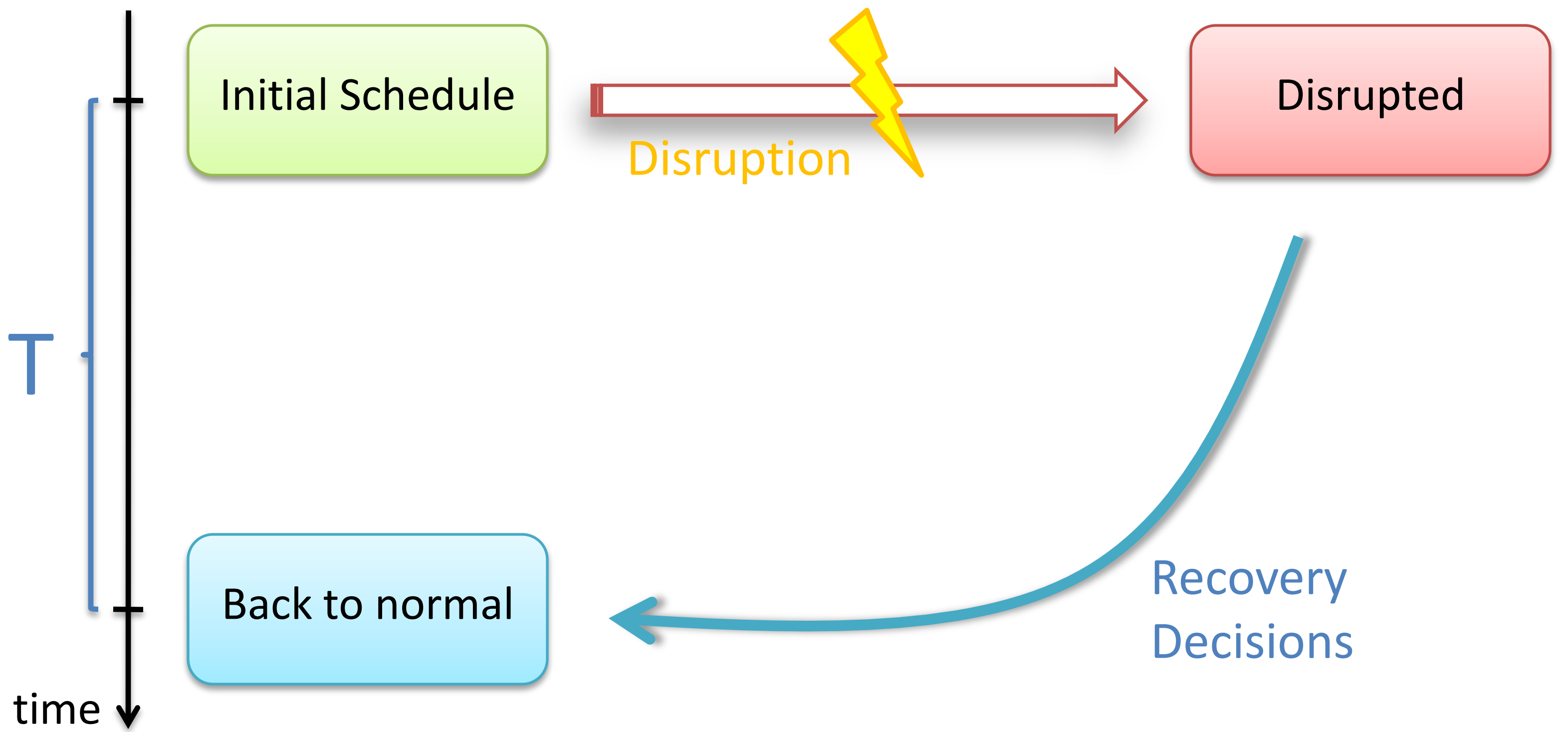
(Items 2–3: Technical Schedule)

Maintenances

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

Resources are renewed during maintenance

Disrupted Schedule and Recovery

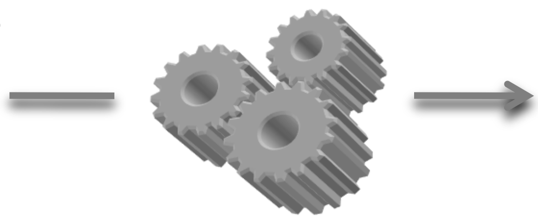


Survey: Kohl (2004)

The Airplane Recovery Problem (ARP)

Input

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



Output

- T
- New schedule up to T
- Recovery cost

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

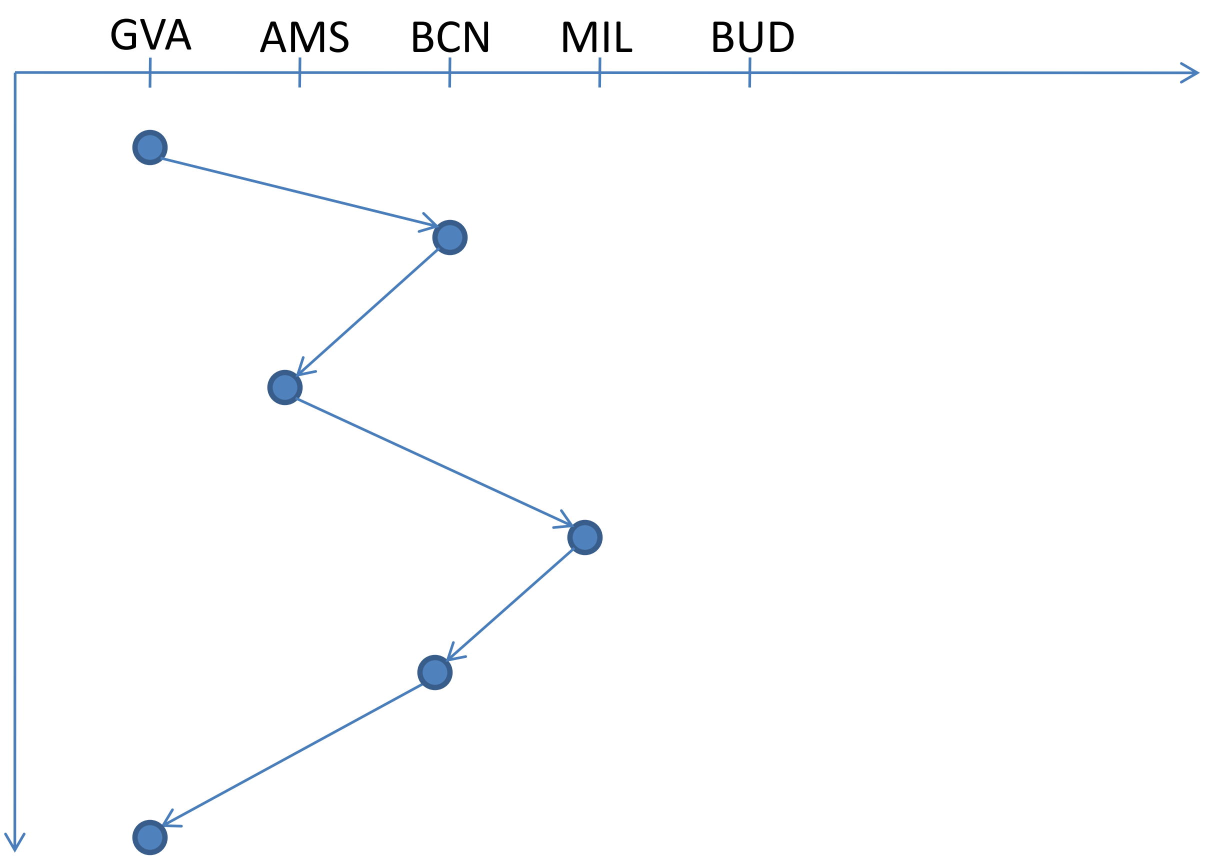
Scheduled Departure Time (SDT)

Flight Duration

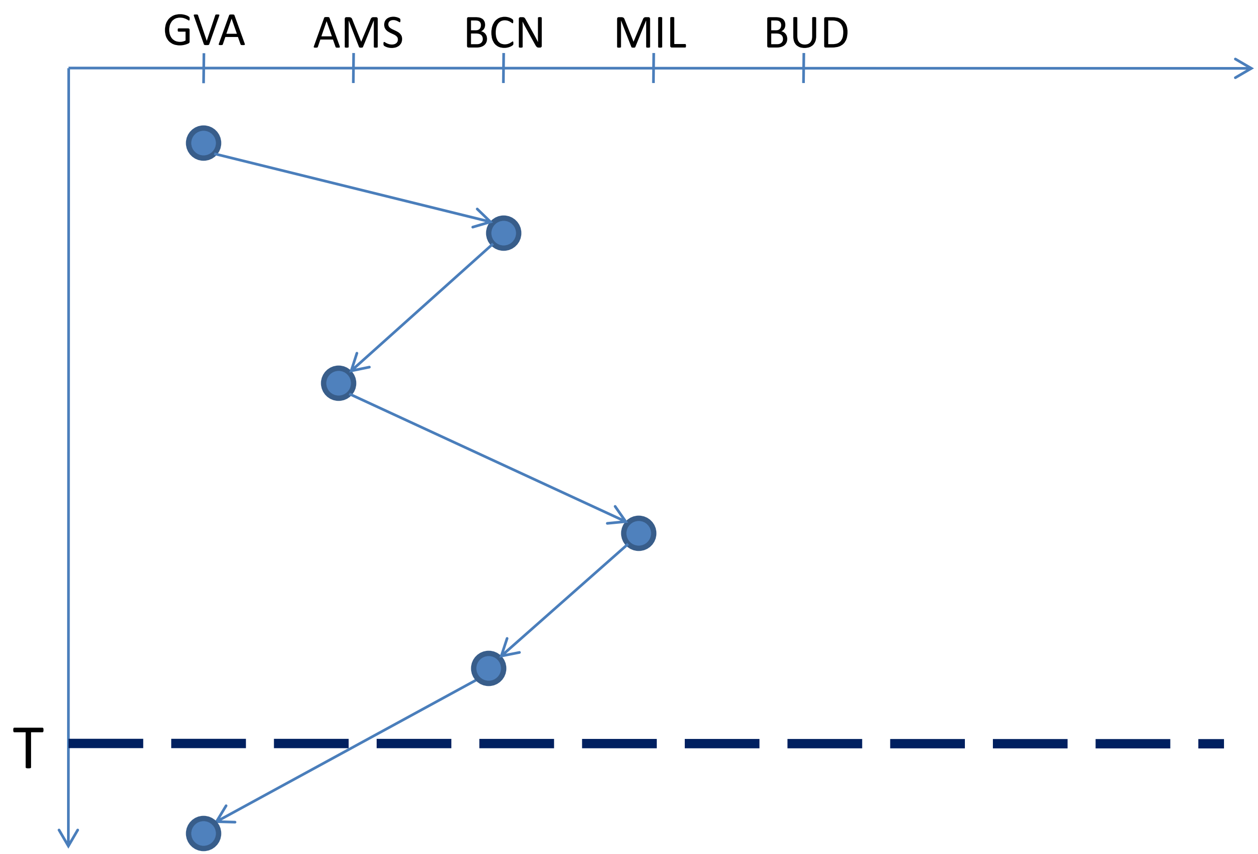
Flight Cost

Cancelation Cost

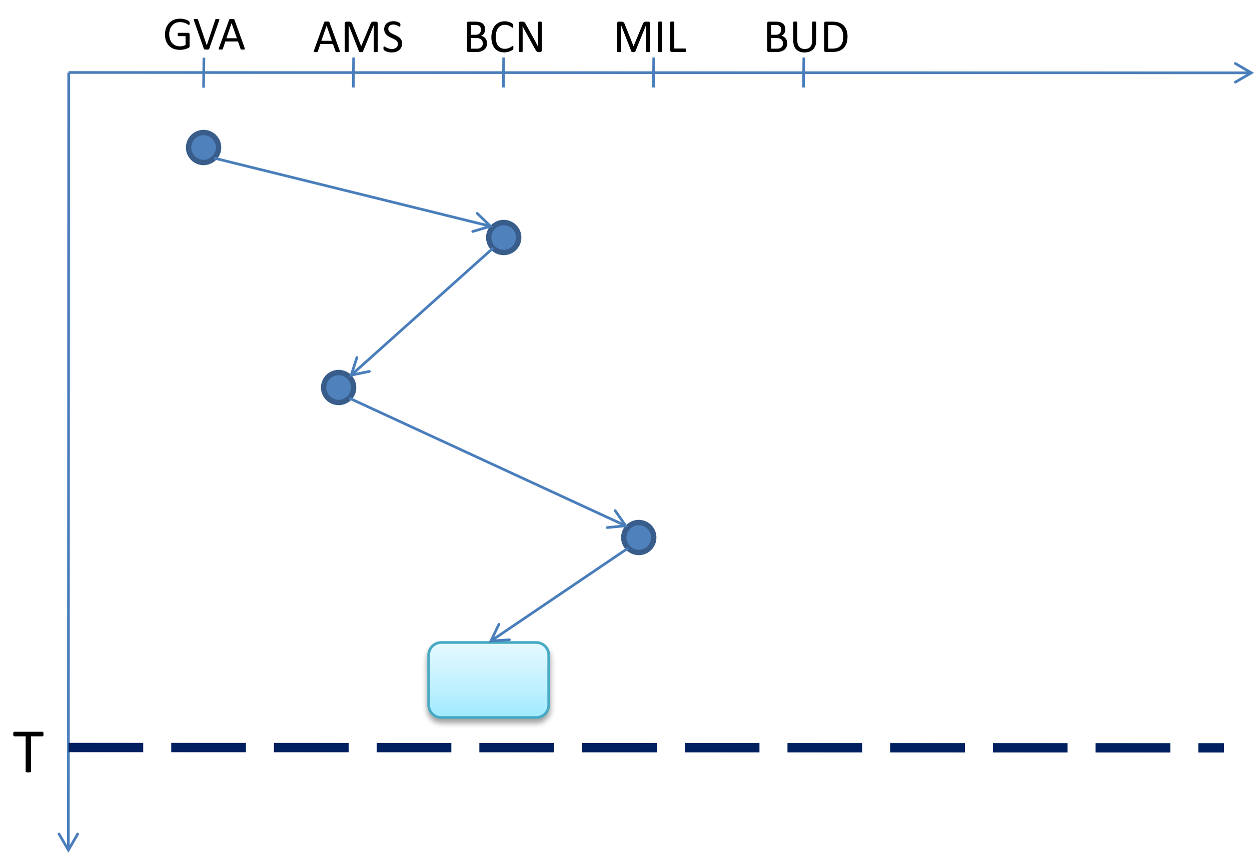
Determine a Final State:



Determine a Final State:

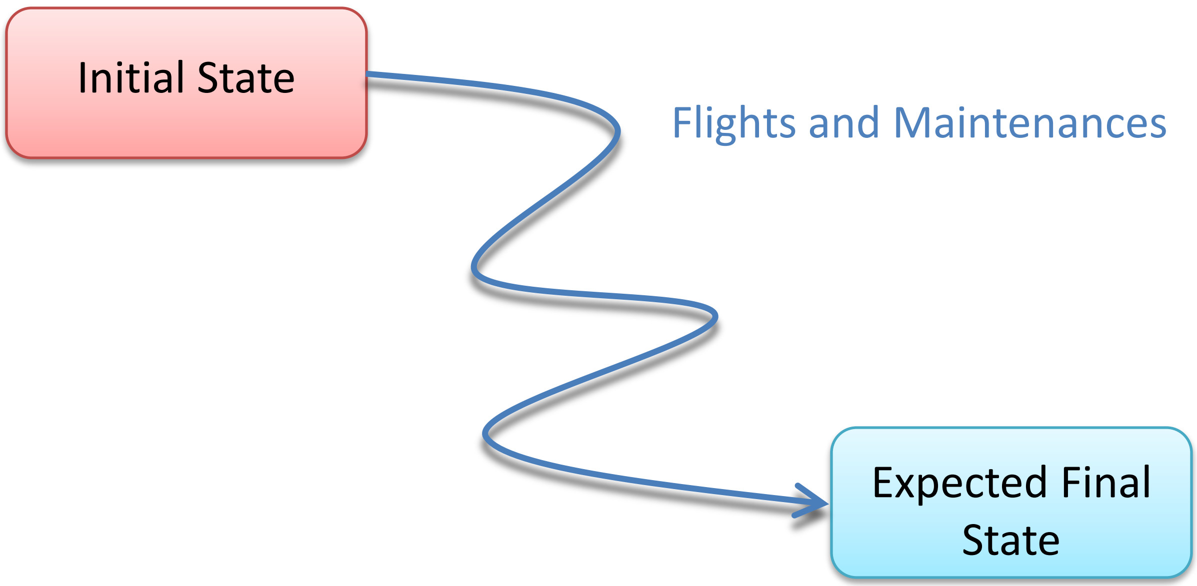


Determine a Final State:



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 $r \in R\text{-}R'$ 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*

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)

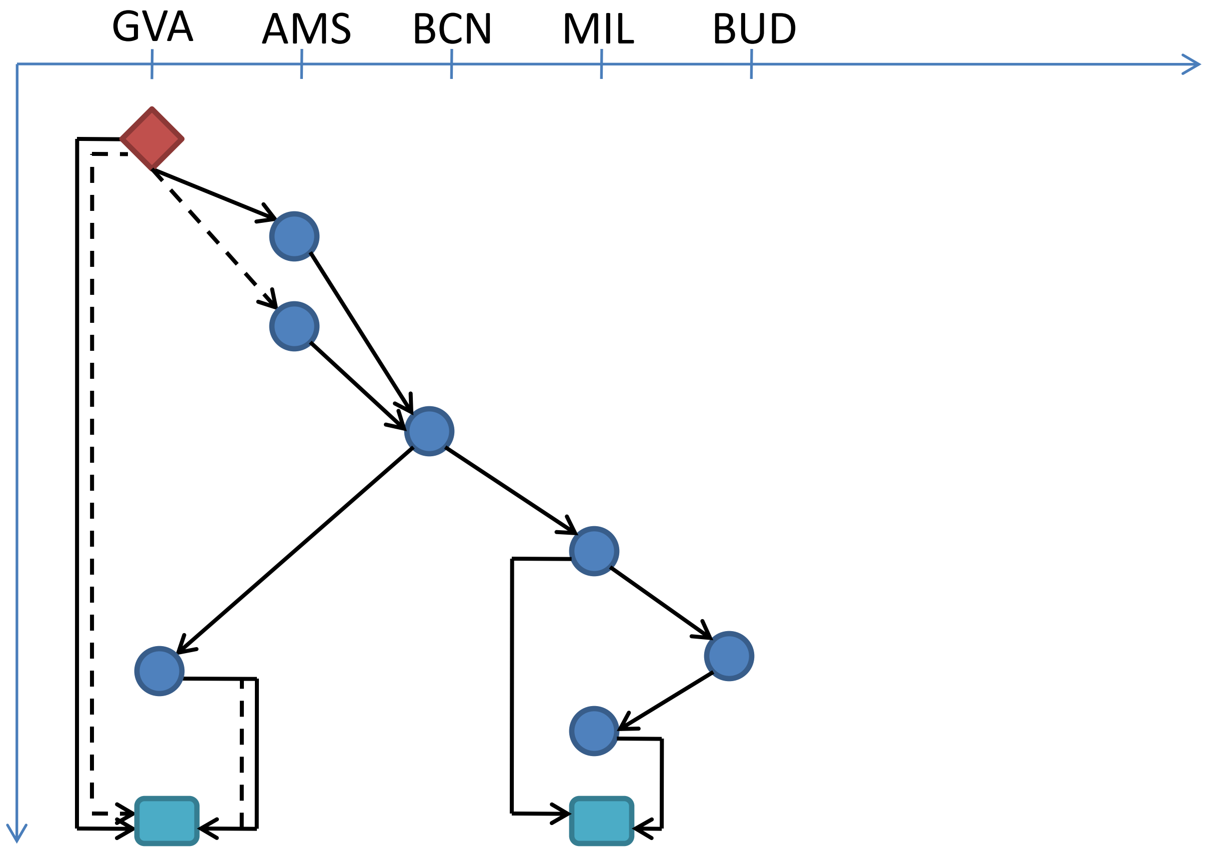
1. Generate a recovery network for each plane
2. Update arc costs according to dual variables
3. Solve Resource Constrained Elementary Shortest Path (RCESPP)
4. Add Columns to *R'*
5. Resolve restricted LP until optimality and branch

Time – Space Network with

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

- 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)

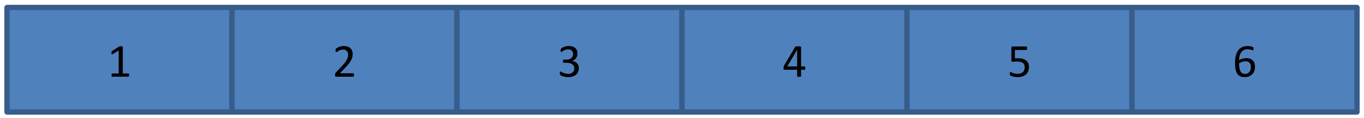
Some References

- 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

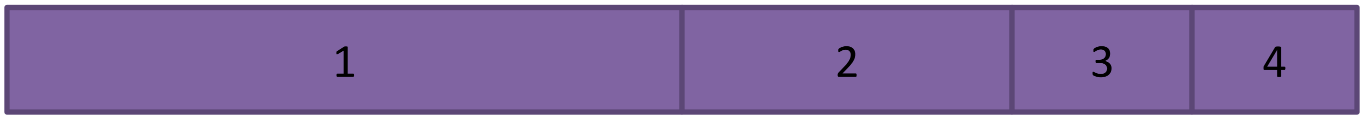
Implementation issues

- 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

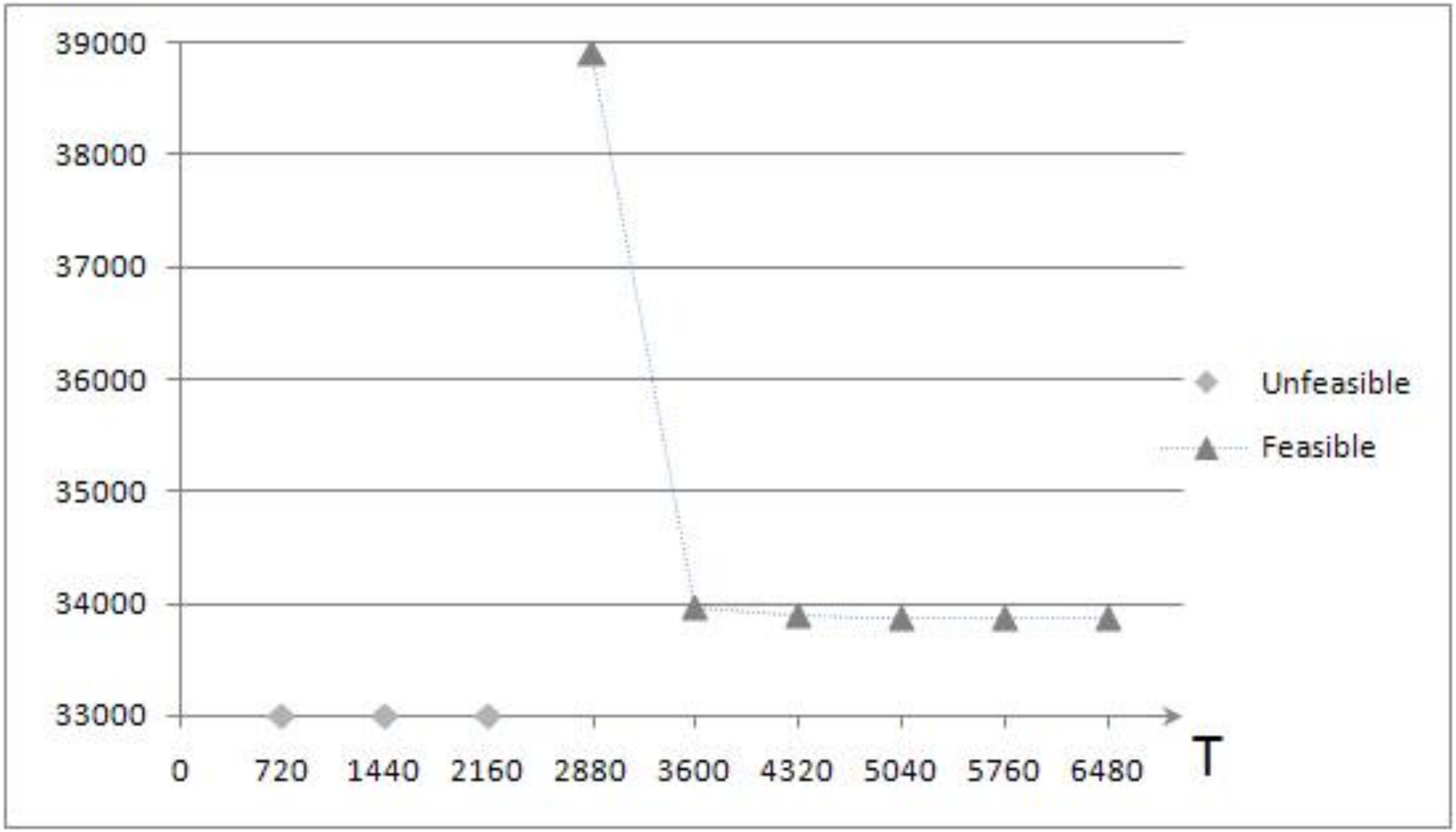
Average results of 10 randomly generated instances

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?