Large scale airline optimization

Dag Wedelin
Chalmers University of Technology
Gothenburg, SWEDEN

Traditional steps in airline scheduling

1. timetable is determined
2. fleet assignment
3. crew pairing
4. crew rostering

Usually done by different departments.

Traditional steps in airline scheduling

1. timetable is determined
2. fleet assignment
3. crew pairing
4. crew rostering

Usually done by different departments.

Clients

AEROMEXICO
AIR FRANCE
Alitalia
BRITISH AIRWAYS
Condor
Die Bahn DB
FINNAIR
green cargo
IBERIA
KLM Royal Dutch Airlines
Lufthansa
MEXICANA
nwa NORTHWEST AIRLINES
Philippine Airlines
SAS
SAS Commuter
SAUDI ARABIAN AIRLINES
SINGAPORE AIRLINES
Spanair
SJ
widerøe

Crew pairing example

Leg number	From	To	Departure	Arrival
1	A	B	6.30	13.30
2	B	A	14.30	21.30
3	B	C	10.15	11.45
4	C	B	12.15	13.45
5	B	C	14.15	15.45
6	C	B	16.15	17.45

Rules:

- A pairing must start and end in A or B
- Max 2 day pairings
- Max duration 12 hours/day
- Layover min 1.5 × duration of first day

PAROS

The two main steps of optimization

1) Model the problem, i.e. define it and formulate it so that a suitable algorithm can accept it.

2) Solve the resulting problem.

Both steps can be very challenging for large and complex real world problems. The steps must be carefully interfaced.

What is an optimization problem?

Example: shortest path problem

Minimize	Minimize
the length	*the objective function*
over	over
all paths from a to b in a graph	*the set of feasible solutions*

Mathematical optimization

Minimize

$x_1 + 3\ x_2 + x_3$

subject to

$x_1 + x_2 = 1$

$x_1 + 2\ x_3 <= 2$

x_i binary

Good standard software available

Standard optimization models in transportation

Polynomial:

- Assignment, transportation, transshipment.
- Single depot vehicle scheduling (SDVSP, scheduling= tasks fixed in time).

(can all be translated into network flow)

More difficult (polynomial or exponential depending on details):

- Multiple depot vehicle scheduling (translates into multicommodity flow).
- Shortest path with time windows and/or resource constraints (use dynamic programming),
- Vehicle routing (VRP), Vehicle routing with time windows (VRPTW), pickup/delivery, … (try greedy+local search or two step approach),

Very difficult to model the rules mathematically.

Rules are defined in an application specific rule language (RAVE).

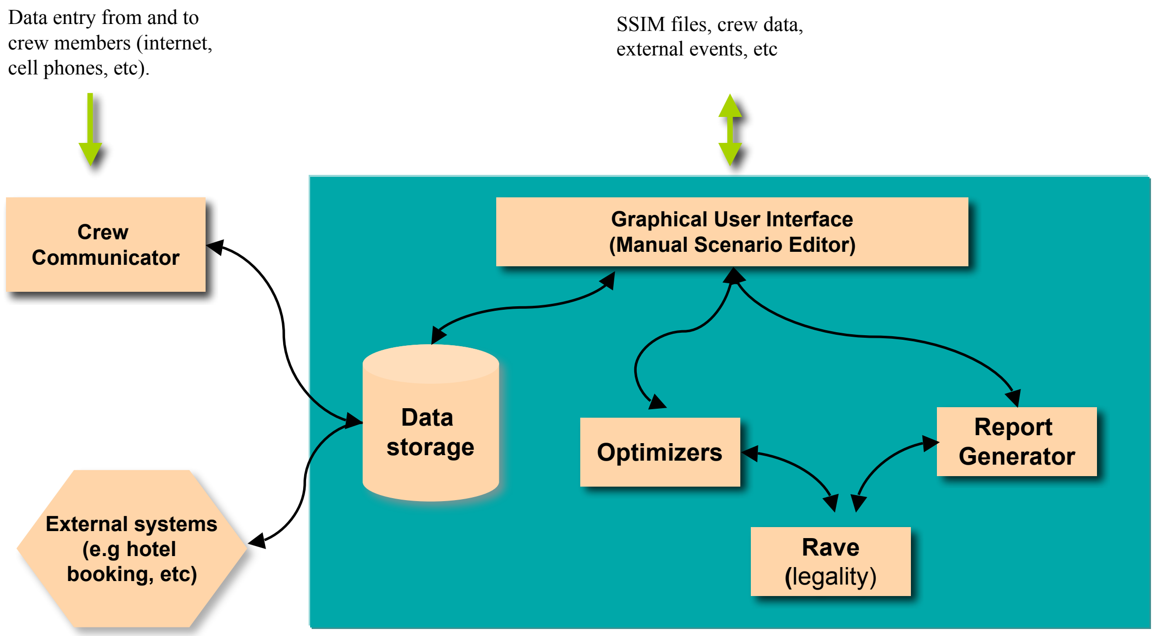
Compiles to test functions in C code.

Rules can be changed and tried out by users anytime!

```
RULE max_duty =
  %duty% <= 9:00;
  REMARK "Maximal
  duty";
END
```

```
rule max_duty_a =

%fdp_length% <= %max_duty_acclimatized%;

remark "Maximum scheduled FDP acclimatized";

end

%fdp_length% = last(leg, arrival) - first(leg, departure) + 2:00;

%max_duty_acclimatized% = matrix m_max_duty_acclimatized;

matrix m_max_duty_acclimatized =

%landings%, %local_checkin% -> %max_duty_acclimatized%;
(0,1), 2,      3,      4,      5, - ;
(06:00,07:59); 12:00, 11:45, 11:00, 10:15, 9:00, 0:00;
(08:00,12:59); 12:00, 12:00, 12:00, 11:45, 10:15,  0:00;
(13:00,17:59); 12:00, 12:00, 11:30, 10:45,  9:15, 0:00;
(18:00,21:59); 12:00, 11:15, 10:30,  9:45, 9:00,  0:00;
(22:00,05:59); 11:00, 10:15,  9:30,  9:00, 9:00,  0:00;

end
```

In this code, the rule **%max_duty_a%** specifies that the expression **%fdp_length%** must be less than the expression **%max_duty_acclimatized%**. These expressions are then defined, **%fdp_length%** by accessing some data in the line of work, and **%max_duty_acclimatized%** by the use of a "matrix", whose (scalar) value is determined by table lookup based on the values of **%landings%** and **%local_checkin%**.



Standard methods for solving transport optimization problems

1. Time-space networks and network flow, or special cases thereof (assignment, transportation, transshipment)
2. Construction heuristics with greedy and local search
3. Two-step approach for general task scheduling: generate separate paths first, then optimize how to combine them. Column generation

Attention to small details and special properties of the problem at hand is cruical for best performance.

Crew pairing example

Leg number	From	To	Departure	Arrival
1	A	B	6.30	13.30
2	B	A	14.30	21.30
3	B	C	10.15	11.45
4	C	B	12.15	13.45
5	B	C	14.15	15.45
6	C	B	16.15	17.45

Rules:

- A pairing must start and end in A or B
- Max 2 day pairings
- Max duration 12 hours/day
- Layover min 1.5 × duration of first day

PAROS

Enumeration of all legal pairings

P1	1-lo-2 (2)	P7	5-6 (1)	P13	3-lo-4-5-6 (2)
P2	1-lo-3-4-2 (2)	P8	3-4-5-6 (1)	P14	3-4-5-lo-4 (2)
P3	1-5-lo-4-2 (2)	P9	2-lo-1 (2)	P15	3-4-5-lo-4-5-6
P4	1-5-6-lo-2 (2)	P10	2-lo-1-5-6 (2)	P16	5-lo-4 (2)
P5	3-4 (1)	P11	3-lo-4 (2)	P17	5-lo-4-5-6 (2)
P6	3-6 (1)	P12	3-lo-6 (2)		

lo: layover

PAROS

The resulting optimization problem

$$
\begin{aligned}
\text{minimize} \quad & 2x_1 + 2x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 + x_8 + 2x_9 + 2x_{10} \\
& +2x_{11} + 2x_{12} + 2x_{13} + 2x_{14} + 2x_{15} + 2x_{16} + 2x_{17} \\
\text{subject to} \quad & \\
& x_1 + x_2 + x_3 + x_4 + x_9 + x_{10} = 1 \\
& x_1 + x_2 + x_3 + x_4 + x_9 + x_{10} = 1 \\
& x_2 + x_5 + x_6 + x_8 + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1 \\
& x_2 + x_3 + x_5 + x_8 + x_{11} + x_{13} + 2x_{14} + 2x_{15} + x_{16} + x_{17} = 1 \\
& x_3 + x_4 + x_7 + x_8 + x_{10} + x_{13} + x_{14} + 2x_{15} + x_{16} + 2x_{17} = 1 \\
& x_4 + x_6 + x_7 + x_8 + x_{10} + x_{12} + x_{13} + x_{15} + x_{17} = 1
\end{aligned}
$$

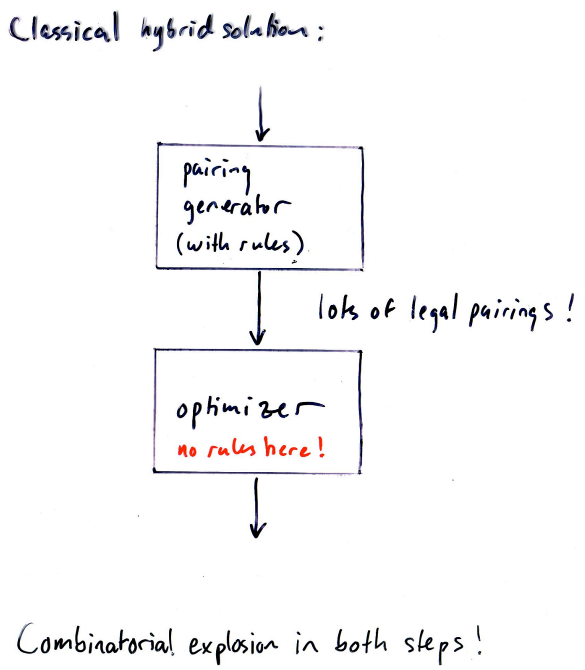
$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17} \in \{0, 1\}$$

PAROS

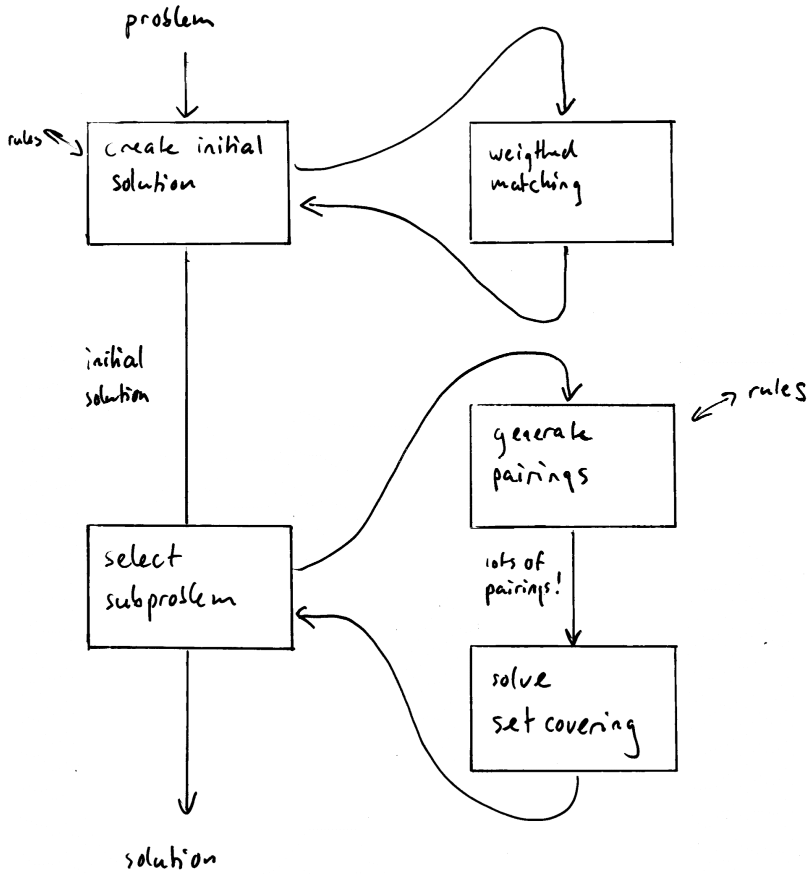
Crew scheduling

Main design conflict:
rule modelling versus *optimization*

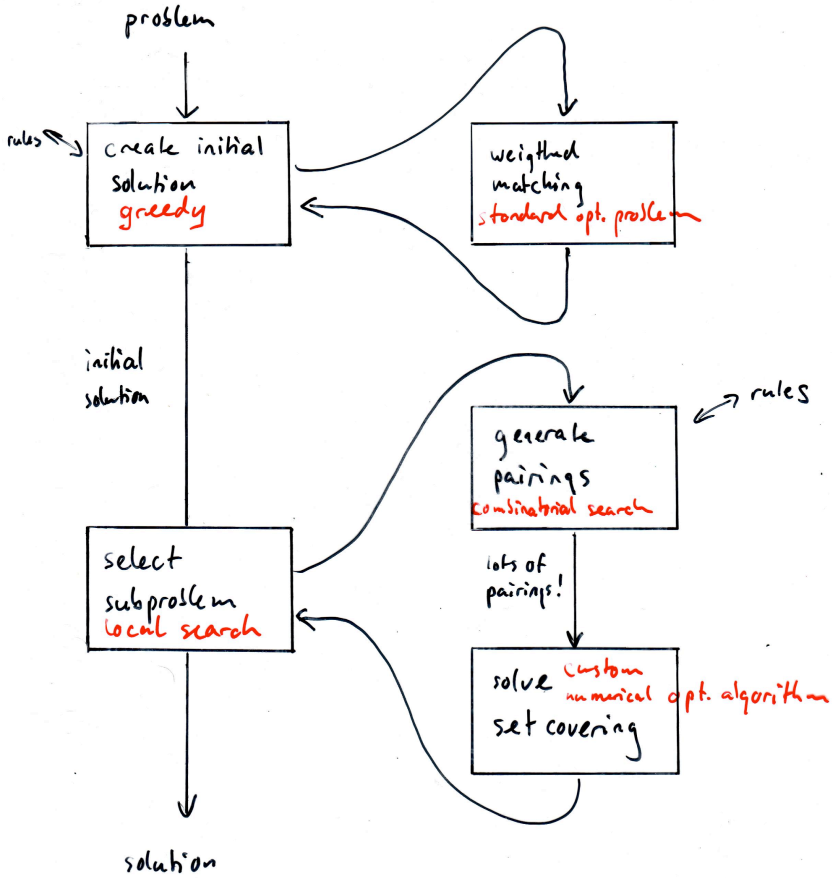
Try to separate them!



Pairing optimization overview



Algorithm design techniques



How solve the large problems?

- Selection of daily subproblems.
- Generator search heuristics. E.g. generate pairings that follow the aircraft.
- Special purpose algorithm for the large optimization problems (10^6 variables, 10^4 constraints).
- Heuristic solution to deadheading

More recent developments

Send feedback from optimizer to generator.

The idea is to get new pairings that may improve current solution (solve constrained shortest path problem in generator).

Nice mathematical theory but tricky in practice.

A challenge to integrate with the separate rule system.

The set covering optimizer

Solves the problem

$$\min \; cx$$
$$Ax \geq 1$$
$$Cx \leq d \quad \text{(base capacity constraints)}$$

Solved using numerical algorithms:

- CPLEX for small and medium sized problems
- probl or paqs for large problems

  (developed at Chalmers!)

The simple assignment problem

$$\min \sum_i \sum_j c_{ij} x_{ij}$$

subject to

$$\sum_j x_{ij} = 1$$

$$\sum_i x_{ij} = 1$$

$$x_{ij} \in \{0, 1\}$$

tasks

persons

5	6	2	9
4	6	7	1
1	5	3	6
9	2	5	5

Some observations

Observation 1: some instances are easy to solve just by inspecting the cost matrix

4	-3	11	2
5	2	-8	7
-6	9	2	4
3	3	4	-1

Observation 2: the problem is invariant with respect to additions or subtractions of any constant to a row or column

Simple idea: use the invariants to make the problem easy!

Operational description

ex assignment problem

old $\bar{c}$

5	2	9	7
6	1	11	8
1	9	5	3
4	5	0	6

r^+ r^-

$\Rightarrow$

new $\bar{c}$

5	2	9	7
6	1	11	8
1	9	5	3
2	3	-2	4

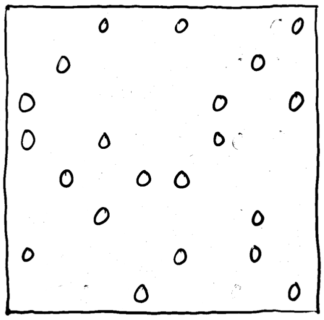
Look at each constraint separately and choose

$$y_i = -\frac{r^+ + r^-}{2}$$

Iterate!

What happens when we run this simple algorithm?

But for less trivial problems we get lots of zeros in $\bar{c}$:



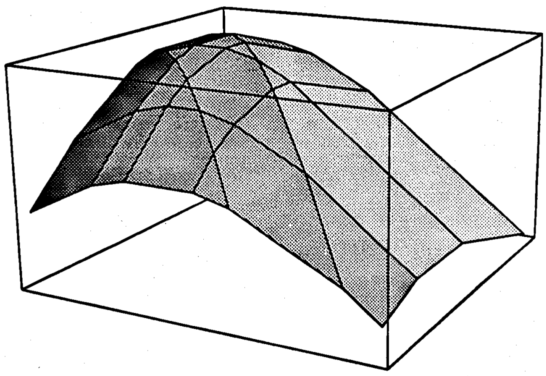
eg. 8-queens with diagonal constraints

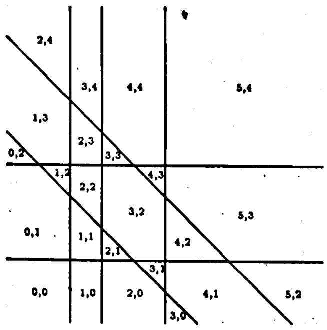
For simple assignment it works fine!

No unique solution => useless!

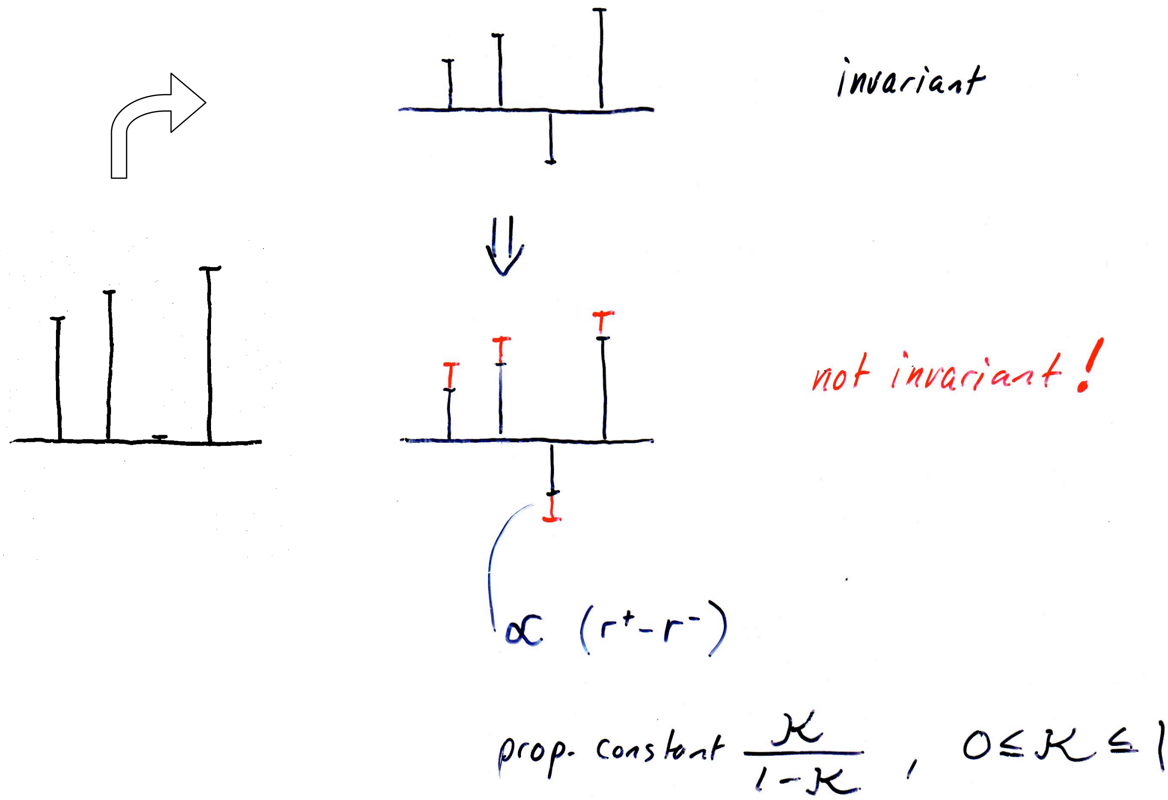
What shall we do?

Coordinate search in dual space





The heuristic



Now it suddenly works!

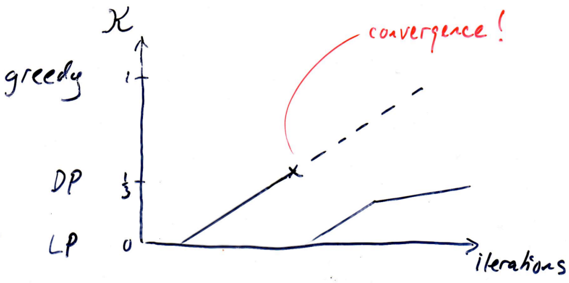
(weighted 8-queens problem)

1	3	7	5	8	3	2	9
5	3	8	7	5	1	1	9
4	3	7	5	8	2	3	8
7	1	2	4	1	4	6	2
2	6	2	8	4	4	4	1
3	5	4	5	4	5	4	5
1	2	3	4	5	6	7	8
3	2	2	2	2	2	6	8

0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0

Sweep mechanism

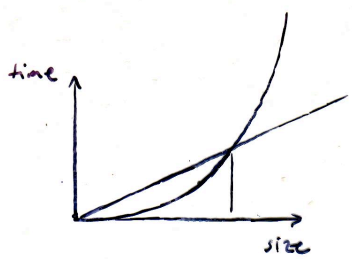
$\mathcal{K} > 0$ gives fast convergence and usually a high quality integer solution!



Interpolation between different algorithmic design principles!

No exponential blowup ⇒ very large problems ok!

Performance



linear

size after preprocessing			CPLEX	our method
rows	columns	nonzero		
29	157	372	0.2s	1.4s
118	1165	4155	1s	11s
199	6931	31801	23s	1m4s
235	18753	82229	2m12s	5m2s
742	10370	42695	23m	3m58s
1366	25032	110881	5h53m	7m46s
1585	105804	566905	-	36m10s

same quality!

CPLEX	our method
92800	92800
5017500	5017500
529250	529250
565000	565000
4737768	4737892
4335780	4333450
-	4329750

Same general picture supported by regular benchmarking (2004)

Client Statement

“Carmen is one of the best investments we have ever made”

British Airways

Optimization trends

10 years ago - an overnight run,
today - within seconds.

Combined effect of faster algorithms and faster computers.

This makes it possible to:

improve modelling detail (gives larger problems)

integrate planning steps (gives larger problems)

approach day-of-operation planning (requires faster optimizers)

Carmen day of operation tools