Optimizing crew pairing in an airline

Frédéric Meunier

June 8th, 2018

Joint work with Axel Parmentier

Crew in an airline

The crew pairing problem consists in generating minimum-cost, multiple-day work schedules.

It has to be carefully addressed since (Barnhart et al., 2003)

> *Crews represent the airlines' second highest operating cost after fuel, so even slight improvements in their utilization can translate into significant savings.*

Rough description of the problem

- 6 months before the realization of the flights, the crew pairing problem has to be solved.

- Set $\mathcal{F}$ of all flight legs are fixed beforehand.

- Sequence of flight legs realized by a crew: pairing.

- A pairing must satisfy many constraints about rest, time spent in a flying airplane, etc. (more than 70 working rules have to be satisfied).

- Each pairing has a cost (wage, indemnity, etc.).

Crew pairing problem

Find a partition of $\mathcal{F}$ into feasible pairings so as to minimize the total cost.

This talk

Method for solving the crew pairing problem

- developed within a collaboration with AIR FRANCE.
- able to solve almost all industrial instances to near-optimality.
- relies on improved shortest path algorithms.

Plan.

1. The problem
2. A column generation model
3. Shortest paths
4. Experimental results

The crew pairing problem

Feasible pairings

Pairing: sequence $p = (\ell_1, \ldots, \ell_k)$ of flight legs to be operated by a crew.

To be feasible, p must satisfy connection constraints

- ℓ_i must end in the airport from which ℓ_{i+1} departs.
- departure_time(ℓ_{i+1}) − arrival_time$(\ell_i) \geqslant \tau$, where τ is a predetermined duration.
- ℓ_1 must start in Paris and ℓ_k must end in Paris.

It has also to satisfy all working rules

- total number of days spanned by p,
- total flying duration in p,
- rest,
- etc.

The crew pairing problem

Input.

- Set $\mathcal{L}$ of all flight legs realized by the airplanes, over a one-week horizon
- For each pair (ℓ, ℓ') of flight legs satisfying connection constraints: cost $c_{\ell,\ell'}$ of chaining ℓ with ℓ' in a pairing
- Working rules

Output.

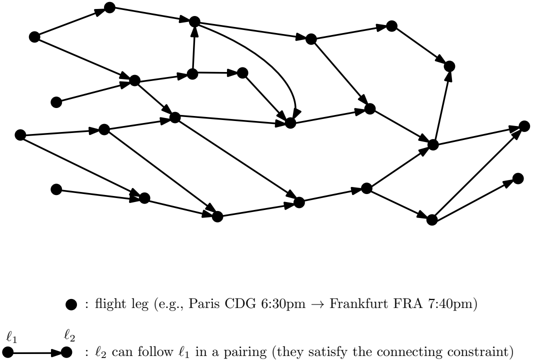
- A partition of $\mathcal{L}$ into feasible pairings with minimal total cost

  (total cost $=$ sum of the costs of the pairings in the partition)

The crew pairing problem – a graphic formulation

Input.

- Digraph $(\mathcal{L}, A)$, where $(\ell, \ell') \in A$ means that they satisfy connection constraints
- For each arc $a = (\ell, \ell') \in A$, cost c_a giving the cost of chaining ℓ with ℓ' in a pairing
- Working rules



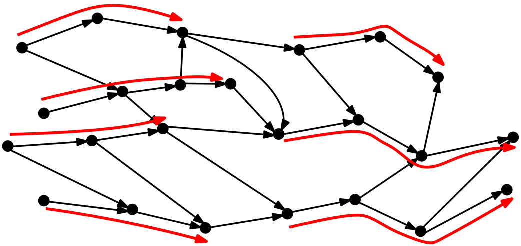
Output.

- A partition of $\mathcal{L}$ into "feasible" paths with minimal total cost

The crew pairing problem – a graphic formulation

Input.

- Digraph $(\mathcal{L}, A)$, where $(\ell, \ell') \in A$ means that they satisfy connection constraints
- For each arc $a = (\ell, \ell') \in A$, cost c_a giving the cost of chaining ℓ with ℓ' in a pairing
- Working rules



● : flight leg (e.g., Paris CDG 6:30pm → Frankfurt FRA 7:40pm)

$\ell_1 \longrightarrow \ell_2$: ℓ_2 can follow ℓ_1 in a pairing (they satisfy the connecting constraint)

Output.

- A partition of $\mathcal{L}$ into "feasible" paths with minimal total cost

Column generation model

Integer programming model

Standard Operations Research approach:

write the problem as

$$
\begin{aligned}
\text{Min} \quad & \sum_{p \in \mathcal{P}} c_p x_p \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}\,:\; p \ni \ell} x_p = 1 && \forall \ell \in \mathcal{L} \\
& x_p \in \{0, 1\} && \forall p \in \mathcal{P}
\end{aligned}
$$

where

- $\mathcal{P}$ is the set of all feasible pairings
- $c_p = \sum_{a \in p} c_a$

$p \in \mathcal{P} \iff$ p satisfies the connection constraints and the working rules

Method

Step 1. SOLVE (LR($\mathcal{P}$)) via column generation

Step 2. USE the optimal solution of (LR($\mathcal{P}$)) to discard pairings that are in no optimal solution of the original integer program: $\mathcal{P}^* \subseteq \mathcal{P}$

Step 3. SOLVE original integer program on $\mathcal{P}^*$ directly with any standard IP solver

When $\mathcal{P}^*$ is small, last step is doable.

$$\begin{array}{lll} \text{Min} & \displaystyle\sum_{p \in \mathcal{P}} c_p x_p & \\ \text{s.t.} & \displaystyle\sum_{p \in \mathcal{P}\,:\, p \ni \ell} x_p = 1 & \forall \ell \in \mathcal{L} \\ & 0 \leqslant x_p \leqslant 1 & \forall p \in \mathcal{P} \end{array} \qquad (\text{LR}(\mathcal{P}))$$

$$\downarrow$$

$$\downarrow$$

$$\begin{array}{lll} \text{Min} & \displaystyle\sum_{p \in \mathcal{P}^*} c_p x_p & \\ \text{s.t.} & \displaystyle\sum_{p \in \mathcal{P}^*\,:\, p \ni \ell} x_p = 1 & \forall \ell \in \mathcal{L} \\ & x_p \in \{0, 1\} & \forall p \in \mathcal{P}^* \end{array}$$

(Standard method relies on column generation and branch-and-bound)

Step 1. Solve linear relaxation via column generation

$$\begin{aligned} \text{Min} \quad & \sum_{p \in \mathcal{P}'} c_p x_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}' \,:\, p \ni \ell} x_p = 1 \quad \forall \ell \in \mathcal{L} \\ & 0 \leqslant x_p \leqslant 1 \qquad \forall p \in \mathcal{P}' \end{aligned} \qquad (\mathrm{LR}(\mathcal{P}'))$$

Column generation:

INITIALIZE with a small $\mathcal{P}'$ so that $(\mathrm{LR}(\mathcal{P}'))$ is feasible;
REPEAT
    SOLVE $(\mathrm{LR}(\mathcal{P}'))$ with any standard solver;
    FIND a pairing $p \in \mathcal{P}$ of minimum reduced cost $\tilde{c}_p$;
    IF $(\tilde{c}_p < 0)$
    THEN add p to $\mathcal{P}'$;
UNTIL $(\tilde{c}_p \geqslant 0$ for all $p \in \mathcal{P})$

Step 1. Solve linear relaxation via column generation

$$
\begin{aligned}
\text{Min} \quad & \sum_{p \in \mathcal{P}'} c_p x_p \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}' :\ p \ni \ell} x_p = 1 \quad \forall \ell \in \mathcal{L} \\
& 0 \leqslant x_p \leqslant 1 \qquad \forall p \in \mathcal{P}'
\end{aligned}
\qquad (\mathrm{LR}(\mathcal{P}'))
$$

Column generation:

INITIALIZE with a small $\mathcal{P}'$ so that $(\mathrm{LR}(\mathcal{P}'))$ is feasible;
REPEAT
    SOLVE $(\mathrm{LR}(\mathcal{P}'))$ with any standard solver;
    FIND a pairing $p \in \mathcal{P}$ of minimum reduced cost $\tilde{c}_p$;   (pricing subproblem)
    IF $(\tilde{c}_p < 0)$
    THEN add p to $\mathcal{P}'$;
UNTIL $(\tilde{c}_p \geqslant 0$ for all $p \in \mathcal{P})$

Method

Step 1. SOLVE (LR($\mathcal{P}$)) via column generation

Step 2. USE the optimal solution of (LR($\mathcal{P}$)) to discard pairings that are in no optimal solution of the original integer program: $\mathcal{P}^* \subseteq \mathcal{P}$

Step 3. SOLVE original integer program on $\mathcal{P}^*$ directly with any standard IP solver

Step 2. Use optimal solution to discard pairings

Step 1. has produced

- a subset $\mathcal{P}'$ on which linear relaxation is optimal.
- a lower bound lb of the original integer program.

SOLVE original integer program on $\mathcal{P}'$ produces

- a feasible solution of the original integer program.
- an upper bound ub of the original integer program.

DETERMINE all pairings p for which $\tilde{c}_p \leqslant \text{ub} - \text{lb}$ and ADD them to $\mathcal{P}'$:

- produces a subset $\mathcal{P}^* \subseteq \mathcal{P}$.

This last step is again a “pricing subproblem”.

Step 2. Rationale

$$\begin{array}{rl} \text{Min} & \boldsymbol{c} \cdot \boldsymbol{z} \\ \text{s.t.} & A\boldsymbol{z} = \boldsymbol{b} \\ & \boldsymbol{z} \in \{0,1\}^n \end{array} \qquad \text{(P)}$$

lb $=$ optimal value of lin. rel.

$\tilde{c}_i =$ reduced cost of ith variable in the optimal solution of lin. rel.

ub $=$ value of some feasible solution

Lemma

If $\tilde{c}_i > \text{ub} - \text{lb}$, *then* $z_i = 0$ *in every optimal solution of* (P).

Method

Step 1. SOLVE (LR($\mathcal{P}$)) via column generation

Step 2. USE the optimal solution of (LR($\mathcal{P}$)) to discard pairings that are in no optimal solution of the original integer program: $\mathcal{P}^* \subseteq \mathcal{P}$

Step 3. SOLVE original integer program on $\mathcal{P}^*$ directly with any standard IP solver

Step 3. Solve original integer program on $\mathcal{P}^*$

SOLVE

$$\begin{aligned} \text{Min} \quad & \sum_{p \in \mathcal{P}^*} c_p x_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}^* \,:\, p \ni \ell} x_p = 1 \qquad && \forall \ell \in \mathcal{L} \\ & x_p \in \{0, 1\} && \forall p \in \mathcal{P}^* \end{aligned}$$

with any standard solver.

$\mathcal{P}^*$ has to be very small ($\mathcal{P}^* \ll \mathcal{P}$) for the method to work.

In particular, ub $-$ lb has to be small.
(Luckily, this is the case for this problem.)

Method

Step 1. SOLVE (LR($\mathcal{P}$)) via column generation

Step 2. USE the optimal solution of (LR($\mathcal{P}$)) to discard pairings that are in no optimal solution of the original integer program: $\mathcal{P}^* \subseteq \mathcal{P}$

Step 3. SOLVE original integer program on $\mathcal{P}^*$ directly with any standard IP solver

$$\begin{aligned} \text{Min} \quad & \sum_{p \in \mathcal{P}} c_p x_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}\,:\, p \ni \ell} x_p = 1 && \forall \ell \in \mathcal{L} \\ & 0 \leqslant x_p \leqslant 1 && \forall p \in \mathcal{P} \end{aligned}$$

$$\downarrow$$

$$\downarrow$$

$$\begin{aligned} \text{Min} \quad & \sum_{p \in \mathcal{P}^*} c_p x_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}^*\,:\, p \ni \ell} x_p = 1 && \forall \ell \in \mathcal{L} \\ & x_p \in \{0, 1\} && \forall p \in \mathcal{P}^* \end{aligned}$$

Shortest paths

Pricing subproblem

Only step for which we have flexibility: pricing subproblem.

REPEAT
  SOLVE (LR($\mathcal{P}'$)) with any standard solver;
  FIND a pairing $p \in \mathcal{P}$ of minimum reduced cost $\tilde{c}_p$;
  IF ($\tilde{c}_p < 0$)
  THEN add p to $\mathcal{P}'$;
UNTIL ($\tilde{c}_p \geqslant 0$ for all $p \in \mathcal{P}$)

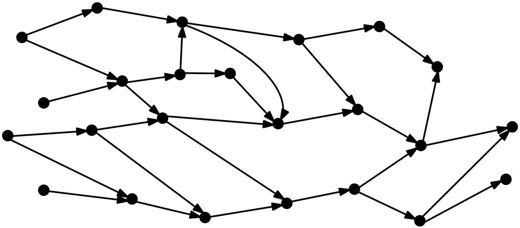
$$\begin{aligned} \text{Min} \quad & \sum_{p \in \mathcal{P}'} c_p x_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}' \,:\, p \ni \ell} x_p = 1 && \forall \ell \in \mathcal{L} \\ & 0 \leqslant x_p \leqslant 1 && \forall p \in \mathcal{P}' \end{aligned} \qquad (\mathrm{LR}(\mathcal{P}'))$$

Desrosiers and Lübbecke (2006)

> *"[in a column generation context] accelerating the pricing algorithm itself usually leads most significant speeds-up."*

Pricing subproblem

$\text{Min}_{p \in \mathcal{P}} \tilde{c}_p$, consists in finding a “shortest” path p in the digraph $(\mathcal{L}, A)$:



● : flight leg (e.g., Paris CDG 6:30pm → Frankfurt FRA 7:40pm)

$\ell_1 \longrightarrow \ell_2$: ℓ_2 can follow ℓ_1 in a pairing (they satisfy the connecting constraint)

$\tilde{c}_p$ being “additive” (it is of the form $\sum_{a \in p} \overline{c}_a$), this a standard problem,

... except that $p \in \mathcal{P}$ means p has to satisfy about 70 working rules!

Pricing subproblem and ordered monoid

It is possible to

- *define a monoid* $(M, \oplus)$, *with a partial order* $\preccurlyeq$,
- *assign to each arc a of* $D = (\mathcal{L}, A)$ *an element* m_a *of* M *(its resource)*
- *define an non-decreasing oracle* $\rho \colon M \to \{0, 1\}$

such that

$$p \in \mathcal{P} \qquad \Longleftrightarrow \qquad \rho\left(\bigoplus_{a \in p} m_a\right) = 0.$$

Example

Example of working rules:

1. Flying duration per day $\leqslant F$
2. max. 4 flight legs per day if the preceding night rest is long, and 3 otherwise.

Resource of an arc $a = (\ell, \ell')$, with f flying duration of ℓ'

- If both ℓ and ℓ' are on a same day: $m_a = (1, f)$
- If there is a night between ℓ and ℓ':
  $$m_a = \begin{cases} (0,0,2,f) & \text{if long night rest} \\ (0,0,1,f) & \text{if short night rest} \end{cases}$$

$$(n,f) \oplus (\tilde{n},\tilde{f}) = (n+\tilde{n}, f+\tilde{f})$$
$$(n,f) \oplus (\tilde{n}^b,\tilde{f}^b,\tilde{n}^e,\tilde{f}^e) = (n+\tilde{n}^b, f+\tilde{f}^b, \tilde{n}^e, \tilde{f}^e)$$
$$(n^b,f^b,n^e,f^e) \oplus (\tilde{n},\tilde{f}) = (n^b, f^b, n^e+\tilde{n}, f^e+\tilde{f})$$
$$(n^b,f^b,n^e,f^e) \oplus (\tilde{n}^b,\tilde{f}^b,\tilde{n}^e,\tilde{f}^e) = (n^b, f^b, \tilde{n}^e, \tilde{f}^e)$$

$\rho(n,f) := 1$
if $n > 4$ or $f > F$

$\rho(n^b,n^e,f^b,f^e) := 1$
if $n^b > 4$ or $n^e > 4$ or $f^b > F$ or $f^e > F$

Pricing subproblem as a shortest path problem

The pricing subproblem is thus

Input.

- Digraph $D = (\mathcal{L}, A)$ with extra origin o and destination d
- Elements $\overline{c}_a \in \mathbb{R}$ and $m_a \in M$ for each $a \in A$
- A non-decreasing oracle $\rho\colon M \to \{0, 1\}$

Output.

- An o-d path p satisfying $\rho\left(\bigoplus_{a\in p} m_a\right) = 0$ while minimizing $\sum_{a\in p} \overline{c}_a$

Axel has developed in his PhD thesis an efficient methodology to deal with this kind of problems.

Methodology for the shortest path problems

Oriented graph $(\mathcal{L} \cup \{o, d\}, A)$.

Precompute quickly a bound $b_\ell \in M$ for each $\ell \in \mathcal{L}$

$$b_\ell \preccurlyeq \bigoplus_{a \in q} m_a. \qquad \forall \ell\text{-}d \text{ path } q.$$

Algorithm enumerates (implicitly) all paths by starting at o and extending them arc by arc.

Implicitly: an o-ℓ path p is discarded if

- “Bound”: $\rho(p \oplus b_\ell) = 1$ (and similarly for the cost)
- “Dominance”: an o-ℓ path p' is currently considered with $\bigoplus_{a \in p'} m_a \preccurlyeq \bigoplus_{a \in p} m_a$.

Numerical experiments

Numerical results

128 Gb of RAM and 12 cores at 2.4 GHz.

Linear and integer programs are solved with CPLEX 12.1.0.

Instance	Legs	Crew connect	Pricing time	LP time	MIP time	Total time (hh:mm:ss)
A318	669	3,742	86.60%	13.34%	0.05%	01:21:22
A319	957	3,738	60.66%	39.14%	0.15%	00:10:47
A320	918	3,813	74.54%	25.20%	0.20%	00:08:35
A321	778	3,918	65.82%	32.60%	1.25%	00:33:51
A318-9	1,766	8,070	69.71%	30.21%	0.07%	05:43:00
A320fam	3,398	21,563	43.28%	56.62%	0.10%	104:05:59

Crew pairing results – Instances are solved to optimality

♣ Previously, the largest instances in the literature had $\simeq$ 750 flight legs (and were not necessarily solved to optimality).

♣ We are able to increase substantially the size of the instances solved to optimality.

Concluding remarks

Aircraft routing

- Aircraft routing: Determine the routes (sequence of flight legs) followed by the airplanes.

- At AIR FRANCE, the methodology presented in this talk is currently developed for the (more simple) aircraft routing problem.

Integrated problem

Traditional methodology:

- First SOLVE Aircraft routing.
- Then SOLVE Crew pairing.

(The routes impose constraints on $\mathcal{P}$: some connections are impossible if the crew has to change airplanes.)

♣ We are also able to address the Integrated problem, which aims at computing the routes and the pairings simultaneously.

♣ This leads to an additional reduction of costs.

Thank you