

Solving Large Crew Pairing Problems for Integrated and Robust Planning

Shervin Ahmadbeygi and Amy M. Cohn

INFORMS Annual Meeting

San Francisco Nov 2005



Motivation

- ❑ Integrated and robust planning require the simultaneous consideration of aircrafts and crews
 - Typically treated independently in current systems
- ❑ Results in a large-scale network of flights
- ❑ Existing methods mostly rely on enumeration with bounds to generate crew pairings
 - Loses tractability as the size of the problem grows
- ❑ Our goal: an alternative approach based on an Integer Programming (IP) formulation to generate crew pairings
- ❑ Can be used in conjunction with existing methods to solve significantly larger instances

Crew scheduling problem

- ❑ Crew cost is the second largest operating cost in the airline industry after fuel
- ❑ Assignment of **crew** to flights such that each flight is covered by a crew
- ❑ Elements of the crew scheduling problem
 - A **duty** is a sequence of flights to which a crew is assigned
 - A **pairing** is a sequence of duties and rest periods that starts and ends at the crewbase
 - A pairing must satisfy many rules and regulations (e.g. max flying time in duties)

Cost structure

□ Cost of a duty:

- $c_d = \max \{ \sum_{f \in d} \text{fly}_f, f_d \cdot e_d, \text{ming} \}$

□ Cost of a pairing:

- $c_p = \max \{ \sum_{d \in p} c_d, f_p \cdot \text{TAFB} \}$

□ Where:

- c_d = cost of a duty
- $\sum_{f \in d} \text{fly}_f$ = total flying time of the duty
- $f_d e_d$ = fraction of the elapsed time of the duty
- ming = a minimum guaranteed payment (contract)
- $f_p \cdot \text{TAFB}$ = fraction of pairing's elapsed time

Crew scheduling as a set partitioning problem

$$\begin{aligned} \text{Min} \quad & \sum_{p \in P} c_p x_p \\ \text{st.} \quad & \sum_{p \in P} \delta_{fp} x_p = 1 \quad \forall f \in F \\ & x_p \in \{0,1\} \quad \forall p \in P \end{aligned}$$

- P is the set of pairings, F is the set of flights
- c_p = the cost of pairing p
- $x_p = 1$ if pairing p is chosen and 0 otherwise
- $\delta_{fp} = 1$ if flight f is included in pairing p and 0 otherwise
(is given data)

Solving the SP problem

- ❑ Branch-and-price
 - Solve the SP problem as an integer programming problem using branch-and-bound
 - Solve each node of the branch-and-bound tree using the column generation technique

- ❑ Column generation
 - Start with a minimal subset of variables and construct a restricted master problem
(Gilmore and Gomory - 1961)
 - Construct an optimization problem (sub-problem or pricing problem) to find the most negative reduced cost variable

Sub-problem: generating pairings

- ❑ Complicating rules moved to the sub-problem
 - The problem is extremely challenging
- ❑ Common solution approaches
 - Enumeration (Klabjan and Schwan 2001)
 - Heuristics (Klabjan et al 2001)
 - Multi Label Shortest Path (MLSP)
 - ❑ The most commonly used approach to solve the sub-problem (Desrochers and Soumis 1988)
 - ❑ The algorithm becomes very inefficient as the number of flights or regulations increases

Our alternative approach

- Integer programming formulation
 - No IP-based formulation for the sub-problem in the literature to the best of our knowledge
 - Find a feasible pairing with the minimum reduced cost
 - Guaranteed optimality
 - The problem does not change in each iteration
 - Only the objective function is modified

Integer programming formulation

□ Binary decision variables to define a pairing:

- $xd_{(f_1, f_2), d}$: whether flight f_1 is followed by f_2 within the same duty d
- $xo_{(f_1, f_2), d}$: whether flight f_1 in duty d is followed by flight f_2 in the next duty
- w_f : whether flight f is the first flight of the pairing
- $wp_{f, d}$: whether flight f in duty d is the last flight of the pairing

Integer programming formulation

□ Binary decision (auxiliary) variables

- $z_{f,d}$: whether flight f is in duty d
- u_d : whether duty d is defined

□ Continuous decision variables: to linearize the cost function

- c : Cost of the pairing
- c_d : Cost of duty d
- ed_d : Elapsed time of duty d
- T : Elapsed time of the pairing (TAFB)

Integer programming formulation

□ Objective function:

$$\text{Min} \quad c - \sum_{f \in F} \sum_{d \in D} \pi_f z_{fd}$$

- Minimize the reduced cost of the pairing

Cost constraints

1. $c_d \geq \text{ming} \cdot u_d \quad \forall d \in \{1 \dots D\}$
2. $c_d \geq \sum_{f \in F} \text{fly}_f \cdot z_{f,d} \quad \forall d \in \{1 \dots D\}$
3. $c_d \geq \text{fd} \cdot ed_d \quad \forall d \in \{1 \dots D\}$
4. $c \geq \sum_{d \in \{1 \dots D\}} c_d$
5. $c \geq \text{fpa} \cdot T$
6. $T = \sum_{d \in \{1 \dots D\}} ed_d + \sum_{(f1,f2) \in \text{Opairs}} \sum_{d \in \{1 \dots D-1\}} \mathbf{O}_{(f1,f2)} \cdot \mathbf{xo}_{(f1,f2),d}$

Variables relationship constraints

1. $u_1 = 1$
2. $u_d = \sum_{(f1,f2) \in \text{Opairs}} \mathbf{xo}_{(f1,f2),d-1} \quad \forall d \in \{2 \dots D\}$
3. $z_{f,1} = w_f + \sum_{(f1,f) \in \text{Dpairs}} \mathbf{xd}_{(f1,f),1} \quad \forall f \in \text{FFlights}$
4. $z_{f,1} = \sum_{(f1,f) \in \text{Dpairs}} \mathbf{xd}_{(f1,f),1} \quad \forall f \notin \text{FFlights}$
5. $z_{f,D} = wp_{f,D} + \sum_{(f,f1) \in \text{Dpairs}} \mathbf{xd}_{(f,f1),D} \quad \forall f \in \text{EFlights}$
6. $z_{f,D} = \sum_{(f,f1) \in \text{Dpairs}} \mathbf{xd}_{(f,f1),D} \quad \forall f \notin \text{EFlights}$
7. $z_{f,d} = wp_{f,d} + \sum_{(f,f1) \in \text{Opairs}} \mathbf{xo}_{(f,f1),d} + \sum_{(f,f1) \in \text{Dpairs}} \mathbf{xd}_{(f,f1),d} \quad \forall f \in \text{EFlights}, \forall d = \{2 \dots D-1\}$
8. $z_{f,d} = \sum_{(f,f1) \in \text{Opairs}} \mathbf{xo}_{(f,f1),d} + \sum_{(f,f1) \in \text{Dpairs}} \mathbf{xd}_{(f,f1),d} \quad \forall f \notin \text{EFlights}, \forall d = \{2 \dots D-1\}$

Pairing and duty rules

1. $\sum_{f \in F} \mathbf{fly}_f \cdot \mathbf{z}_{f,d} \leq \mathbf{maxfd} \quad \forall d \in \{1 \dots D\}$
2. $\mathbf{ed}_d = \sum_{f \in F} \mathbf{fly}_f \cdot \mathbf{z}_{f,d} + \sum_{(f,f1) \in D_{\text{pairs}}} \mathbf{sit}_{(f,f1)} \cdot \mathbf{xd}_{(f,f1),d} \quad \forall d \in \{1 \dots D\}$
3. $\mathbf{ed}_d \leq \mathbf{maxed} \quad \forall d \in \{1 \dots D\}$
4. $\sum_{d \in \{1 \dots D\}} \mathbf{z}_{f,d} \leq 1 \quad \forall f \in F$

Balance constraints

1. $\sum_{f \in F} w_f = 1$
2. $w_f + \sum_{f1 \in F, (f1, f) \in D_{\text{pairs}}} xd_{(f1, f), 1} = \sum_{f1 \in F, (f, f1) \in D_{\text{pairs}}} xd_{(f, f1), 1} + \sum_{f1 \in F, (f, f1) \in O_{\text{pairs}}} xo_{(f1, f2), 1} \quad \forall f \in F_{\text{Flights}}$
3. $\sum_{f1 \in F, (f1, f) \in D_{\text{pairs}}} xd_{(f1, f), 1} = \sum_{f1 \in F, (f, f1) \in D_{\text{pairs}}} xd_{(f, f1), 1} + \sum_{f1 \in F, (f, f1) \in O_{\text{pairs}}} xo_{(f, f1), 1} + wp_{f, 1} \quad \forall f \in E_{\text{Flights}}$
4. $\sum_{f1 \in F, (f1, f) \in D_{\text{pairs}}} xd_{(f1, f), 1} = \sum_{f1 \in F, (f, f1) \in D_{\text{pairs}}} xd_{(f, f1), 1} + \sum_{f1 \in F, (f, f1) \in O_{\text{pairs}}} xo_{(f, f1), 1} \quad \forall f \notin F_{\text{Flights}} \cup E_{\text{Flights}}$

Balance constraints (cont'd)

5.
$$\sum_{f1 \in F, (f1,f) \in Dpairs} \mathbf{x}d_{(f1,f),D} + \sum_{f1 \in F, (f1,f) \in Opairs} \mathbf{x}o_{(f1,f),D-1} = \sum_{f1 \in F, (f,f1) \in Dpairs} \mathbf{x}d_{(f,f1),D} + \mathbf{w}p_{f,D}$$
$$\forall f \in EFlights$$
6.
$$\sum_{f1 \in F, (f1,f) \in Dpairs} \mathbf{x}d_{(f1,f),D} + \sum_{f \in F, (f1,f) \in Opairs} \mathbf{x}o_{(f1,f),D-1} = \sum_{f1 \in F, (f,f1) \in Dpairs} \mathbf{x}d_{(f,f1),D}$$
$$\forall f \in EFlights$$
7.
$$\sum_{f1 \in F, (f1,f) \in Dpairs} \mathbf{x}d_{(f1,f),d} + \sum_{f1 \in F, (f1,f) \in Opairs} \mathbf{x}o_{(f1,f),d-1} = \sum_{f1 \in F, (f,f1) \in Dpairs} \mathbf{x}d_{(f,f1),d} + \sum_{f1 \in F, (f,f1) \in Opairs} \mathbf{x}o_{(f,f1),d} + \mathbf{w}p_{f,d}$$
$$\forall f \in Eflights, \forall d \in \{2 \dots D-1\}$$
8.
$$\sum_{f1 \in F, (f1,f) \in Dpairs} \mathbf{x}d_{(f1,f),d} + \sum_{f1 \in F, (f1,f) \in Opairs} \mathbf{x}o_{(f1,f),d-1} = \sum_{f1 \in F, (f,f1) \in Dpairs} \mathbf{x}d_{(f,f1),d} + \sum_{f1 \in F, (f,f1) \in Opairs} \mathbf{x}o_{(f,f1),d}$$
$$\forall f \notin EFlights, \forall d \in \{2 \dots D-1\}$$

Discussion

☐ Reusability

- Do not need to set up a new problem in each iteration

☐ Scalability

- Order of $|F|^2D$ variables and $|F|D$ constraints

☐ Existence of efficient algorithms

- CPLEX solver

☐ Performance

- Unlike enumeration approaches, performance of the IP-based approach does not diminish as the master problem approaches optimality

Implementation

- ❑ Implemented the model using CPLEX8.1/C++
 - Sun Fire 280R
 - 1.2 GHz Superscalar SPARC V9 processor
 - Real memory: 8 GB
 - Swap memory: 1 GB

- ❑ Tested the model using real world data sets

Improvements

- Setting only z variables to integer is sufficient to get an integer solution
 - z variables: whether a duty covers a flight or not
- Observation: At the optimal iteration the IP can no longer find a negative reduced cost pairing. So the IP may take a long time to converge

$$\sum_{f \in F\text{Flights}} w_f = 1 \quad \Rightarrow \quad \sum_{f \in F\text{Flights}} w_f \leq 1$$

- However, this may loosen the LP relaxation

Improvements

□ Strengthening the LP relaxation

$$\sum_{f \in F\text{lights}} w_f \leq 1$$

$$\sum_{f \in F} \text{fly}_f z_{f,d} \leq \max f.d. \sum_{f \in F\text{lights}} w_f$$

- If a flight is partially used as the first flight, the corresponding z variables should also be adjusted

Computational results

<i>data</i>	<i>Dimension</i>	<i>1st integer solution</i>		<i>1st neg red. cost int. sol</i>		<i>% 15 optimality gap</i>		<i>optimality</i>	
		<i>time</i>	<i>node#</i>	<i>time</i>	<i>node#</i>	<i>time</i>	<i>node#</i>	<i>time</i>	<i>node#</i>
143	<i>R = 424 C = 1,406 NZ = 6,104</i>	<i>~ 0</i>	<i>0+</i>	<i>~ 0</i>	<i>0+</i>	<i>~ 0</i>	<i>0+</i>	<i>3 secs</i>	<i>20+</i>
509	<i>R = 2,259 C = 28,105 NZ = 122,113</i>	<i>~ 50 secs</i>	<i>0+</i>	<i>~ 50 secs</i>	<i>0+</i>	<i>~100 secs</i>	<i>0+</i>	<i>1202 secs</i>	<i>3300</i>
745	<i>R = 4,033 C = 84,482 NZ = 365,425</i>	<i>~ 100 secs</i>	<i>0+</i>	<i>~ 100 secs</i>	<i>0+</i>	<i>~ 150 secs</i>	<i>10+</i>	<i>3496 secs</i>	<i>2000</i>
1173	<i>R = 6,839 C = 173,852 NZ = 752,686</i>	<i>217 secs</i>	<i>10+</i>	<i>217 secs</i>	<i>10+</i>	<i>300 secs</i>	<i>20+</i>	<i>~5000 secs</i>	<i>3000 +</i>
2320	<i>R = 13,594 C = 665,332 NZ = 2,919,598</i>	<i>~ 3 hrs</i>	<i>0+</i>	<i>~ 4 hrs</i>	<i>0+</i>	<i>~14 hrs</i>	<i>300</i>	<i>NA</i>	<i>NA</i>

Introduction	Problem definition	Problem formulation	Results
--------------	--------------------	---------------------	---------

Acceleration techniques

□ Hot starts

- Developing a heuristic to generate initial feasible pairings to use in the master problem instead of the artificial variables

□ Generating multiple columns in each iteration

- Defining optimality gaps
- Output all the columns with reduced cost less than some gap

Acceleration techniques cont'd

□ Feasibility vs. optimality

- In the initial iterations we just look for a column with negative reduced cost (not necessarily the most neg)
- Add the constraint: $RC \leq -\varepsilon$
- Solve for a feasible solution
- Stop as soon as n feasible solutions are found

Acceleration techniques cont'd

□ Defining lower bounds on the flights potentials

- Based on an idea first introduced by (Cohn, Magazine and Polok - 2005)

- Define the potential for each flight as:

- $p_f = \mathbf{fly}_f - \pi_f$

- Disregard a flight if $p_f \geq P \sum_{f: p_f \geq P} \sum_{d=1}^D z_{fd} = 0$

Results after acceleration

IP-based formulation **Time to generate a feasible pairing with negative RC**

Data set	# flights	# rows*	# columns*	# nonzero elements*	CPU time (min) **
4	1,173	6,839	173,852	752,253	< 1 sec
5	2,320	13,596	665,333	2,921,912	(49 secs , 3.75 min)

* After reducing the problem

** Solved on the SOL server

Conclusion

- ❑ IP-based formulation can be used in order to generate good pairing in a reasonable time for large networks of flights
- ❑ It can be used in conjunction with the existing methods (future work)

Questions

