An Optimization Approach to Solving the Airline Crew Pairing Problem

Amy Cohn and Shervin AhmadBeygi

October, 2004

INFORMS Denver

MichiganEngineering
Industrial and Operations Engineering

Motivation

- Our original goal was to integrate fleet assignment (assigning aircraft types to flights) and crew scheduling (assigning pilots to flights)
- Just addressing the crew scheduling problem in the integrated model becomes much harder because it's *all* flights, not just one fleet type at a time
- Therefore we need to find a way to solve bigger instances of the crew scheduling problem, and to solve these instances more quickly so they can be imbedded in a broader framework

Crew Scheduling

- Crews must be assigned to flights such that each flight is covered
- Solution must satisfy many crew rules dictated by labor negotiations and government (e.g. FAA) regulations
  - Max flying time
  - Max time away from base
  - Min time between flights
  - Min rest overnight
  - …
- Cost of a crew's assignment is a non-linear function of flying time, time-away-from-base, idle time, etc.
- Many of the constraints are non-linear to capture as well

The Crew Pairing Problem

- Non-linearities are typically avoided by formulating a set partitioning (SP) problem based on selecting *pairings*
- A pairing is a sequence of flights that can be flown by a single crew over a series of days, starting and ending at their home base
- We need to select a set of feasible pairings such that every flight is included in exactly one pairing

The Crew Pairing Formulation

$$\text{Min} \quad \Sigma_p \, c_p \, x_p$$

$$\text{St} \quad \Sigma_p \, \delta_{fp} \, x_p = 1 \qquad \forall \, f$$

$$x_p \in \{0, 1\} \qquad \forall \, p$$

- x_p is a binary variable that takes value 1 if pairing p is included in the solution, else 0
- δ_{fp} is a constant with value 1 if flight f is included in pairing p, else 0
- c_p is the cost of pairing p

Solving the Crew Pairing Problem

- Can't write out the SP model explicitly – hundreds of billions of pairings (or more!)
- Solve using *branch-and-price*
  - Solve the IP using branch-and-bound
  - Solve each node of the branch-and-bound tree using *column generation*
    - Instead of explicitly computing the reduced cost of each non-basic variable, solve an optimization problem – *pricing problem* – to identify the most negative reduced cost column

Generating Pairings

- One of the key challenges is how to find negative reduced cost pairings
  - All of the non-linearities that we eliminated from the master problem are now in the pricing problem!
- There are many interesting approaches for trying to find good pairings (e.g. Klabjan, Johnson, and Nemhauser 2001)
  - Randomized approaches
  - Parallel algorithms
  - Approximations

Generating Pairings, cont.

- Predominant approach is *multi-label shortest path* (e.g. Desrochers and Soumis 1988; Lavoie, Minoux, and Odier 1988)
  - A pairing corresponds to a path in a flight network
  - The cost of a path is the cost of the corresponding pairing minus the sum of the duals associated with the flights in that path
  - Labels are used to capture feasibility criteria

Challenges of MLSP

- As the number of flights grows, performance deteriorates quickly
- Furthermore, **we** claim that the use of dominance in MLSP can in fact lead to sub-optimality
  - Because flights cannot be repeated in a pairing, no fragment can ever truly dominate another – will always be lacking opportunities based on flights already covered

Alternative Approach

- Goal:
  - Solve larger problem instances (i.e. more flights)
  - Guarantee an optimal solution if desired
- Approach: IP-based formulation
  - Need to capture duty rules, pairing rules, and non-linear objective function

Model – Binary Variables

- $xd_{(f1,f2),\,d}$: whether flight f1 is followed by f2 within duty d
- $xo_{(f1,f2),\,d}$: whether flight f1 in duty d is followed by flight f2 in duty d+1
- w_f : whether flight f is the first flight
- $wp_{f,d}$: whether flight f is the last flight and occurs in duty d
- $z_{f,d}$: whether flight f appears in duty d
- u_d : whether there is a dth duty

Model – Continuous Variables

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

Model – Objective Function

- Minimize reduced cost of the chosen pairing:

$$\text{Min} \qquad c - [\, \Sigma_{f \in F} \Sigma_{d \in \{1 \ldots D\}} (pi_f * z_{f,d}) \,]$$

Model – Balance Constraints

- $\sum_{f \in F} w_f = 1$
- $w_f + \sum_{f \in F,(f1,f) \in Dpairs} xd_{(f1,f),1} = \sum_{f1 \in F,(f,f1) \in Dpairs} xd_{(f,f1),1} + \sum_{f1 \in F,(f,f1) \in Opairs} xo_{(f1,f2),1}$
- $\sum_{f \in F,(f1,f) \in D$ [illegible] $xd_{(f1,f),1} \sum_{f1 \in F,(f,f1)}$ [illegible] $xd_{(f,f1),1} +$ $\sum_{f1 \in F,(f,f1) \in}$ [illegible]
- $\sum_{f \in F,(f1,f) \in D}$ [illegible] $\sum_{f1 \in F,(f,f1) \in}$ [illegible]
- $\sum_{f \in F,(f1,f) \in D}$ [illegible] $_1 = \sum_{f1 \in F,(f,f}$ [illegible]
- $\sum_{f \in F,(f1,f) \in Dpairs}$ [illegible] $=$ $\sum_{f1 \in F,(f,f1) \in Dpairs} xd_{(f,f1),D}$
- $\sum_{f \in F,(f1,f) \in Dpairs} xd_{(f1,f),d} + \sum_{f1 \in F,(f1,f) \in Opairs} xo_{(f1,f),d-1} = \sum_{f1 \in F,(f,f1) \in Dpairs} xd_{(f,f1),d} + \sum_{f1 \in F,\ (f,f1) \in Opairs} xo_{(f,f1),d} + wp_{f,d}$
- $\sum_{f \in F,(f1,f) \in Dpairs} xd_{(f1f),d} + \sum_{f \in F,(f1,f) \in Opairs} xo_{(f1,f),d-1} = \sum_{f1 \in F,(f,f1) \in Dpairs} xd_{(f,f1),d} + \sum_{f1 \in F,(f,f1) \in Opairs} xo_{(f,f1),d}$

If flight f appears in the pairing, then it must be preceded by exactly one flight (or start the pairing) and followed by exactly one flight (or end the pairing)

Model – Cost Constraints

- $c_d \geq ming * u_d$
- $c_d \geq \sum_{f \in F} fly_f * z_{f,d}$
- $c_d \geq fd * ed_d$
- $c \geq \sum_{d \in \{1 \ldots D\}} c_d$
- $c \geq fpa*T$
- $T = \sum_{d \in \{1 \ldots D\}} ed_d + \sum_{(f1,f2) \in Opairs} \sum_{d \in \{1 \ldots D-1\}} O_{(f1,f2)} * xo_{(f1,f2),d}$

Model – Duty and Pairing Constraints

- $\sum_{f \in F} fly_f * z_{f,d} \leq maxfd$
- $ed_d = \sum_{f \in F} fly_f * z_{f,d}$
  $+ \sum_{(f,f1) \in Dpairs} sit_{(f,f1)} * xd_{(f,f1),d}$
- $ed_d \leq maxed$
- $\sum_{d \in \{1 \ldots D\}} z_{f',d} \leq 1$
- …

Model – Auxiliary Constraints

- $u_1 = 1$
- $u_d = \sum_{(f1,f2)\in Opairs} xo_{(f1,f2),d-1}$
- $z_{f,1} = w_f + \sum_{(f1,f)\in Dpairs} xd_{(f1,f),1}$
- $z_{f,1} =$
- $z_{f,D} =$
- $z_{f,D} = \sum_{(f,f1)\in Dpairs} xd_{(f,f1),D}$
- $z_{f,d} = wp_{f,d} + \sum_{(f,f1)\in Opairs} xo_{(f,f1),d} + \sum_{(f,f1)\in Dpairs} xd_{(f,f1),d}$
- $z_{f,d} = \sum_{(f,f1)\in Opairs} xo_{(f,f1),d} + \sum_{(f,f1)\in Dpairs} xd_{(f,f1),d}$

-- Determine the number of duties
-- Determine which flights are covered

Benefits of IP Approach

- IP approach has a number of benefits…
  - More tractable for large instances
  - Provable optimality
  - Model does not change with duals, except in the objective function
    - Use of "hot starts"
  - Better when duals are close to optimal?

Addressing Drawbacks

- …and a number of drawbacks…
  - One column at a time
  - Worse when duals are far from optimal?
  - May take a long time to find the “best” column when any negative column will do
- …so let’s use both!
  - MLSP early on to find lots of columns
  - Switch to IP when MLSP tree grows too large or takes to long to converge

Current Status

- Prototype of pricing problem is in place but not fast enough
- Implementing directly with C++, CPLEX to speed up pre-processing and setting up of model, enable use of “hot starts”
- Next step is to integrate with master problem

Future Research

- Modeling improvements to strengthen the LP relaxation
- Manipulation of CPLEX parameters for finding negative columns quickly
- Hybrid approach using IP approach and MLSP
- Integration of crew scheduling and fleet assignment