DM204 – Spring 2011

Scheduling, Timetabling and Routing

Lecture 9
Workforce Scheduling

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Workforce Scheduling
2. Employee Timetabling
   - Shift Scheduling
   - Nurse Scheduling
3. Crew Scheduling
4. Transportations

Course Overview

- ✔ Scheduling
  - ✔ Classification
  - ✔ Complexity issues
  - ✔ Single Machine
  - ✔ Parallel Machine
  - ✔ Flow Shop and Job Shop
  - ✔ Resource Constrained Project Scheduling Model

- Timetabling
  - ✔ Sport Timetabling
  - ✔ Reservations and Education
  - ✔ University Timetabling
  - Crew Scheduling
  - Public Transports

- Vechicle Routing
  - Capacited Models
  - Time Windows models
  - Rich Models

Outline

1. Workforce Scheduling

2. Employee Timetabling
   - Shift Scheduling
   - Nurse Scheduling

3. Crew Scheduling

4. Transportations

Workforce Scheduling
Overview

Shift: consecutive working hours

Roster: shift and rest day patterns over a fixed period of time (a week or a month)

Two main approaches:

- coordinate the design of the rosters and the assignment of the shifts to the employees, and solve it as a single problem.
- consider the scheduling of the actual employees only after the rosters are designed, solve two problems in series.

Features to consider: rest periods, days off, preferences, availabilities, skills.

Workforce Scheduling
Overview

Workforce Scheduling:

1. Crew Scheduling and Rostering
2. Employee Timetabling

1. Crew Scheduling and Rostering is workforce scheduling applied in the transportation and logistics sector for enterprises such as airlines, railways, mass transit companies and bus companies (pilots, attendants, ground staff, guards, drivers, etc.)

   The peculiarity is finding logistically feasible assignments.

Workforce Scheduling

Overview

2. Employee timetabling (aka labor scheduling) is the operation of assigning employees to tasks in a set of shifts during a fixed period of time, typically a week.

Examples of employee timetabling problems include:

- assignment of nurses to shifts in hospitals
- assignment of workers to cash registers in a large store
- assignment of phone operators to shifts and stations in a service-oriented call-center

Differences with Crew scheduling:

- no need to travel to perform tasks in locations
- start and finish time not predetermined

Outline

1. Workforce Scheduling

2. Employee Timetabling
   - Shift Scheduling
   - Nurse Scheduling

3. Crew Scheduling

4. Transportations

Shift Scheduling

Creating daily shifts:

- during each period, b_i persons required
- decide working rosters made of m time intervals not necessarily identical
- n different shift patterns (columns of matrix A) each with a cost c

$$\begin{array}{ll} \min & c^T x \\ st & Ax \geq b \\ & x \geq 0 \text{ and integer} \end{array}$$

$$\min c^T x$$

$$\begin{array}{r} 10am - 11pm \\ 11am - 12am \\ 12am - 1pm \\ 1pm - 2pm \\ 2pm - 3pm \\ 3pm - 4pm \\ 4pm - 5pm \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} x \geq \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 4 \\ 2 \\ 2 \end{bmatrix}$$

$$x \geq 0 \text{ and integer}$$

(k, m)-cyclic Staffing Problem

Assign persons to an m-period cyclic schedule so that:

- requirements b_i are met
- each person works a shift of k consecutive periods and is free for the other $m - k$ periods. (periods 1 and m are consecutive)

and the cost of the assignment is minimized.

$$
\begin{array}{lll}
\min & c^T x & \\
st & \begin{array}{r} \textit{Monday} \\ \textit{Tuesday} \\ \textit{Wednesday} \\ \textit{Thursday} \\ \textit{Friday} \\ \textit{Saturday} \\ \textit{Sunday} \end{array}
\begin{bmatrix}
1 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix} x \geq \begin{bmatrix} 3 \\ 4 \\ 6 \\ 4 \\ 7 \\ 8 \\ 7 \end{bmatrix} & \text{(IP)} \\
& x \geq 0 \text{ and integer} &
\end{array}
$$

Total Unimodular Matrices

Resume'

Recall: Totally Unimodular Matrices

Definition: A matrix A is totally unimodular (TU) if every square submatrix of A has determinant +1, -1 or 0.

Proposition 1: The linear program $\max\{cx : Ax \leq b, x \in \mathbf{R}_+^m\}$ has an integral optimal solution for all integer vectors b for which it has a finite optimal value if A is totally unimodular

Recognizing total unimodularity can be done in polynomial time (see [Schrijver, 1986])

Total Unimodular Matrices
Resume'

Definition

A $(0,1)$–matrix B has the consecutive 1's property if for any column j, $b_{ij} = b_{i'j} = 1$ with $i < i'$ implies $b_{lj} = 1$ for $i < l < i'$.
That is, if there is a permutation of the rows such that the 1's in each column appear consecutively.

Whether a matrix has the consecutive 1's property can be determined in polynomial time [D. R. Fulkerson and O. A. Gross; Incidence matrices and interval graphs. 1965 Pacific J. Math. 15(3) 835-855.]

A matrix with consecutive 1's property is called an interval matrix

Proposition: Consecutive 1's matrices are TUM.

What about this matrix?

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Definition A $(0,1)$-matrix B has the circular 1's property for rows (resp. for columns) if the columns of B can be permuted so that the 1's in each row are circular, that is, appear in a circularly consecutive fashion

The circular 1's property for columns does not imply circular 1's property for rows.

Whether a matrix has the circular 1's property for rows (resp. columns) can be determined in $O(m^2n)$ time [A. Tucker, Matrix characterizations of circular-arc graphs. (1971) Pacific J. Math. 39(2) 535-545]

Integer programs where the constraint matrix A have the circular 1's property for **rows** can be solved efficiently as follows:

Step 1 Solve the linear relaxation of (IP) to obtain $x_1', \ldots, x_n'$. If $x_1', \ldots, x_n'$ are integer, then it is optimal for (IP) and STOP. Otherwise go to Step 2.

Step 2 Form two linear programs LP1 and LP2 from the relaxation of the original problem by adding respectively the constraints

$$x_1 + \ldots + x_n = \lfloor x_1' + \ldots + x_n' \rfloor \tag{LP1}$$

and

$$x_1 + \ldots + x_n = \lceil x_1' + \ldots + x_n' \rceil \tag{LP2}$$

From LP1 and LP2 an integral solution certainly arises (P)

Cyclic Staffing with Overtime

- Hourly requirements b_i
- Basic work shift 8 hours
- Overtime of up to additional 8 hours possible

minimize $\quad cx$

subject to

$$
\begin{array}{c}
07\\08\\09\\10\\11\\12\\13\\14\\15\\16\\17\\18\\19\\20\\21\\22\\23\\24\\01\\02\\03\\04\\05\\06
\end{array}
\left[\begin{array}{ccccccccc@{\quad}ccccccccc@{\quad}ccccccccc}
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1\\
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1
\end{array}\right]
x \geq b
$$

$x \geq 0$ and integer.

Days-Off Scheduling

- Guarantee two days-off each week, including every other weekend.

IP with matrix A:

$$
\begin{array}{l}
\\ \\ \\ \texttt{first week} \\ \\ \\ \\ \\ \\ \\ \texttt{second week} \\ \\ \\ \\
\end{array}
\left[\begin{array}{cccccc|cccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
\hline
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
$$

Cyclic Staffing with Part-Time Workers

- Columns of A describe the work-shifts
- Part-time employees can be hired for each time period i at cost c'_i per worker

$$
\begin{aligned}
\min \quad & cx + c'x' \\
st \quad & Ax + Ix' \geq b \\
& x, x' \geq 0 \text{ and integer}
\end{aligned}
$$

Cyclic Staffing with Linear Penalties for Understaffing and Overstaffing

- demands are not rigid
- a cost c'_i for understaffing and a cost c''_i for overstaffing
- x' level of understaffing

$$
\begin{aligned}
\min \quad & cx + c'x' + c''(b - Ax - x') \\
st \quad & Ax + Ix' \geq b \\
& x, x' \geq 0 \text{ and integer}
\end{aligned}
$$

Nurse Scheduling
A CP approach

- Hospital: head nurses on duty seven days a week 24 hours a day
- Three 8 hours shifts per day (1: daytime, 2: evening, 3: night)
- In a day each shift must be staffed by a different nurse
- The schedule must be the same every week
- Four nurses are available (A,B,C,D) and must work at least 5 days a week.
- No shift should be staffed by more than two different nurses during the week
- No employee is asked to work different shifts on two consecutive days
- An employee that works shifts 2 and 3 must do so at least two days in a row.

Mainly a feasibility problem

A CP approach

Two solution representations

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Shift 1	A	B	A	A	A	A	A
Shift 2	C	C	C	B	B	B	B
Shift 3	D	D	D	D	C	C	D

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Worker A	1	0	1	1	1	1	1
Worker B	0	1	0	2	2	2	2
Worker C	2	2	2	0	3	3	0
Worker D	3	3	3	3	0	0	3

Variables: w_{sd} nurse assigned to shift s on day d
and y_{id} the shift assigned to i on day d

$$w_{sd} \in \{A, B, C, D\} \qquad y_{id} \in \{0, 1, 2, 3\}$$

Three different nurses are scheduled each day

$$\texttt{alldiff}(w_{\cdot d}) \qquad \forall d$$

Every nurse is assigned to at least 5 days of work

$$\texttt{cardinality}(w_{\cdot\cdot} \mid (A, B, C, D), (5, 5, 5, 5), (6, 6, 6, 6))$$

At most two nurses work any given shift

$$\texttt{nvalues}(w_{s\cdot} \mid 1, 2) \qquad \forall s$$

All shifts assigned for each day

$$\texttt{alldiff}(y_{\cdot d}) \qquad \forall d$$

Maximal sequence of consecutive variables that take the same values

$$\texttt{stretch-cycle}(y_{i\cdot} \mid (2,3),(2,2),(6,6),P)$$
$$\forall i,\ P = \{(s,0),(0,s) \mid s = 1,2,3\}$$

Channeling constraints between the two representations:
on any day, the nurse assigned to the shift to which nurse i is assigned must be nurse i (element constraint)

$$w_{y_{id},d} = i \qquad \forall i, d$$
$$y_{w_{sd},d} = s \qquad \forall s, d$$

The complete CP model

$$\text{Alldiff:} \left\{ \begin{array}{c} (w_{\cdot d}) \\ (y_{\cdot d}) \end{array} \right\}, \text{ all } d$$

$$\text{Cardinality: } (w_{\cdot\cdot} \,|\, (A, B, C, D), (5, 5, 5, 5), (6, 6, 6, 6))$$

$$\text{Nvalues: } (w_{s\cdot} \,|\, 1, 2), \text{ all } s$$

$$\text{Stretch-cycle: } (y_{i\cdot} \,|\, (2, 3), (2, 2), (6, 6), P), \text{ all } i$$

$$\text{Linear:} \left\{ \begin{array}{l} w_{y_{id}d} = i, \text{ all } i \\ y_{w_{sd}d} = s, \text{ all } s \end{array} \right\}, \text{ all } d$$

$$\text{Domains:} \left\{ \begin{array}{l} w_{sd} \in \{A, B, C, D\}, \ s = 1, 2, 3 \\ y_{id} \in \{0, 1, 2, 3\}, \ i = A, B, C, D \end{array} \right\}, \text{ all } d$$

Constraint Propagation:

- alldiff: matching
- nvalues: max flow
- stretch: poly-time dynamic programming
- index expressions $w_{y_{id}d}$ replaced by z and constraint: $\texttt{element}(y, x, z)$: z be equal to y-th variable in list $x_1, \ldots, x_m$

Search:

- branching by splitting domanins with more than one element
- first fail branching
- symmetry breaking:
  - employees are indistinguishable
  - shifts 2 and 3 are indistinguishable
  - days can be rotated

  Eg: fix A, B, C to work $1, 2, 3$ resp. on sunday

Heuristic Methods

- Local search and metaheuristic methods are used if the problem has large scale.
- Procedures are very similar to what we saw for course timetabling.

Outline

1. Workforce Scheduling

2. Employee Timetabling
   - Shift Scheduling
   - Nurse Scheduling

3. Crew Scheduling

4. Transportations

Crew Scheduling

Usually divided into two distinct subproblems:

- Pairings problem
  construction of sequences of flights, ie, trips, duties
  sequence of flight legs that originate and terminates at crew's home
  $\rightsquigarrow$ set partitioning approach

- Rostering
  assignment of the trips or duties to individual members of the crew over the rosting period (ie, 4 weeks)
  trips range in length from one day to 15 days including rest periods
  $\rightsquigarrow$ Bidline Systems or heuristics

The two problems can however be solved together via a generalization of set partitioning.

Crew Scheduling

Pairings problem

Input:

- A set of m flight legs (departure, arrival, duration)
- A set of crews
- A set of n (very large) feasible and permissible combinations of flights legs that a crew can handle (eg, round trips)
- A flight leg i can be part of more than one round trip
- Each round trip j has a cost c_j

Output: A set of round trips of mimimun total cost

Set partitioning problem:

$$
\begin{aligned}
\min \quad & c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
& a_{11} x_1 + a_{12} x_2 + \ldots + \ldots a_{1n} x_n = 1 \\
& a_{21} x_1 + a_{22} x_2 + \ldots + \ldots a_{2n} x_n = 1 \\
& \vdots \\
& a_{m1} x_1 + a_{m2} x_2 + \ldots + \ldots a_{mn} x_n = 1 \\
& x_j \in \{0, 1\}, \qquad \forall j = 1, \ldots, n
\end{aligned}
$$

Route	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
c_j	8	10	4	4	2	14	10	8	8	10	11	12	6	6	5
	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0
	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0
	0	0	0	1	0	0	0	1	0	0	1	0	1	0	1
	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1

Set partitioning or set covering??

Often treated as set covering because:

- its linear programming relaxation is numerically more stable and thus easier to solve
- it is trivial to construct a feasible integer solution from a solution to the linear programming relaxation
- it makes it possible to restrict to only rosters of maximal length

Crew Scheduling

Generalization of set partitioning

With a set of p crew members
Generalized set partitioning problem:

$$
\begin{array}{llllll}
\min & c_1x_1 + c_2x_2 + \ldots + c_nx_n & & & & \\
 & a_{11}x_1 + a_{12}x_2 + & \ldots & & + \ldots a_{1n}x_n & = b_1 \\
 & a_{21}x_1 + a_{22}x_2 + & \ldots & & + \ldots a_{2n}x_n & = b_2 \\
 & \vdots & & & & \\
 & a_{m1}x_1 + a_{m2}x_2 + & \ldots & & + \ldots a_{mn}x_n & = b_3 \\
 & x_1 + x_2 + \ldots x_l & & & & = 1 \\
 & & x_i + x_{i+1} + \ldots x_s & & & = 1 \\
 & \vdots & & & & \\
 & x_j \in \{0, 1\}, \qquad \forall j = 1, \ldots, n & & & &
\end{array}
$$

Ryan & Foster branching rule

Solving the SPP and SCP integer program

- trivial 1–0 branching leads to a very unbalanced tree in which the 0-branch has little effect

- constraint branching [Ryan, Foster, 1981]
  Identify constraints r_1, r_2 with

$$0 < \sum_{j \in J(s, r_2)} x_j < 1$$

  $J(r_1, r_2)$ all columns covering r_1, r_2 simultaneously.
  (there certainly exists one such pair of constraints)
  Branch on:

$$\sum_{j \in J(r_1, r_2)} x_j \leq 0 \qquad\qquad \sum_{j \in J(r_1, r_2)} x_j \geq 1$$

Motivation:

A balanced matrix B is an integer matrix that does not contain any submatrix of odd order having row and column sums equal to two (ie, a cycle without chords in the corresponding graph).

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

NO

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

OK

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

OK

Other insights:

- constraint ordering (petal structure)
- unique subsequence

The remaining fraction must be given by variables that do not cover r_1 and r_2 simultaneously

$$\sum_{j \in J(r_1, r_2)} x_j \leq 0 \quad \text{0-branch}$$
$$\sum_{j \in J(r_1, r_2)} x_j \geq 1 \quad \text{1-branch}$$

- 0-branch: constraints r_1 and r_2 must not be covered together
- 1-branch: constraints r_1 and r_2 must be covered together
  In SPP can be imposed by forcing to zero all variables/duties in complementary sets $J(\bar{r}_1, r_2)$, $J(r_1, \bar{r}_2)$ ($\bar{r} \equiv$ constraint not covered)

In practice, select r_1 and r_2 such that $\sum_{j \in J(r_1, r_2)} x_j$ is maximized and descend first in the 1-branches

Further Readings

- D.M. Ryan. The Solution of Massive Generalized Set Partitioning Problems in Aircrew Rostering. The Journal of the Operational Research Society, Palgrave Macmillan Journals on behalf of the Operational Research Society, 1992, 43(5), 459-467
- D.M. Ryan and B.A. Foster. An integer programming approach to scheduling. A. Wren (ed.). Computer Scheduling of Public Transport, North-Holland, Amsterdam, 1981, 269-280
- M. Pinedo, Planning and Scheduling in Manufacturing and Services. Springer Verlag, 2005 (Sec. 12.6)

Outline

1. Workforce Scheduling

2. Employee Timetabling
   - Shift Scheduling
   - Nurse Scheduling

3. Crew Scheduling

4. Transportations

OR in Transports

We consider here:

- Tanker Scheduling
- Daily Aricraft Scheduling
- Train Timetabling
- Vehicle (Truck) Routing

Tanker Scheduling
(sec. 11.2)

Input:

- p ports

  limits on the physical characteristics of the ships

- n cargoes:

  type, quantity, load port, delivery port, time window constraints on the load and delivery times

- ships (tanker): s company-owned plus others chartered
  Each ship has a capacity, draught, speed, fuel consumption, starting location and times

  These determine the costs of a shipment: c_i^l (company-owned) c_j^* (chartered)

Output: A schedule for each ship, that is, an itinerary listing the ports visited and the time of entry in each port within the rolling horizon such that the total cost of transportation is minimized

Two phase approach:

determine for each ship i the set S_i of all possible itineraries

select the itineraries for the ships by solving an IP problem

Phase 1 can be solved by some ad-hoc enumeration or heuristic algorithm that checks the feasibility of the itinerary and its cost.
Phase 2 Set packing problem with additional constraints (next slide)

For each itinerary l of ship i compute the profit with respect to charter:

$$\pi_i^l = \sum_{j=1}^{n} a_{ij}^l c_j^* - c_i^l$$

where $a_{ij}^l = 1$ if cargo j is shipped by ship i in itinerary l and 0 otherwise.

A set packing model with additional constraints

Variables

$$x_i^l \in \{0,1\} \qquad \forall i = 1,\dots,s;\ l \in S_i$$

Each cargo is assigned to at most one ship:

$$\sum_{i=1}^{s} \sum_{l \in S_i} a_{ij}^l x_i^l \leq 1 \qquad \forall j = 1,\dots,n$$

Each tanker can be assigned at most one itinerary

$$\sum_{l \in S_i} x_i^l \leq 1 \qquad \forall i = 1,\dots,s$$

Objective: maximize profit

$$\max \sum_{i=1}^{s} \sum_{l \in S_i} \pi_i^l x_i^l$$

Customized branching mechanisms

- select variable x_i^l and branch with $x_i^l = 0$ and $x_i^l = 1$
  select variable with value closest to 0.5
  for $x_i^l = 1$ remove schedules for other ships that have cargo in common
- select a ship i and generate for each schedule l in S_i a branch with $x_i^l = 1$
  select the ship for example on the basis of its importance, or select the one with most fractional number

OR in Air Transport Industry

- Aircraft and Crew Schedule Planning
  - Schedule Design (specifies legs and times)
  - Fleet Assignment
  - Aircraft Maintenance Routing
  - Crew Scheduling
    - crew pairing problem
    - crew assignment problem (bidlines)
- Airline Revenue Management
  - number of seats available at fare level
  - overbooking
  - fare class mix (nested booking limits)
- Aviation Infrastructure
  - airports
    - runaways scheduling (queue models, simulation; dispatching, optimization)
    - gate assignments
  - air traffic management

Daily Aircraft Routing and Scheduling
(Sec. 11.3)

[Desaulniers, Desrosiers, Dumas, Solomon and Soumis, 1997]

Input:

- L set of flight legs with airport of origin and arrival, departure time windows $[e_i, l_i]$, $i \in L$, duration, cost/revenue
- Heterogeneous aircraft fleet T, with m_t aircrafts of type $t \in T$

Output: For each aircraft, a sequence of operational flight legs and departure times such that operational constraints are satisfied:

- number of planes for each type
- restrictions on certain aircraft types at certain times and certain airports
- required connections between flight legs (thrus)
- limits on daily traffic at certain airports
- balance of airplane types at each airport

and the total profits are maximized.

- L_t denotes the set of flights that can be flown by aircraft of type t
- S_t the set of feasible schedules for an aircraft of type t (inclusive of the empty set)
- $a^l_{ti} = \{0, 1\}$ indicates if leg i is covered by $l \in S_t$
- π_{ti} profit of covering leg i with aircraft of type i

$$\pi^l_t = \sum_{i \in L_t} \pi_{ti} a^l_{ti} \qquad \text{for } l \in S_t$$

- P set of airports, P_t set of airports that can accommodate type t
- o^l_{tp} and d^l_{tp} equal to 1 if schedule l, $l \in S_t$ starts and ends, resp., at airport p

A set partitioning model with additional constraints

Variables

$$x_t^l \in \{0,1\} \qquad \forall t \in T;\ l \in S_t \qquad \text{and} \qquad x_t^0 \in \mathbf{N} \qquad \forall t \in T$$

Maximum number of aircraft of each type:

$$\sum_{l \in S_t} x_t^l = m_t \qquad \forall t \in T$$

Each flight leg is covered exactly once:

$$\sum_{t \in T} \sum_{l \in S_t} a_{ti}^l x_t^l = 1 \qquad \forall i \in L$$

Flow conservation at the beginning and end of day for each aircraft type

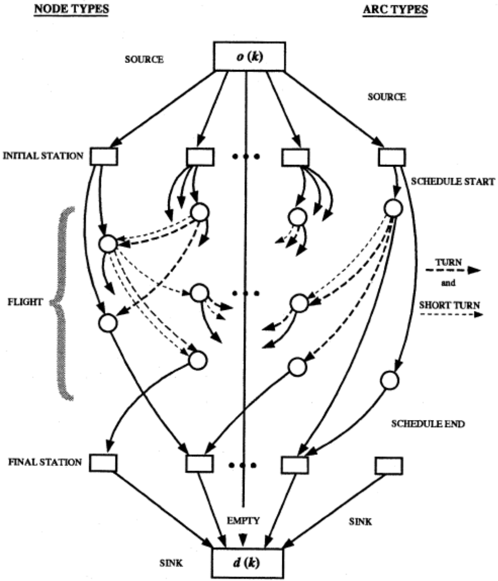
$$\sum_{l \in S_t} (o_{tp}^l - d_{tp}^l) x_t^l = 0 \qquad \forall t \in T;\ p \in P$$

Maximize total anticipate profit

$$\max \sum_{t \in T} \sum_{l \in S_t} \pi_t^l x_t^l$$

Solution Strategy: branch-and-price

- At the high level branch-and-bound similar to the Tanker Scheduling case
- Upper bounds obtained solving linear relaxations by column generation.
  - Decomposition into
    - Restricted Master problem, defined over a restricted number of schedules
    - Subproblem, used to test the optimality or to find a new feasible schedule to add to the master problem (column generation)
  - Each restricted master problem solved by LP.
    It finds current optimal solution and dual variables
  - Subproblem (or pricing problem) corresponds to finding longest path with time windows in a network defined by using dual variables of the current optimal solution of the master problem. Solve by dynamic programming.


NODE TYPES
ARC TYPES
SOURCE
o (k)
SOURCE
INITIAL STATION
SCHEDULE START
FLIGHT
TURN
and
SHORT TURN
SCHEDULE END
FINAL STATION
EMPTY
SINK
SINK
d (k)

$$\text{Maximize} \sum_{k\in K} \sum_{(i,j)\in A^k} c_{ij}^k X_{ij}^k \tag{8}$$

subject to:

$$\sum_{k\in K} \sum_{j:(i,j)\in A^k} X_{ij}^k = 1 \quad \forall i \in N, \tag{9}$$

$$\sum_{i:(i,s)\in NS_2^k} X_{is}^k - \sum_{j:(s,j)\in S_1 N^k} X_{sj}^k = 0 \quad \forall k \in K, \forall s \in S^k, \tag{10}$$

$$\sum_{s\in S_1^k} X_{o(k),s}^k + X_{o(k),d(k)}^k = n^k \quad \forall k \in K, \tag{11}$$

$$\sum_{i:(i,j)\in A^k} X_{ij}^k - \sum_{i:(j,i)\in A^k} X_{ji}^k = 0$$

$$\forall k \in K, \forall j \in V^k \setminus \{o(k), d(k)\}, \tag{12}$$

$$\sum_{s\in S_2^k} X_{s,d(k)}^k + X_{o(k),d(k)}^k = n^k \quad \forall k \in K, \tag{13}$$

$$X_{ij}^k \geq 0 \quad \forall k \in K, \forall (i,j) \in A^k, \tag{14}$$

$$a_i^k \leq T_i^k \leq b_i^k \quad \forall k \in K, \forall i \in V^k, \tag{15}$$

$$X_{ij}^k (T_i^k + d_{ij}^k - T_j^k) \leq 0 \quad \forall k \in K, \forall (i,j) \in A^k, \tag{16}$$

$$X_{ij}^k \text{ integer} \quad \forall k \in K, \forall (i,j) \in A^k. \tag{17}$$

B&B strategies

- 0–1 branching on set partitioning formulation:
  1 leads to removal of flights covered from the network
  but 0 is more complicated to handle

- multicommodity formulation
  can decompose from node-path to node-arc branching forced on the flow variables x_{ij}^k variables

- binary decision on linear combination of flow variables such as:

$$X_{ij} = \sum_{k \in K} X_{ij}^k \quad \forall i, j \in N$$

  (connection ij whatever is the aircraft) and

$$X_i^k = \sum_{ij \in A^k} X_{ij}^k \quad \forall k \in K, i \in N^k$$

  (assignment of aircraft of type k to flight leg i)

$$X_{\sigma(k)}^k = \sum_{s \in S_i^k} X_{\sigma(k),s}^k \quad \forall k \in K$$