DMP204
SCHEDULING,
TIMETABLING AND ROUTING

Lecture 20
Timetabling in Transportation

Marco Chiarandini

Outline

Outline

1. Transportation Timetabling
   Tanker Scheduling
   Coping with hard IPs
   Air Transport

Outline

Problems

- Tanker Scheduling
- Aircraft Routing and Scheduling
- Public Transports

MIP Models using complicated variables: Let a variable represent a road trip, a schedule section, or a whole schedule for a crew.

- Set packing
- Set partitioning

Solution techniques

- Branch and bound
- Lagrangian relaxation (solution without Simplex)
- Branch and price (column generation)

Tanker Scheduling

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

Network Flow

Network representation of the tanker scheduling problem:

- a node for each shipment
- an arc from i to j if possible to accomplish j after completing i
- a directed path corresponds to a feasible schedule for the tank

Model as minimum value problem solvable by maximum flow algorithm in the following network:

- split each node i into i' and i''
- introduce shipment arcs (i', i'') of flow lower bound 1
- introduce source and sink
- set all flow upper bounds to 1

Finds minimum number of ships required to cover the cargos. Does not include costs.

IP model

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.

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.

Phase 2:

A set packing model with additional constraints

Variables

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

Each cargo is assigned to at most one ship:

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

Each tanker can be assigned at most one itinerary

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

Objective: maximize profit

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

Branch and bound (Variable fixing)

Solve LP relaxation (this provides an upper bound) and branch by:

- selecting a fractional variable with value closest to 0.5 (keep tree balanced)
  set a branch $x_i^l = 0$ and
  the other $x_i^l = 1$ (this rules out the other itineraries of ship i and of other ships covering the same cargo)
- selecting one ship and branching on its itineraries
  select the ship that may lead to largest profit or largest cargo or with largest number of fractional variables.

Primal heuristics

- Improve the formulation: the goal of improving the lower bounds or solutions whose real variables are closer to be integer
- Use heuristics within the IP framework. Goal: finding good feasible solutions
  - construction heuristics
  - improvement heuristics

The following heuristics can be applied at each node of a branch-and-cut/bound tree

Truncated MIP

Run branch-and-cut/bound for a fixed amount of time and return the best solution when time exceeds.

Diving

Carry out a depth-first search in branch-and-cut/bound tree.
At each node, fix variables that take integer values in the LP relaxation and branch on the others

- LP-driven dives: fix the variable that is closest to integer
- IP-driven or guided dives: given an incumbent solution, choose the variable to be fixed next and assign it the value it has in the incumbent

These are typically already implemented in MIP systems
LP or incumbent solutions are the guide.

LP-and-fix or Cut-and-Fix

Fix everything that is integer and solve the resulting MIP^{LP-FIX}

Either the new problem is infeasible or it provides and LP-and-fix heuristic solution

(best solutions if formulation is tight and has few fractional variables)

Relax-and-fix

Partition the variables into R disjoint sets and solve sequentially R MIPs, MIP^r with $1 \leq r \leq R$.
(For example partitions correspond to variables of a tank, machine, product family, location, most often time periods)

- In the first MIP^1 impose integrality in the first partition and relax all the others
- Fix the variables in the first partition at the values found in MIP^1
- In the subsequent MIP^r, for $2 \leq r \leq R$ additionally fix the values of the variables of the $r-1$-th partition at the optimal value from MIP^{r-1} and add integrality restriction for the variables in the r-th partition.
- Either MIP^r is infeasible for some r and the heuristic has failed or else the solution found at $r = R$ is a relax-and-fix heuristic solution

(allow overlap between the partitions may be a good idea)
(Note: only MIP^1 is a valid lower bound to the MIP)

Exchange

Improvement version of the relax-and-fix heuristic

At each step r with $1 \leq r \leq R$ the MIP solved is obtained by fixing at their value in the best solution all the variables in the set $r-1$ partitions and imposing integrality to the variables in the r partition

Relaxation Induced Neighborhood Search

Explore neighborhood between LP solution $\hat{s}$ and best known feasible solution $\bar{s}$

Fix a variable that has same value in $\hat{s}$ and $\bar{s}$ and solve the IP problem

Either the solution found is infeasible or it is not found within a time limit so the heuristic has failed or the solution found is an heuristic solution

Local Branching

- The procedure is in the spirit of heuristic local search paradigm.
- The neighborhoods are obtained through the introduction in the MIP model of (invalid) linear inequalities called local branching cuts.
- Takes advantage of black box efficient MIP solvers.

In branch and bound most often unclear how to fix variables
➜ Idea: soft fixing

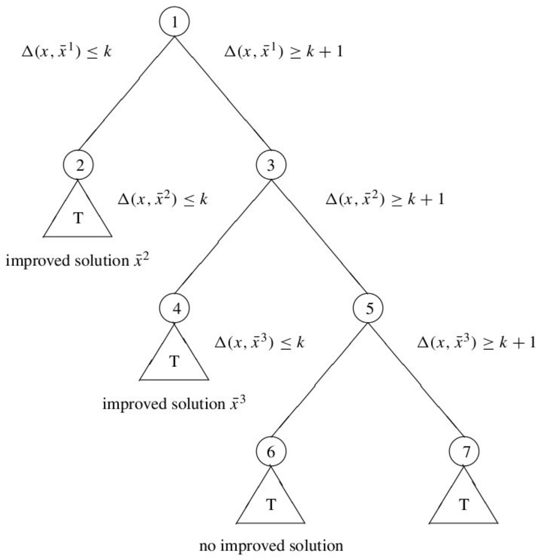
Given a feasible solution $\bar{x}$ let $\bar{O} := \{i \in B : \bar{x}_i = 1\}$.
Define the k-opt neighborhood $\mathcal{N}(\bar{x}, k)$ as the set of feasible solutions satisfying the additional local branching constraint:

$$\Delta(x, \bar{x}) := \sum_{i \in \bar{O}} (1 - x_i) + \sum_{i \in B \setminus \bar{O}} x_i \leq k$$

Δ counts number of flips

Partition at the branching node:

$\Delta(x, \bar{x}) \leq k$ (left branching) **or** $\Delta(x, \bar{x}) \geq k + 1$ (right branching)



- The idea is that the neighborhood $N(\bar{x}, k)$ corresponding to the left branch must be "sufficiently small" to be optimized within short computing time, but still "large enough" to likely contain better solutions than x.

- According to computational experience, good values for k are in $[10, 20]$

This procedure coupled with an efficient MIP solver (subgradient optimization of Lagrangian multipliers) was shown able to solve very large problems with more than 8000 variables.

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

[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)
- $a_{ti}^l = \{0, 1\}$ indicates if leg i is covered by $l \in S_t$
- π_{ti} profit of covering leg i with aircraft of type i

$$\pi_t^l = \sum_{i \in L_t} \pi_{ti} a_{ti}^l \qquad \text{for } l \in S_t$$

- P set of airports, P_t set of airports that can accommodate type t
- o_{tp}^l and d_{tp}^l 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

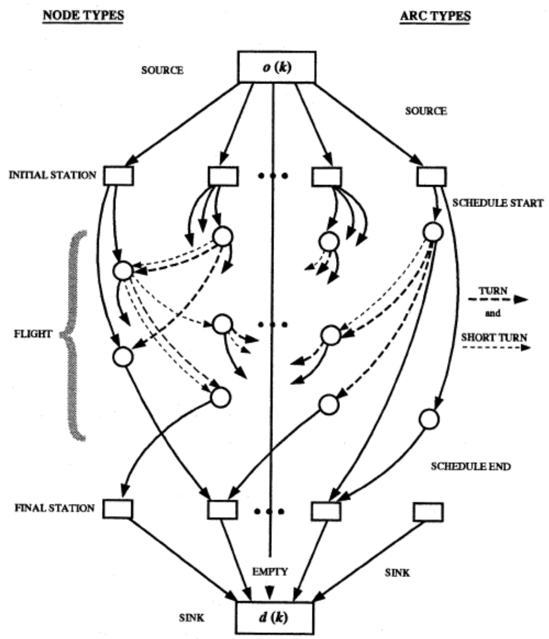
$$\sum_{l \in S_t} (o_{tp}^l - d_{tp}^l) x_t^l = 0 \qquad \forall t \in T;\ p \in P$$

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.



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

subject to:

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

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

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

$$\sum_{i:(i,j) \in A^k} X_{ij}^k - \sum_{i:(j,i) \in A^k} X_{ji}^k = 0$$
$$\forall k \in K, \forall j \in V^k \setminus \{o(k), d(k)\}, \tag{12}$$

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

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

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

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

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