DM87
SCHEDULING,
TIMETABLING AND ROUTING

Lecture 16

Transportation Timetabling

Marco Chiarandini

Outline

Outline

Traveling tournament problem

Input: A set of teams $T = \{1, \dots, n\}$; D an $n \times n$ integer distance matrix with elements d_{ij}; l, u integer parameters.

Output: A double round robin tournament on the teams in T such that

1. the length of every home stand and road trip is between l and u inclusive
2. the total distance traveled by the teams is minimized

A metaheuristic approach: Simulated Annealing

Constraints:

- DRRT constraints always satisfied (enforced)
- constraints on repeaters (i may not play at j and host j at home in consecutive slots) are relaxed in soft constraints

Objective made of:

- total distance
- a component to penalize violation of constraints on repeaters

➜ Penalties are dynamically adjusted to prevent the algorithm from spending too much time in a space where the soft constraints are not satisfied.

Neighborhood operators:

- Swap the positions of two slots of games
- Swap the schedules of two teams (except for the games when they play against)
- Swap venues for a particular pair of games (i at j in slot s and j at i in slot s' becomes i at j in slot s' and j at i in slot s)

Use reheating in SA.

Outline

Outline

Problems

- Tanker Scheduling
- Aircraft Routing and Scheduling
- Train Timetabling

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
- Local branching
- Branch and price (column generation)
- Subgradient optimization of Lagrangian multipliers (solution without Simplex)

Planning problems in public transport

Phase:	Planning	Scheduling	Dispatching
Horizon:	Long Term	Timetable Period	Day of Operation
Objective:	Service Level	Cost Reduction	Get it Done
Steps:	Network Design Line Planning Timetabling Fare Planning	Vehicle Scheduling Duty Scheduling Duty Rostering	Crew Assignment Delay Management Failure Management Depot Management

Master Schedule $\xrightarrow{\text{Dynamic Management}}$ Conflict resolution

[Borndörfer, Grötschel, Pfetsch, 2005, ZIB-Report 05-22]

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

Two phase approach:

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

2. 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:

- select 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)

- select one ship and branch on its itineraries
  select the ship that may lead to largest profit or largest cargo or with largest number of fractional variables.

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 the previous branch and bound, unclear how to fix variables
➜ Idea: soft fixing

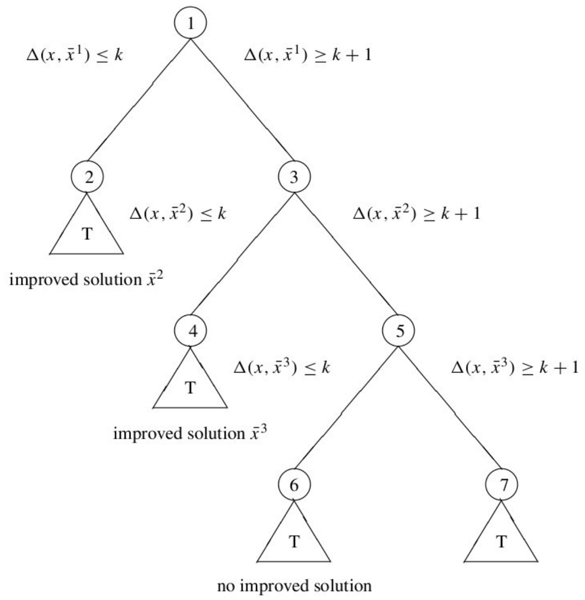
Given a feasible solution $\bar{x}$ let $\bar{O} := \{i \in B : \bar{x}_i = 1\}$.
Define the k-opt neighborhood $\mathcal{N}(\bar{x}, k)$ as the set of feasible solutions satisfying the additional local branching constraint:

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

(Δ counts the number of flips)

Partition at the branching node:

$\Delta(x, \bar{x}) \le k$ (left branching) **or** $\Delta(x, \bar{x}) \ge k + 1$ (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 (DARS)

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

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

Maximum number of aircraft of each type:

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

Each flight leg is covered exactly once:

$$\sum_{t \in T} \sum_{l \in S_t} a^l_{ti} x^l_t = 1 \qquad \forall i \in L$$

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

$$\sum_{l \in S_t} (o^l_{tp} - d^l_{tp}) x^l_t = 0 \qquad \forall t \in T;\ p \in P$$

Maximize total anticipate profit

$$\max \sum_{t \in T} \sum_{l \in S_t} \pi^l_t x^l_t$$

Solution Strategy: branch-and-price (branch-and-bound + column generation)

- 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.

Train Timetabling

Input:

- Corridors made up of two independent one-way tracks
- L links between $L+1$ stations.
- T set of trains and T_j, $T_j \subseteq T$, subset of trains that pass through link j

Output: We want to find a periodic (eg, one day) timetable for the trains on one track (the other can be mirrored) that specifies:

- y_{ij} = time train i enters link j
- z_{ij} = time train i exists link j

such that specific constraints are satisfied and costs minimized.

Constraints:

- Minimal time to traverse one link
- Minimum stopping times at stations to allow boarding
- Minimum headways between consecutive trains on each link for safety reasons
- Trains can overtake only at train stations
- There are some "*predetermined*" upper and lower bounds on arrival and departure times for certain trains at certain stations

Costs due to:

- deviations from some "*preferred*" arrival and departure times for certain trains at certain stations
- deviations of the travel time of train i on link j
- deviations of the dwelling time of train i at station j

Solution Approach

- All constraints and costs can be modeled in a MIP with the variables: y_{ij}, z_{ij} and $x_{ihj} = \{0, 1\}$ indicating if train i precedes train h
- Two dummy trains T' and T'' with fixed times are included to compact and make periodic
- Large model solved heuristically by decomposition.
- Key Idea: insert one train at a time and solve a simplified MIP.
- In the simplified MIP the order in each link of trains already scheduled is maintained fixed while times are recomputed. The only order not fixed is the one of the new train inserted k (x_{ihj} simplifies to x_{ij} which is 1 if k is inserted in j after train i)

Overall Algorithm

Step 1 (Initialization)
Introduce two "dummy trains" as the first and last trains in T_0

Step 2 (Select an Unscheduled Train) Problem) Select the next train k through the train selection priority rule

Step 3 (Set up and preprocess the MIP) Include train k in the set T_0
Set up MIP(K) for the selected train k
Preprocess MIP(K) to reduce number of 0–1 variables and constraints

Step 4 (Solve the MIP) Solve MIP(k). If algorithm does not yield feasible solution STOP.
Otherwise, ass train k to the list of already scheduled trains and fix for each link the sequences of all trains in T_0.

Step 5 (Reschedule all trains scheduled earlier) Consider the current partial schedule that includes train k.
For each train $i \in \{T_0 - k\}$ delete it and reschedule it

Step 6 (Stopping criterion) If T_0 consists of all train, then STOP otherwise go to Step 2.

Further References

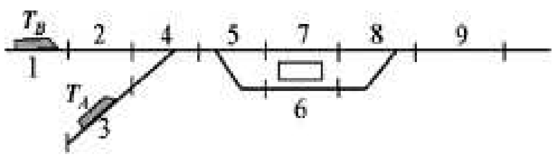
M. Fischetti and A. Lodi, Local Branching, Mathematical Programming, 98(1-3), pp 23-47, 2003.

C. Barnhart, P. Belobaba, A. Odoni, Applications of Operations Research in the Air Transport Industry, Transportation Science, 2003, vol. 37, issue 4, p 368.

Exercise

Short-term Railway Traffic Optimization

Conflict resolution problem (CRP) with two trains traveling at different speed:



Block sections: track segment with signals (fixed NS54)

At time $t = 0$ there are two trains in the network.

Train T_A is a slow train running from block section 3 to block section 9, and stopping at platform 6. It can enter a block section only if the signal aspect is yellow or green.

Train T_B is a fast train running from block section 1 to block section 9 through platform 7 without stopping. It can enter a block section at high speed only if the signal aspect is green.

A blocking job shop model:

Given:

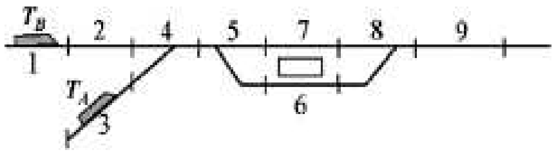
- Passing of trains in a block ➜ Operation
- Traverse (running) times ➜ Processing times
- Itinerary of the train ➜ Precedences
- Safety standards between blocks ➜ Setup times

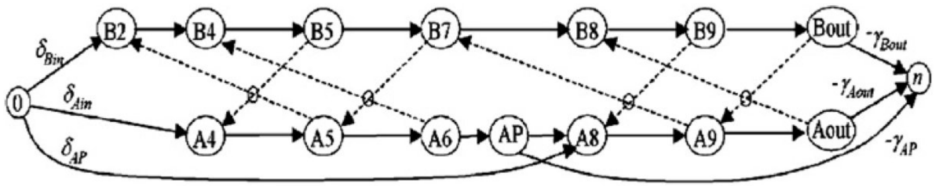
Task:

- Find the starting times $t_1, t_2, \ldots, t_n$, (or the precedences) such that:
  - No conflict (two trains on the same track segment at the same time)
  - Minimize maximum delay (or disrupt least possible the original plan)

- Signals and train speed constraints can be modeled as blocking constraints ➜ Alternative graph

- Speed and times goals can be modeled with time lags





- δ_{AP} scheduled departing time from platform P

- $-\gamma_{AP}$ planned due dates