Outline

Lecture 21
Timetabling in Transportation

Marco Chiarandini

Planning problems in public transport

<table>
<thead>
<tr>
<th>Phase</th>
<th>Planning</th>
<th>Scheduling</th>
<th>Dispatching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon</td>
<td>Long Term</td>
<td>Timetable Period</td>
<td>Day of Operation</td>
</tr>
<tr>
<td>Objective</td>
<td>Service Level</td>
<td>Cost Reduction</td>
<td>Get it Done</td>
</tr>
<tr>
<td>Steps</td>
<td>Network Design</td>
<td>Vehicle Scheduling</td>
<td>Crew Assignment</td>
</tr>
<tr>
<td></td>
<td>Line Planning</td>
<td>Duty Scheduling</td>
<td>Delay Management</td>
</tr>
<tr>
<td></td>
<td>Timetabling</td>
<td>Duty Rostering</td>
<td>Failure Management</td>
</tr>
<tr>
<td></td>
<td>Fare Planning</td>
<td></td>
<td>Depot Management</td>
</tr>
</tbody>
</table>

↑ Master Schedule ← Dynamic Management → Conflict resolution

[Borndörfer, Grötschel, Pfetsch, 2005, ZIB-Report 05-22]
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 (e.g., 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 \( i \) exits 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: \( y_{ij}, z_{ij}, x_{ihj} = \{0, 1\} \) 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 in \( T_0 \) two “dummy trains” as first and last trains
- **Step 2** (Select an Unscheduled Train) Select the next train \( k \) through the train selection priority rule
- **Step 3** (Set up and preprocess the MIP) Include train \( k \) in 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, add 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.