Optimal allocation of buffer times to increase train schedule robustness

Pavle Kecman

KAJT dagarna, 27 April 2016
Dala Storsund
Contents

- Introduction and motivation
- Knapsack problem approach
- Parameter computation
- Case study and results
Capacity4Rail WP3.2

- Simulation and models to evaluate enhanced capacity
- The aim of this task is to evaluate existing tools for their suitability to assess and **improve** capacity utilization
- ”Capacity depends on the way it is utilised” (UIC 406)
- Timetabling (and traffic control) determine the way capacity is utilised
Timetabling – C4R Perspective

[Graph showing the relationship between UIC capacity costs, delay, and the number of trains.]

- Current state
- Improved tactical and operational control
- Optimal tactical and operational control
Research question

Robust timetabling enable more trains to run with the desired quality of service

Diagram showing the relationship between the number of trains and operational costs, with an arrow pointing to the area labeled "Robust timetabling."
Delay propagation - microscopic
Delay propagation - macroscopic

Solution - Buffer times

- Critical block headway
- Minimum line headway
- Buffer time

Diagram showing the relationship between distance and time with various blocks indicating different times and distances.
Another problem
Existing solutions

Robust optimisation:
+ Attacks general problems
- Very difficult to solve

Domain knowledge used to relax the problem:
+ Emma: Critical points
+ Fahimeh: Travel time dependent buffering

Produce a general solution using the domain knowledge
Problem definition

- Input – Timetable A
  - Number of trains
  - Scheduled running and dwell times
  - Fixed train sequence
  - Time window constraint
- Output – Timetable B
  - All properties of Timeble A are kept
  - Buffer times (re)distributed to increase robustness
Knapsack problem (1/2)

- Hikers wants to go on a trip
- The backpack is small, no more than 10 kg of things in the bag
- He has prepared a list of items that he would like to bring on a trip
- Water, bread, cans, maps & compass, laptop, trousers, jacket, socks & underwear, knife and cutlery, sweater, tent, sleeping bag
Knapsack problem (2/2)

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight [kg]</th>
<th>Utility: 1 (not useful) to 10 (very useful)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cans</td>
<td>2.2</td>
<td>7</td>
</tr>
<tr>
<td>Water</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Tent</td>
<td>3.5</td>
<td>8</td>
</tr>
<tr>
<td>Food</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Jacket</td>
<td>0.5</td>
<td>7</td>
</tr>
<tr>
<td>Maps &amp; compass</td>
<td>0.1</td>
<td>10</td>
</tr>
<tr>
<td>Sleeping bag</td>
<td>0.8</td>
<td>9</td>
</tr>
<tr>
<td>Laptop</td>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>Trousers</td>
<td>0.3</td>
<td>6</td>
</tr>
<tr>
<td>Socks &amp; underwear</td>
<td>0.2</td>
<td>9</td>
</tr>
<tr>
<td>Knife &amp; cutlery</td>
<td>0.5</td>
<td>9</td>
</tr>
<tr>
<td>Sweater</td>
<td>0.5</td>
<td>5</td>
</tr>
</tbody>
</table>
# Knapsack problem (2/2)

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight [kg]</th>
<th>Utility: 1 (not useful) to 10 (very useful)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cans</td>
<td>2.2</td>
<td>7</td>
</tr>
<tr>
<td>Water</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Tent</td>
<td>3.5</td>
<td>8</td>
</tr>
<tr>
<td>Food</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Jacket</td>
<td>0.5</td>
<td>7</td>
</tr>
<tr>
<td>Maps &amp; compass</td>
<td>0.1</td>
<td>10</td>
</tr>
<tr>
<td>Sleeping bag</td>
<td>0.8</td>
<td>9</td>
</tr>
<tr>
<td>Laptop</td>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>Trousers</td>
<td>0.3</td>
<td>6</td>
</tr>
<tr>
<td>Socks &amp; underwear</td>
<td>0.2</td>
<td>9</td>
</tr>
<tr>
<td>Knife &amp; cutlery</td>
<td>0.5</td>
<td>9</td>
</tr>
<tr>
<td>Sweater</td>
<td>0.5</td>
<td>5</td>
</tr>
</tbody>
</table>

15.1 kg! Which items to bring to maximise utility?
Knapsack problem for buffer times (1/2)

Timetable compression UIC 406-ish
Knapsack problem for buffer times (1/2)
Knapsack problem for buffer times (2/2)

• How to coordinate multiple sections?
• How to prioritize items (candidates)?
• Marginal profit: is the second minute (time unit) of buffer as valuable as the first? How about the third?
Multidimensional Knapsack Problem

Figure 3: Illustrative example for the knapsack capacity

Figure 4: Illustrative example for the knapsack capacity
Prioritisation

- Efficient graph algorithms can be used to compute for each candidate:
  - 1. Delay impact \((I)\): if the candidate is delayed for \(D\), how many events will have secondary delay?
  - 2. Delay sensitivity \((S)\): how many other events can be delayed for \(D\) so that it propagates to the candidate?

- The bigger \(I\) and \(S\), the bigger the profit for including the candidate!
Marginal profit

Marginal profit from including an additional minute depends on the number of already included minutes of the same buffer.
Case study
Case study
Case study

Connections + 30min frequency
Experimental setup

• 3 schedules generated by using different parameter setup
• 500 hundred primary delay scenarios generated
• All departured events are delayed with a uniform distribution upto 10 minutes
• On average 28 events have primary delay
• Total primary delay 150.14 min on average
• Deterministic delay propagation algorithm computed secondary delays in each scenario for each timetable (500 x 4 experiments in total)
Results

- Upto 11% decrease in secondary delay

<table>
<thead>
<tr>
<th></th>
<th>Total delay [min]</th>
<th>Average delay per event [min]</th>
<th>Delay per 1 min prim. [min]</th>
<th>Delay per init. delayed event [min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>1146.70</td>
<td>8.49</td>
<td>8.87</td>
<td>40.95</td>
</tr>
<tr>
<td>TB 0-1</td>
<td>1034.20</td>
<td>7.66</td>
<td>7.99</td>
<td>36.94</td>
</tr>
<tr>
<td>TB Bounded</td>
<td>1033.80</td>
<td>7.65</td>
<td>7.98</td>
<td>36.92</td>
</tr>
<tr>
<td>TTB</td>
<td>1017.20</td>
<td>7.53</td>
<td>7.84</td>
<td>36.32</td>
</tr>
</tbody>
</table>

- In upto 87% cases, original timetable performs worse
Next steps

• Prioritisation of buffering based on historical data
• Computational experiments on networks
• More details about the approach available soon:
Thank you for your attention

www.liu.se