Marshaling and shunting yards

- Bundling different commodities with close origins and destinations
Problem setting

- Existing SBB Cargo network
  - 2 inner marshaling yards
  - 3 border marshaling yards
  - Approx. 70 shunting yards – 50 can be changed
- Determine optimal number and locations of the yards
Related problems

- Hub location problem (HLP)
  - Missing track capacities and hub operation costs
- Multicommodity flow problem (MFP)
  - Missing hub capacities and operation costs
- Multicommodity network design problem (MNDP)
  - Missing hub types and operation costs
Problem definition

• Extension of the HLP

• Network elements:
  • \( N \) – Set of stations, including potential marshaling and shunting yards
  • \( A \) – Set of direct links between the stations
  • \( K \) – Set of transported commodities each described with the origin, destination, weight and number of wagons
Problem definition (cont.)

• Constants:
  • $d_{ij}$ – Shortest distance between nodes $i$ and $j$
  • $k_o$ – Origin of commodity $k$
  • $k_d$ – Destination of commodity $k$
  • $w^k$ – Weight of commodity $k$
  • $P_W$ – Cost of transporting a weight unit of any commodity per distance unit
  • $P_D$ – Cost of the locomotive and driver per distance unit
  • $v^k$ – Number of wagons of commodity $k$
  • $V_{max}$ – Maximum number of wagons in a train
  • $S$ – Cost of shunting one wagon in a shunting yard
  • $M$ – Cost of shunting one wagon in a marshaling yard
Problem definition (cont.)

- Transportation cost per distance unit:
  
  - Unbundled commodity:
    
    \[ c^k = w^k P_W + P_D \quad \forall k \in K \]

  - Bundled commodity:
    
    \[ c_r^k = w^k P_W + \frac{V^k}{V_{max}} P_D \quad \forall k \in K \]
Problem definition (cont.)

- **Objective function:**
  \[
  \min \sum_{k \in K} \left( \sum_{i \in N} \sum_{j \in N} X_{ij}^k \left( (d_{ik} + d_{jk})c^k + d_{ij}c_r^k + (s_i S + m_i M + s_j S + m_j M)v^k \right) + (1 - z^k) \right) d_{ko}d_{kd}c^k
  \]

- **Variables:**
  - \(X_{ij}^k\) – Determines if commodity \(k\) is transported via hubs \(i\) and \(j\)
  - \(s_i\) – Determines if the node \(i\) is a shunting yard
  - \(m_i\) – Determines if the node \(i\) is a marshaling yard
  - \(z^k\) – Determines if the commodity \(k\) is transported bundled
Problem definition (cont.)

- Node type constraints:

  \[ r_i + s_i + m_i = 1, \quad \forall i \in N \]

- Commodity shunting constraints:

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

  \[ 2X_{ij}^k \leq s_i + m_i + s_j + m_j, \quad \forall k \in K, \forall i \in N, \forall j \in N \]
• Node capacity constraints:

\[
\sum_{k \in K} \sum_{j \in N} X_{ij}^k v^k + \sum_{k \in K} \sum_{j \in N} X_{ji}^k v^k = a_i, \quad \forall i \in N
\]

\[
a_i \leq r_i M + s_i C_S + m_i C_M, \quad \forall i \in N
\]

Variables:
• \( a_i \) – Required capacity of the node \( i \)

Constants:
• \( M \) – Sufficiently large number
• \( C_S \) – Maximum capacity of a shunting yard
• \( C_M \) – Maximum capacity of a marshaling yard
Problem definition (cont.)

• Arc capacity constraints:

\[
\sum_{k \in K} v^k \left( \sum_{i \in N} \sum_{j \in N} x_{ij}^k \left( b_{lm}^{k,i} + b_{lm}^{k,j} + b_{lm}^{ij} \right) + (1 - z^k)b_{lm}^{k,0,k_d} \right) \leq u_{lm}, \forall (l, m) \in A
\]

Constants:

• \( u_{lm} \) – Capacity of the arc (l, m)

• \( b_{lm}^{ij} \) – Determines if arc (l, m) belongs to the shortest path between i and j
Problem definition (cont.)

- Maximum number of yards:
  \[
  \sum_{i \in N} s_i \leq L_S \quad \sum_{i \in N} m_i \leq L_M
  \]

- Integrality constraints:
  \[
  X_{ij}^k \in \{0,1\}, \quad \forall k \in K, \forall i \in N, \forall j \in N
  \]
  \[
  z^k \in \{0,1\}, \quad \forall k \in K
  \]
  \[
  r_i \in \{0,1\}, \quad \forall i \in N
  \]
  \[
  s_i \in \{0,1\}, \quad \forall i \in N
  \]
  \[
  m_i \in \{0,1\}, \quad \forall i \in N
  \]
Problem size

- Size of the SBB Cargo network:
  - Approx. 2100 stations
  - Approx. 2500 direct links
  - Over 65000 commodities
Heuristic algorithm

- Hub location:
  - Adaptive large neighborhood search
  - Variable neighborhood search
Heuristic algorithm - Neighborhoods

- Select the busiest station close to the MY and exchange their locations

- Select the least used MY and convert it into SY

- Select SY with fully utilized, maximum capacity and convert it into MY
Heuristic algorithm - Neighborhoods

- Select SY with most unused capacity and decrease it

- Select SY with fully utilized, below-maximum capacity and increase it
Select underused SY with minimum capacity and convert it into a regular station.

Select frequently used regular station and convert it into a SY with minimum capacity.
Heuristic algorithm

- Commodity routing:
  - Prioritized assignment algorithm

  Select commodity by priority

  Path alternatives
  1. Shortest path
  2. Via marshaling yards
  3. Via shunting yards

  Path selection
  Cheapest, available path
Heuristic algorithm - Path alternatives

• Direct (shortest) path
  • Unbundled commodity

• Via marshaling yards
  • Bundled commodity
Heuristic algorithm - Path alternatives

- Via shunting yards
  - Bundled commodity

- Via one marshaling and one shunting yard
  - Bundled commodity
Heuristic algorithm – development details

- Developed algorithm is very flexible:
  - Easily extendable with additional neighborhood operators, i.e. network transformations
  - Easy definition of specific initial network states, e.g. all marshaling yards closed, several additional marshaling yards open, etc.
Preliminary results

- Network states with potential transportation cost reduction identified with two strategies (thus far):
  1. S1: Allowing opening of new marshaling yards
  2. S2: Disallowing opening of new marshaling yards
Preliminary results (cont.)

- Best resulting networks:

<table>
<thead>
<tr>
<th>Strat.</th>
<th>New MY</th>
<th>Rem. MY</th>
<th>Mov. MY</th>
<th>Total MY</th>
<th>New SY</th>
<th>Rem. SY</th>
<th>Total SY</th>
<th>Cost reduct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orig. net.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>50</td>
<td>-</td>
</tr>
<tr>
<td>S1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>46</td>
<td>8.505%</td>
</tr>
<tr>
<td>S2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>5</td>
<td>64</td>
<td>1.857%</td>
</tr>
</tbody>
</table>

- Daily transportation cost in the original network: over 38 Million CHF
- Running time: approx. 9h
Results discussion

- Costs of transportation are dominant over yard operation costs
- Cost of yard maintenance is not taken into account
  - This cost contributes to reducing the number of yards and their size
  - Opening new yards will be less favored by the algorithm
  - Could be included in another case study
- New yards can be near the existing ones
  - E.g. in S1, new MY Territet is opened close to Lausanne MY
  - The objective function should penalize this situation
Conclusions

• Developed algorithm explores various network changes, their combinations and their influence to the transportation costs
  • Flexible, easily extendable algorithm
• The algorithm identified network changes resulting in transportation cost reduction
• The objective function should be extended with the real costs of maintenance of the marshaling and shunting yards
  • Relevant change in the algorithm result
Future work

• Include penalty for having two yards near each other

• Solve the problem exactly on the subset of input data
  • To benchmark the heuristic result

• Implement visualization of the results

• Develop models based on MFP and MNDP
  • To compare results
  • If the current formulation cannot be solved exactly
Thank you!

Questions?

nikola.obrenovic@epfl.ch
References


