Recovery-Robust Platforming
by Network Buffering

A. Caprara\textsuperscript{1} \quad L. Galli\textsuperscript{1} \quad S. Stiller\textsuperscript{2} \quad P. Toth\textsuperscript{1}

\textsuperscript{1}University of Bologna
\textsuperscript{2}Technische Universität Berlin

RailZurich2009, February 11 - 13
Outline

Train Platforming Problem
  In and Out
  TPP deterministic model
Outline

Train Platforming Problem
  In and Out
  TPP deterministic model

Recovery-Robust Train Platforming
  Definitions
  Delay propagation network
  Buffers linking constraints
Outline

Train Platforming Problem
  In and Out
  TPP deterministic model

Recovery-Robust Train Platforming
  Definitions
  Delay propagation network
  Buffers linking constraints

Computational results
Outline

Train Platforming Problem
   In and Out
   TPP deterministic model

Recovery-Robust Train Platforming
   Definitions
   Delay propagation network
   Buffers linking constraints

Computational results

References
Problem definition

The objective of train platforming is assigning trains to platforms in a railway station.
Problem definition

The objective of train platforming is assigning trains to platforms in a railway station. Platforming is carried out:
Problem definition

The objective of train platforming is assigning trains to platforms in a railway station. Platforming is carried out:

- for a specific railway station
The objective of train platforming is assigning trains to platforms in a railway station. Platforming is carried out:

- for a specific railway station
- after the timetable has been defined
In and Out

Input

- Train schedule: arrival, departure times, directions and allowed shifts
- Railway station topology: platforms, paths and directions
In and Out

Input

- Train schedule: arrival, departure times, directions and allowed shifts
- Railway station topology: platforms, paths and directions

Output

- Assign each train a platform and two paths for arrival and departure s.t. no operational constraint is violated
The train schedule of a railway station contains info on arrival and departure times, directions and allowed shifts of each train passing through it.
Railway station topology

The topology of a railway station includes platforms, paths and directions.
Resources and operational constraints

Platform conflicts are forbidden, path conflicts are allowed to some extent.

Figure: Platform and path.

A pattern $P$ for a train $t$ is a 5-tuple defining: platform, arrival/departure paths and shifts. Operational constraints can be expressed using an incompatibility graph among patterns.

Figure: Incompatibility graph.
TPP deterministic model

\[
\min \sum_{t \in T} \sum_{P \in \mathcal{P}_t} c_{t,P} \ x_{t,P} \tag{1}
\]

s.t.

\[
\sum_{P \in \mathcal{P}_t} x_{t,P} = 1, \quad t \in T \tag{2}
\]

\[
\sum_{(t_1,P_1) \in K} x_{t_1,P_1} + \sum_{(t_2,P_2) \in K} x_{t_2,P_2} \leq 1, \quad (t_1,t_2) \in T^2, K \in \mathcal{K}(t_1,t_2) \tag{3}
\]

\[
x_{t,P} \in \{0,1\}, \quad t \in T, \ P \in \mathcal{P}_t \tag{4}
\]

Details in Caprara et al. 2007 [2].
Robust Optimization

Robust Optimization finds best solutions, which are feasible for all likely scenarios.
Robust Optimization

Robust Optimization finds best solutions, which are feasible for all likely scenarios.

Pros

- no knowledge of the underlying distribution is required
- models are easier to solve
Robust Optimization

Robust Optimization finds best solutions, which are feasible for all likely scenarios.

Pros

- no knowledge of the underlying distribution is required
- models are easier to solve

Cons

*Strict robustness* is generally overconservative, because:

- solutions must cope with every likely scenarios without any recovery
- it is unable to account for limits to the sum of all disturbances
Informally speaking, a solution to an optimization problem is called *recovery robust* if it can be adjusted to all likely scenarios by limited recovery action. Thus a recovery-robust solution provides a service guarantee (Liebchen *et al.* 2007 [3]).
Robust Network Buffering

We are interested in the special case in which the recovery problem is a delay ($y_i^a$) propagation in some directed graph $N$, which is buffered on the arcs by means of $f$ against disturbances on the arcs $a \in A(N)$. Denoting by $A(N)$ the set of arcs in $N$, we get:

$$\min_{f \in P} c(f)$$

s.t. $\forall a \in A(N) \ \exists y^a \in \mathbb{R}^{\mid A(N)\mid}$ :

$$f(i,j) + y_i^a - y_i^a \geq \Delta \cdot \chi_a((i,j)), \quad a = (i,j) \in A(N)$$

$$D - d' y^a \geq 0$$

Details in Liebchen et al. 2007 [3].
Delay propagation network

The platforming gives rise in a natural way to a network in which the delay caused by disturbances propagates. This delay propagation network is a directed acyclic graph in which each vertex represents the delay of a particular train for a particular resource.
Delay propagation network

Each train has three associated vertices in this graph: (i) a for the arrival path, (ii) s for the stopping platform, and (iii) d for the departure path, corresponding to the delay (with respect to the nominal schedule) with which it will free up each of the three resources assigned to it.
Example

Assume in the nominal schedule the first train frees up the platform at 10:00, the second train occupies it at 10:05 and frees it up at 10:10. Then a delay of more than 5 minutes for the first train results in a delay also for the second.

Figure: Example
Delay propagation network model

\[
D \geq \sum_{t \in T} (a_t^\xi + s_t^\xi + d_t^\xi), \quad \xi \in \{\delta_t|t \in T\} \cup \{\delta'_t|t \in T\}
\]

(5)

\[
a_t^\xi \geq \delta_t^\xi, \quad t \in T
\]

(6)

\[
s_t^\xi \geq a_t^\xi + \delta'_t^\xi, \quad t \in T
\]

(7)

\[
d_t^\xi \geq s_t^\xi, \quad t \in T
\]

(8)

\[
m_{t_2}^\xi \geq h_{t_1}^\xi - f(h_{t_1}, m_{t_2}), \quad a = (h_{t_1}, m_{t_2}) \in A(N)
\]

(9)
Buffers linking constraints

A straightforward link between the buffer value of a given arc \( a \in A(N) \) associated with train pair \( (t_1, t_2) \in T^2 \) and the choice of patterns for the given pair of trains is the following:

\[
\sum_{P_1 \in P_{t_1}} \sum_{P_2 \in P_{t_2}} c_{P_1,P_2,a} x_{t_1,P_1} x_{t_2,P_2}
\]

where \( c_{a,P_1,P_2} \) is a constant associated to arc \( a \) and to the corresponding choice of patterns \((P_1, P_2)\) for trains \((t_1, t_2)\).
Buffers linking constraints

\[ f_a \leq \sum_{P_1 \in \mathcal{P}_{t_1}} \alpha_{P_1}^a x_{t_1,P_1} + \sum_{P_2 \in \mathcal{P}_{t_2}} \beta_{P_2}^a x_{t_2,P_2} - \gamma^a, \quad a \in A(N), \quad (\alpha, \beta, \gamma) \in \mathcal{F}_a \quad (10) \]

Following Caprara \textit{et al.} 2007 [2], the separation of Constraints (10) is done by a sort of polyhedral brute force, given that, for each pair of trains \( t_1, t_2 \), and for each arc \( a \in A(N) \) the number of vertices in \( Q_{t_1,t_2,a} \) is small. Specifically, \( Q_{t_1,t_2,a} \) has \(|\mathcal{P}_{t_1}| \cdot |\mathcal{P}_{t_2}| \) vertices and lies in \( \mathbb{R}^{(|\mathcal{P}_{t_1}| + |\mathcal{P}_{t_2}|) + 1} \), so we can separate over it by solving an LP with \(|\mathcal{P}_{t_1}| \cdot |\mathcal{P}_{t_2}| \) variables and \(|\mathcal{P}_{t_1}| + |\mathcal{P}_{t_2}| + 1 \) constraints.
## Computational results: Palermo C.Le.

<table>
<thead>
<tr>
<th>time window</th>
<th># trains n.p.</th>
<th>$D_{\text{nom}}$</th>
<th>CPU time nom (sec)</th>
<th>$D_{\text{RR}}$</th>
<th>CPU time RR (sec)</th>
<th>Diff. $D$</th>
<th>Diff. $D$ in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>646</td>
<td>7</td>
<td>479</td>
<td>46</td>
<td>167</td>
<td>25.85</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>729</td>
<td>7</td>
<td>579</td>
<td>3826</td>
<td>150</td>
<td>20.58</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>487</td>
<td>6</td>
<td>356</td>
<td>143</td>
<td>131</td>
<td>26.90</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>591</td>
<td>6</td>
<td>384</td>
<td>228</td>
<td>207</td>
<td>35.03</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>710</td>
<td>9</td>
<td>516</td>
<td>2217</td>
<td>194</td>
<td>27.32</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>560</td>
<td>7</td>
<td>480</td>
<td>18</td>
<td>80</td>
<td>14.29</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>465</td>
<td>11</td>
<td>378</td>
<td>64</td>
<td>87</td>
<td>18.71</td>
</tr>
</tbody>
</table>

**Table:** Results for Palermo Centrale
## Computational results: Genova P.Princ.

<table>
<thead>
<tr>
<th>time window</th>
<th># trains n.p.</th>
<th>$D$ nom</th>
<th>CPU time nom (sec)</th>
<th>$D$ RR</th>
<th>CPU time RR (sec)</th>
<th>Diff. $D$</th>
<th>Diff. $D$ in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>630</td>
<td>9</td>
<td>516</td>
<td>18190</td>
<td>114</td>
<td>18.10</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>838</td>
<td>11</td>
<td>624</td>
<td>3177</td>
<td>214</td>
<td>25.54</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>888</td>
<td>7</td>
<td>509</td>
<td>2495</td>
<td>379</td>
<td>42.68</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>895</td>
<td>8</td>
<td>657</td>
<td>9940</td>
<td>238</td>
<td>26.59</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>616</td>
<td>5</td>
<td>405</td>
<td>37</td>
<td>211</td>
<td>34.25</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>516</td>
<td>5</td>
<td>373</td>
<td>14</td>
<td>143</td>
<td>27.71</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>431</td>
<td>5</td>
<td>219</td>
<td>8</td>
<td>212</td>
<td>49.19</td>
</tr>
</tbody>
</table>

**Table:** Results for Genova Piazza Principe
References


- Liebchen C., Stiller S. Delay Resistant Timetabling Technical Report 0066, EU ARRIVAL project.