Recoverable-Robust Timetables for Trains on Single-Line Corridors

Work partially supported by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL)

Gianlorenzo D’Angelo\textsuperscript{1} Gabriele Di Stefano\textsuperscript{1}
Alfredo Navarra\textsuperscript{2}

\textsuperscript{1}Dept. of Electrical and Information Engineering, University of L’Aquila, Italy
\{gianlorenzo.dangelo,gabriele.distefano\}@univaq.it

\textsuperscript{2}Dept. of Mathematics and Informatics, University of Perugia, Italy
navarra@dipmat.unipg.it
Timetabling
Schedule the departure and arrival time of trains in order to reduce the traveling time for passengers

Delay Management
Modify the timetable when unpredictable events cause delays

Recoverable Robustness
Design the timetable in order to easily recover when delays occur [Liebchen et al. 2007, Cicerone et al. 2007]

This work
This work studies the Recoverable Robustness approach for timetabling in restricted topologies (Tree) and applies it to Italian single-line corridors
Previous Works

- Defined the recoverable robust timetabling ($RTT$) [Cicerone et al. 2008]
- $RTT$ is NP-hard [Cicerone et al. 2008]
- Linear time approximation algorithm [Cicerone et al. 2008]
- $RTT$ remains NP-hard when the topology is restricted to out-trees [D. et al. 2008]
- Pseudo-polynomial time optimal algorithm [D. et al. 2008]
Results of the paper

- We modelled single-line corridors as out-trees
- We implemented the algorithm in [D. 2008] which optimally solves $\mathcal{RTT}$ and applied it to Italian single-line corridors
- We experimentally showed that the algorithm is effective and efficient in practical cases
Outline

Recoverable Robust Timetabling problem

Data description

Algorithm

Experimental results

Conclusions
Outline

Recoverable Robust Timetabling problem

Data description

Algorithm

Experimental results

Conclusions
The timetabling problem [Schöbel, 2007]
Scheduling the departure and arrival time of trains

Instances

- A event activity network \( \mathcal{N} = (\mathcal{E}, \mathcal{A}) \) made of departure and arrival events \( \mathcal{E} \) and activities \( \mathcal{A} \)
- The minimum time \( L(a) \) needed for each activity \( a \in \mathcal{A} \)
- The number of passengers \( w(v) \) involved in each event \( v \in \mathcal{E} \)

Solutions
A scheduled time \( \Pi(v) \) for each event \( v \in \mathcal{E} \) in the network, such that \( \Pi \) satisfies the minimum duration time of each activity

Objective
Minimizing the overall traveling time for passengers
\[
\min f = \sum_{v \in \mathcal{E}} w(v) \Pi(v)
\]
Timetabling problem $TT$

$$\min f = \sum_{v \in \mathcal{E}} w(v) \Pi(v)$$

subject to

$$\Pi(v) - \Pi(u) \geq L(a), \quad \text{for each } a = (u, v) \in \mathcal{A}$$

$$\Pi(v) \in \mathbb{N}, \quad \text{for each } v \in \mathcal{E}$$

- $I$: set of instances $i = (\mathcal{N}, L, w)$
- $F(i)$: set of feasible solutions $\Pi$ for $i \in I$
Modification function $M$

We allow only one delay on an activity $a$ of at most $\alpha$ time.
We can model it as an increase of the minimal duration time $a$.

Given $i = (\mathcal{N}, L, w)$,

\[ M(i) = \{(\mathcal{N}, L', w) : L' \text{ differs from } L \text{ by at most one activity} \} \]

Recovery capabilities $\mathcal{A}$

A recovery algorithm can change the time of at most $\Delta$ affected events.
The Recoverable Robust Timetabling Problem $\mathcal{RTT}$

$\mathcal{RTT} = (TT, M, A)$ is the problem of finding a timetable that can be recovered by changing the time of at most $\Delta$ events when a delay of at most $\alpha$ time occurs.
Robust Algorithm

A robust algorithm for $TT$ is any algorithm $A_r$ which solves $RTT$.

Price of robustness

The worst case ratio between the cost of the solution computed by $A_r$ and the optimal one is called price of robustness of $A_r$.

$$P_{rob}(R TT, A_r) = \max_{i \in I} \left\{ \frac{f(A_r(i))}{\min\{f(x) : x \in F(i)\}} \right\}$$

The minimum price of robustness among all the robust algorithms is called price of robustness of problem
Outline

Recoverable Robust Timetabling problem

Data description

Algorithm

Experimental results

Conclusions
Single line corridors

- A corridor is a set of subsequent stations served by many trains of different type
- In practice (and intuitively) slow trains wait for faster trains to allow passengers to change from one train to another
- We require that changes between trains are only those connecting a fast train to the starting event of a slow train
Recoverable Robust Timetabling problem

Data description

Algorithm

Experimental results

Conclusions

Interregionale

Eurocity

Espresso

D'Angelo, Di Stefano, Navarra  Robust Timetables for Trains on Single-Line Corridors
 Outline

Recoverable Robust Timetabling problem

Data description

Algorithm

Experimental results

Conclusions
Slack times
According to a timetable $\Pi$, the slack time $s(a)$ of an activity $a = (u, v)$ is the amount of time assigned to an activity in addition to its minimum time needed $L(a)$

$$s(a) = \Pi(v) - \Pi(u) - L(a)$$

A robust timetable assigns at least slack time of $\alpha$ every $\Delta$ subsequent events

Idea
Assign the slack times as late as possible
Example for $\Delta = 2$ and $L(a) = 1$ for each $a \in \mathcal{A}$
Example for $\Delta = 2$ and $L(a) = 1$ for each $a \in \mathcal{A}$
Example for $\Delta = 2$ and $L(a) = 1$ for each $a \in \mathcal{A}$
Outline

Recoverable Robust Timetabling problem

Data description

Algorithm

Experimental results

Conclusions
<table>
<thead>
<tr>
<th>Corridor</th>
<th>Line</th>
<th>N. of Stations</th>
<th>N. of Trains</th>
</tr>
</thead>
<tbody>
<tr>
<td>BrBo</td>
<td>Brennero–Bologna</td>
<td>48</td>
<td>68</td>
</tr>
<tr>
<td>MdMi</td>
<td>Modane–Milano</td>
<td>54</td>
<td>291</td>
</tr>
<tr>
<td>BzVr</td>
<td>Bolzano–Verona</td>
<td>27</td>
<td>65</td>
</tr>
<tr>
<td>PzBo</td>
<td>Piacenza–Bologna</td>
<td>17</td>
<td>25</td>
</tr>
</tbody>
</table>

**Table:** Data used in the experiments.
<table>
<thead>
<tr>
<th>Corridor</th>
<th>N. of Nodes</th>
<th>Max Traveling Time</th>
<th>Avg Activity Time</th>
<th>Max N. of Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>BrBo</td>
<td>1103</td>
<td>516</td>
<td>9</td>
<td>66</td>
</tr>
<tr>
<td>MdMi</td>
<td>4358</td>
<td>318</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>BzVr</td>
<td>648</td>
<td>197</td>
<td>5</td>
<td>37</td>
</tr>
<tr>
<td>PzBo</td>
<td>163</td>
<td>187</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

**Table:** Sizes of the trees.

\[ \Delta \in \{1, 2, \ldots, 11\} \]

\[ \alpha \in \{1, 5, 9, 13, 17\} \]
BrBo Objective function

D’Angelo, Di Stefano, Navarra
Robust Timetables for Trains on Single-Line Corridors
BrBo Price of robustness

![Graph showing the price of robustness for different values of alpha (1, 5, 9). The x-axis represents Delta, and the y-axis represents the price of robustness.](image-url)
BrBo Computational time

Delta

alpha=1
alpha=5
alpha=9
MdMi Objective function

D’Angelo, Di Stefano, Navarra
Robust Timetables for Trains on Single-Line Corridors
MdMi Price of robustness

![Graph showing MdMi Price of robustness with different values of \( \alpha \).](image)

- \( \alpha = 1 \) (red line)
- \( \alpha = 5 \) (green line)
- \( \alpha = 9 \) (blue line)

Delta vs. Price of robustness.

---

D’Angelo, Di Stefano, Navarra

Robust Timetables for Trains on Single-Line Corridors
MdMi Computational time

alpha=1
alpha=5
alpha=9
BzVr Objective function

![Graph showing BzVr Objective function with different alpha values and Delta ranging from 0 to 12. The graph includes lines for alpha=1, alpha=5, alpha=9, and an Opt line, with values on the y-axis from 65000 to 30000.](image-url)
BzVr Price of robustness
PzBo Objective function

The graph shows the objective function for different values of $\alpha$. The x-axis represents the value of $\Delta$, and the y-axis represents the objective function values. There are lines for $\alpha = 1$, $\alpha = 5$, $\alpha = 9$, and $\text{Opt}$. The objective function values decrease as $\Delta$ increases for each case.
PzBo Price of robustness
PzBo Computational time

D’Angelo, Di Stefano, Navarra

Robust Timetables for Trains on Single-Line Corridors
We modelled the timetabling problem for single-line corridors as out-trees

We used the algorithm in [D. et al. 2008] in order to cope with one single delay

We experimentally shown the performances of the algorithm applied on real data

Although the problem is proved to be NP-hard, the obtained results show the applicability of the algorithm