Passenger Railway Optimization Problems

Paolo Toth
DEIS, Alma Mater Studiorum - University of Bologna
Outline

- Railway systems
- Train Platforming Problem
- Crew Planning Problem
Railway Systems

- Railway systems are highly complex.

- Planning and operational processes related to railway systems are rich in challenging Combinatorial Optimization problems.

- Railway transportation can be split into:
  - Passenger Transportation,
  - Cargo (Freight) Transportation.
Railway Systems

In many countries new regulations specify that the management of the railway infrastructure should be the responsibility of the governments, but operating trains should be carried out by independent companies on a commercial basis.

Separate organizations:

**Infrastructure Manager (IM)**, responsible for train planning and real-time traffic control,

**Train Operators (TOs)**, providing their preferred timetables, rolling stock and crew.
Collaborators

University of Bologna:
Alberto Caprara, Daniele Vigo,
Valentina Cacchiani, Laura Galli

University of Padova:
Matteo Fischetti, Michele Monaci

Rete Ferroviaria Italiana (main Italian IM):
Pier Luigi Guida
Decomposition

Due to the complexity of railway systems, the planning process is decomposed into sequential phases:

- **Line planning**: deciding the routes for the passenger trains, as well as the types and frequencies of the trains on each route.

- **Timetabling**: fixing the timetable for each train.

- **Platforming**: assigning trains to platforms in the stations they visit.

- **Rolling Stock Circulation**: defining train units (locomotives and train carriages) to be assigned to the trains, each having known timetable and platforms.

- **Crew Planning**: defining the workload of train drivers and conductors to operate a given timetable.
Surveys on Railway Optimization Problems


Figure 1: Main problems solved in the planning of a passenger railway system.
Line Planning

- Most passenger TOs operate a timetable, in which the scheduled trains can be partitioned into so called lines, containing trains with the same route and the same set of stop stations (but different arrival and departure times).

- **Line Planning Problem (LPP):** design a line system such that all travel demands are satisfied. Two main conflicting objectives:
  a) maximize the service towards the passengers;
  b) minimize the operational cost of the railway system (TO).
**Line Planning (2)**

**Direct passengers**: travel from origin to destination without changes.

Maximizing the number of direct passengers results in long lines:
- delays can be propagated in wider geographical areas,
- efficient allocation of rolling stocks may be prohibited.

LPP requires to define for each possible line (if used): frequency (number of trains per day) and capacity (overall number of seats).

**Objective**: minimize a weighted sum of connection costs for the passengers and operating costs for the TO by satisfying the demand of the passengers.

Train Timetabling

- **Train Timetabling Problem (TTP):** provide a *timetable* for a set of trains (many TOs) on a certain part of the railway network (single IM).

- Each TO submits the IM a set of requests of train *paths* in the railway network, each with a *profit* that the TO is willing to pay for it, an *ideal timetable* with desired departure / arrival times for each station in which the path has to stop, and penalties for possible deviations with respect to the ideal timetable (required for satisfying operational constraints imposing a minimum headway between trains traveling on the same track).
Train Timetabling (2)

For each path: determine whether to cancel it or to schedule it (with a corresponding actual timetable).

Objective: maximize the difference between the global profit of the paths scheduled and the global penalties for the deviations of the actual timetables from the ideal ones.

Rolling Stock Planning

The *rolling stock* to be assigned to the trains can be:
- *locomotives* and *train carriages*,
- *aggregated modules (train units)* composed of carriages in a fixed composition.

A train can be composed of several coupled train units. To obtain a better match between the available rolling stock and the passengers’ seat demand, the composition of the trains can be changed at several stations by adding or removing a train unit.

A *trip* is a part of a train timetable that must be performed by the same train unit without changes.

The *Rolling Stock Planning Problem* (RSPP) calls for assigning train units to trips.
Rolling Stock Planning

- Given the train unit availability, RSPP requires the minimization of a weighted sum of the costs of the total distance traveled by the train units, of the composition change costs, and of the seat shortages with respect to the passengers’ requests.

- Operational constraints impose a maximum length for each train, that composition changes are done by respecting the timetable, and that the maintenance is carried out properly.

Train Platforming Problem

Input:

Given a set $T$ of trains to be run every day of a given time horizon, for each train $t \in T$:

- **Train Schedule**: arrival and departure “ideal” times $AT(t)$ and $DT(t)$, directions $AD(t)$ and $DD(t)$, maximum shifts $AS(t)$ and $DS(t)$, weight (priority) $W(t)$.

The “actual” arrival time of train $t$ at a platform in

$$[AT(t) - AS(t), AT(t) + AS(t)]$$

...  

All the times are expressed in minutes (modulo 1440 = number of minutes in a day)
Input:

Railway Station Topology: platforms, directions and paths:

- $B =$ set of platforms,
- $D =$ set of arrival and departure directions.

For each pair:
- $(\text{direction } d, \text{platform } b)$ or $(\text{platform } b, \text{direction } d)$,

a (possibly empty) set of “paths” is defined ($R =$ global set of paths).

Two paths are “incompatible” if they intersect (i.e. they have common “junctions”).

Incompatible paths should not be assigned to “overlapping” trains.
Output

Assign each train $t \in T$:
- a platform $b \in B$,
- an arrival path,
- a departure path,
- possible arrival and departure shifts,

s.t. no operational constraint is violated.

TPP can be proved to be Strongly NP-Hard
(a special case coincides with the “Circular Arc Graph Coloring Problem”)

Different versions of the Train Platforming Problem (TPP) have been proposed in the literature.

The considered problems are easy to be solved for small contexts (stations with few platforms and alternative paths).

Extremely difficult when applied to complex railway station topologies (instances with hundreds of trains, tens of platforms, thousands of path incompatibilities).

Most versions are not concerned with the station topology, and ignore the routing phase (assignment of paths).

Main stations frequently have complex topologies and the routing issue can be quite a complicated task.

A general formulation of TPP, a MILP model and a solution approach will be proposed.
Train Platforming Problem: References


Kroon L.G., Romeijn H.E., Zwaneveld P.J.
Routing Trains through Railway Stations: Complexity Issues - EJOR 98, 1997

Zwaneveld P.J.
Railway Planning and Allocation of Passenger Lines - Ph.D. Series in General Management 25, 1997
(Rotterdam School of Management)

De Luca Cardillo D., Mione N.
k L-list tau coloring of graphs - EJOR 106, 1999

Zwaneveld P.J., Kroon L.G. van Hoesel C.P.M
Routing Trains through a Railway Station based on a Node Packing Model - EJOR 128, 2001

Billionnet A.
Using Integer Programming to Solve the Train-Platforming - Transportation Science 37, 2003

Carey M., Carville S.
Scheduling and platforming trains at busy complex stations - Transportation Research 37, 2003

Caprara A., Galli L., Monaci M., T. P.

Caprara A., Galli L., T. P.
Solution of the Train Platforming Problem – EU Project ARRIVAL, 2009 (under revision).
Goals

- Assign each $\text{train } t \in T$ a $\text{platform } b \in B$
  
  **Platform occupation**: at any instant each platform can be assigned to a single train ("hard" constraints).

- Assign each $\text{train } t \in T$ an arrival and a departure path:
  
  **Path occupation**: incompatible paths can be assigned to different trains at overlapping time intervals if the overlapping is smaller than a given threshold $\pi$ ("soft" constraints: a penalty must be paid in case of path overlapping).

- Assign each $\text{train } t \in T$ an arrival and a departure shift
  (a penalty must be paid in case of shift).
Railway Station Topology

1. MILANO (1)
2. PADOVA (2)
3. FIRENZE (3)
4. DEPOT (4)
Platform assignment \((T = \{A, B, C, D, E, F, G\})\)

Diagram showing platform assignments and conflicts.
Path assignment
(trains A and D assigned to Platform 1)
Preference list

- Each train $t \in T$ has a “preference list” $PL(t)$, containing its preference platforms.

- If a train cannot be assigned to a preference platform, an out-list platform can be used (a penalty must be paid in this case)
Platforms assignment

- If all platforms (paths) are busy, in order to find a feasible solution, dummy platforms can be used (a large penalty must be paid); DB = set of dummy platforms.

- The “virtual” paths connecting the directions with the dummy platforms (and vice versa) are compatible with all the other paths.

- The only incompatibilities between trains assigned to the same dummy platform b are related to the occupation of b.
TPP can be solved for both long term and medium term planning. TPP can:

- produce a feasible platforming plan according to a given timetable, or
- measure the lack of capacity of the considered station in terms of platforms and paths.
- The objective function contains many terms, but the primary objective is to use as few dummy platforms as possible.
- The concept of *dummy platform* is introduced to measure the lack of capacity of the station, in case not all the trains can be assigned to *regular platforms* according to the given timetable.
- **Long term planning**: enlarge the station if dummy platforms are used,
- **Medium term planning**: not all the trains can be scheduled (unless the safety margins are relaxed)
There are two types of costs:

1) Costs associated with the used platforms.

2) Penalties associated with platforming “quality”.

Minimize

\[ k_1 \times B_1 + k_2 \times B_2 + k_3 \times P_3 + k_4 \times P_4 + k_5 \times P_5 + k_6 \times P_6 \]

**Objective function**

**Train Platforming Problem**

- \( B_1 \) = number of used platforms
  - \( k_1 = 1000 \)

- \( B_2 \) = number of dummy platforms used
  - \( k_2 \) very large: \( k_1 = 100000 \)

- \( P_3 \) = global “weighted” shift
  - (1)

- \( P_4 \) = global weight of the trains assigned to non preference platforms
  - (100)

- \( P_5 \) = global weight of the trains assigned to dummy platforms
  - (10000)

- \( P_6 \) = global number of dynamic conflict minutes (sum of the time intervals corresponding to path conflicts)
  - (5)
Patterns

- Given a train $t \in T$, a “pattern” $p$ associated with $t$ is a “quintuple”: $p = (b, ap, dp, as, ds)$

- where:
  $b = \text{platform assigned to } t$,
  $ap, dp = \text{arrival and departure paths assigned to } t$,
  $as, ds = \text{arrival and departure shifts assigned to } t$.

$P(t) = \text{set of patterns } p \text{ associated with train } t$

(for the applications we are aware of, the overall number of patterns allows one to handle explicitly all of them)
Pattern-Incompatibility Graph $G$

Operational constraints can be expressed using a pattern-incompatibility graph $G$:
- a node for each (train - pattern) pair $(t, p)$, with $t \in T$ and $p \in P(t)$;
- an edge joining each pair $(t_1, p_1), (t_2, p_2)$ of incompatible patterns.

Let $K$ define the whole collection of cliques in graph $G$.

Let $K(b)$, with $b \in B$, define the collection of cliques in $G$ associated with sets of patterns using platform $b$ at the same time:
  for each pair $(t_1, p_1), (t_2, p_2) \in K(b)$, pattern $p_1$ for train $t_1$ and pattern $p_2$ for train $t_2$ both use platform $b$, occupying it for overlapping intervals.
Variables

\[ y_b = \begin{cases} 
1 & \text{if platform } b \text{ is used} \\
0 & \text{otherwise} 
\end{cases} \quad b \in B \]

\[ x_{t,p} = \begin{cases} 
1 & \text{if train } t \text{ uses pattern } p \\
0 & \text{otherwise} 
\end{cases} \quad t \in T \quad p \in P(t) \]

\[ \]

B : platform set

T : train set

P(t) : set of patterns associated with train t
Costs definition

- Cost of platform \( b \in B \): \( C_b = k_1 + k_2 \) (if \( b \in DB \))

- Cost of pattern \( p \in P(t) \), with \( t \in T \), \( p = (b, ap, dp, as, ds) \):
  \[
  c_{t,p} = W(t) * (k_3 * (as + ds) + k_4 \text{ (if } b \in B \setminus PL(t)) + k_5 \text{ (if } b \in DB))
  \]

- Cost associated with the assignments of \( p_1 \in P(t_1) \) to train \( t_1 \) and of \( p_2 \in P(t_2) \) to train \( t_2 \) (with \( (t_1, t_2) \in T^2 \)), in case \( p_1 \) and \( p_2 \) have a soft incompatibility:
  \[
  c_{t_1,p_1,t_2,p_2} = k_6 * \text{ (conflict minutes between } p_1 \text{ and } p_2)
  \]

\( T^2 \subseteq \{ (t_1, t_2) : t_1, t_2 \in T, t_1 \neq t_2 \} = \text{ set of pairs of distinct trains whose patterns may have a “hard” or “soft” incompatibility} \)
**INTEGER QUADRATIC PROGRAMMING FORMULATION**

\[
\begin{align*}
\text{min } & \sum_{b \in B} c_b y_b + \sum_{t \in T} \sum_{p \in P(t)} c_{t,p} x_{t,p} + \sum_{(t_1,t_2) \in T^2} \sum_{p_1 \in P(t_1)} \sum_{p_2 \in P(t_2)} c_{t_1,p_1,t_2,p_2} x_{t_1,p_1} x_{t_2,p_2} \\
\text{s.t. } & \sum_{p \in P(t)} x_{t,p} = 1 & t \in T \\
& \sum_{(t,p) \in k} x_{t,p} \leq y_b & k \in K(b), b \in B \\
& \sum_{(t,p) \in k} x_{t,p} \leq 1 & k \in K \\
& y_b \in \{0, 1\} & b \in B, \quad x_{t,p} \in \{0, 1\} & t \in T, p \in P(t)
\end{align*}
\]

Constraints (2) guarantee that each train \( t \) is assigned a pattern.
Constraints (3) impose that at most one train at a time occupies a platform \( b \) (if \( b \) is used).
Constraints (4) forbid the assignment of patterns that are pairwise incompatible.
The number of both constraints (3) and (4) is exponential in the number of patterns.
Improved Integer Programming model

Constraints (3): Clique Constraints for Platforms

\[ \sum_{(t,p) \in k} x_{t,p} \leq y_b \quad k \in K(b), \ b \in B \] (3)

\( K(b) = \) collection of cliques in \( G \) associated with sets of patterns using platform \( b \) at the same time

Each clique \( k \) in \( K(b) \) corresponds to a set of intervals (associated with the occupation of platform \( b \) of each pattern of \( k \)) that intersect pairwise.

From the basic theory of Interval graphs (Fishburn, J. Wiley, 1985): each maximal clique is defined by an interval starting at point \( j \) together with the intervals \([h,l]\) with \( h \leq j \) and \( l > j \) : the number of maximal cliques cannot be larger than the number of intervals.

\( J(b) \) be the set of instants associated with the beginning of the occupation of platform \( b \) by a pattern.

\( K(b,j) \subseteq K \) be the set of patterns that occupy platform \( b \) for the interval \([h,l]\) with \( h \leq j \) and \( l > j \).
Improved Integer Programming model

Constraints (3): Clique Constraints for Platforms

\[ \sum_{(t,p) \in k} x_{t,p} \leq y_b \quad k \in K(b), \ b \in B \]  

(3)

Constraints (3) can be replaced by constraints:

\[ \sum_{(t,p) \in K(b,j)} x_{t,p} \leq y_b \quad b \in B, \ j \in J(b) \]  

(6)

A train \( t \) can begin to occupy a platform for at most \( (AS(t) + DS(t) + 1) \) instants:

A global number of constraints (6):

\[ \sum_{b \in B} |J(b)| \leq |B| \sum_{t \in T} (AS(t) + DS(t) + 1) \]
Constraints (4): Clique Constraints for Paths
(in general hard to separate)

\[ \sum_{(t,p) \in k} x_{t,p} \leq 1 \quad k \in K \] (4)

Restrict attention to cliques in \( K \) containing patterns of two trains only:
family of relaxed constraints that still define a valid formulation for TPP,
and are strong enough to be used in practice.

Given two trains \( t1 \) and \( t2 \), let \( K(t1,t2) \subseteq K \) denote the collection of cliques
containing only incompatible patterns in \( P(t1) \cup P(t2) \):

alternative version of constraints (4)

\[ \sum_{(t1,p1) \in k} x_{t1,p1} + \sum_{(t2,p2) \in k} x_{t2,p2} \leq 1 \quad (t1,t2) \in T^2, \ k \in K(t1,t2) \] (7)
Separation of Constraints (7): Clique Constraints for Paths

\[ \sum_{(t1,p1) \in k} x_{t1,p1} + \sum_{(t2,p2) \in k} x_{t2,p2} \leq 1 \quad (t1,t2) \in T^2, \ k \in K(t1,t2) \quad (7) \]

It can be shown that separation of constraints (7) calls for the separation of cliques inequalities in the complement of a bipartite graph (bipartition corresponding to patterns in \( P(t1) \) and \( P(t2) \), respectively):

- separation of stable set inequalities in a bipartite graph:
- determination of a maximum-weight stable set of a bipartite graph:
- minimum s, t-cut problem on a directed network with source s, terminal t, and the other nodes corresponding to the nodes in the bipartite graph (maximum-flow code):
- computation of \(|T^2|\) maximum flows in a network with \( O(P_{max}) \) nodes, with

\[ P_{max} := \max_{t \in T} |P(t)| \]
Linearizing the Quadratic Term of the Objective Function

\[ \sum_{(t_1,t_2) \in T^2} \sum_{p_1 \in P(t_1)} \sum_{p_2 \in P(t_2)} c_{t_1,p_1,t_2,p_2} x_{t_1,p_1} x_{t_2,p_2} \]

**Standard linearization** amounts to introducing **binary variables**

\[ z_{t_1,p_1,t_2,p_2} \]

forced to take value 1 iff

\[ x_{t_1,p_1} = x_{t_2,p_2} = 1 \]

by imposing the **linear constraints**

\[ z_{t_1,p_1,t_2,p_2} \geq x_{t_1,p_1} + x_{t_2,p_2} - 1 \]

**For TPP:** - too many \( z \) variables

- weak LP Relaxation
Linearizing the Quadratic Term of the Objective Function (2)

\[ \sum_{(t1,t2) \in T^2} \sum_{p1 \in P(t1)} \sum_{p2 \in P(t2)} c_{t1,p1,t2,p2} x_{t1,p1} x_{t2,p2} \]

Introduce an additional continuous variable \( w_{t1,t2} \) for each \( (t1,t2) \in T^2 \):

\[ w_{t1,t2} = \sum_{p1 \in P(t1)} \sum_{p2 \in P(t2)} c_{t1,p1,t2,p2} x_{t1,p1} x_{t2,p2} \quad (t1,t2) \in T^2 \]

**Linear Objective Function:**

\[
\min \sum_{b \in B} c_b y_b + \sum_{t \in T} \sum_{p \in P(t)} c_{t,p} x_{t,p} + \sum_{(t1,t2) \in T^2} w_{t1,t2} \quad (8)
\]
Linearizing the Quadratic Term of the Objective Function (3)

\[ \sum \sum \sum c_{t1,p1,t2,p2} x_{t1,p1} x_{t2,p2} \]

\[ (t1,t2) \in T^2 \quad p1 \in P(t1) \quad p2 \in P(t2) \]

Elementary links between the \( x \) and \( w \) variables:

\[ w_{t1,t2} \geq c_{t1,p1,t2,p2} (x_{t1,p1} + x_{t2,p2} - 1) \]

\[ (t1,t2) \in T^2 \quad p1 \in P(t1) \quad p2 \in P(t2) \]

lead to a weak MILP model equivalent to the standard one.

Stronger inequalities can be introduced.
Separation of the Stronger Constraints

\[ w_{t_1,t_2} \geq \sum_{p_1 \in P(t_1)} \alpha_{p_1} x_{t_1,p_1} + \sum_{p_2 \in P(t_2)} \beta_{p_2} x_{t_2,p_2} - \gamma \quad (t_1,t_2) \in T^2, \ (\alpha,\beta,\gamma) \in F(t_1,t_2) \quad (9) \]

It can be shown that separation of constraints (9) can be performed through the computation of $|T^2|$ LPs with $O(P_{max})$ variables and $O(P_{max}^2)$ constraints

with $P_{max} := \max_{t \in T} |P(t)|$
OVERALL MILP FORMULATION

$$\min \sum_{b \in B} c_b y_b + \sum_{t \in T} \sum_{p \in P(t)} c_{t,p} x_{t,p} + \sum_{(t1,t2) \in T^2} w_{t1,t2}$$ (8)

s.t.

$$\sum_{p \in P(t)} x_{t,p} = 1 \quad t \in T$$ (2)

$$\sum_{(t,p) \in k(b,j)} x_{t,p} \leq y_b \quad b \in B, j \in J(b)$$ (6)

$$\sum_{(t1,p1) \in k} x_{t1,p1} + \sum_{(t2,p2) \in k} x_{t2,p2} \leq 1 \quad (t1,t2) \in T^2, k \in K(t1,t2)$$ (7)

$$w_{t1,t2} \geq \sum_{p1 \in P(t1)} \alpha_{p1} x_{t1,p1} + \sum_{p2 \in P(t2)} \beta_{p2} x_{t2,p2} - \gamma \quad (t1,t2) \in T^2, (\alpha,\beta,\gamma) \in F(t1,t2)$$ (9)

$$y_b \in \{0, 1\} \quad b \in B, \quad x_{t,p} \in \{0, 1\} \quad t \in T, p \in P(t)$$ (5)
Solution methodology

- The algorithm implemented is a Branch-and-Bound procedure based on the continuous relaxation of the MILP model.
Solution methodology

- ...but the model has a huge number of variables and constraints
- It is impossible to handle them directly using a general purpose solver for LP models, so...

Column generation

Separation
A “branch-and-cut-and-price” approach

- rows…

| constraints (2) |
| constraints (6) |
| constraints (7) |
| constraints (9) |

- …and columns

\[ y_b \quad x_{t,p} \quad w_{t1,t2} \]
Computational results

- C language
- CPLEX 10 (as LP solver)
- PC Pentium 4, 3.2 GHz
- 2 GB RAM
- OS: Windows XP Pro
Computational results

Real-World instances from:
Rete Ferroviaria Italiana (RFI),
main Italian Railway Infrastructure Manager

| instance  | station name       | $|T|$ | $|B|$ | $|D|$ | $|\mathcal{R}|$ | # inc. | $g_{d}^{\text{max}}$ |
|-----------|--------------------|-----|-----|-----|---------|--------|------------------|
| PA C.LE.  | Palermo Centrale   | 204 | 11  | 4   | 64      | 1182   | 3                |
| GE P.PR.  | Genova Piazza Principe | 127 | 10  | 4   | 174     | 7154   | 4                |
| BA C.LE.  | Bari Centrale      | 237 | 14  | 5   | 89      | 1996   | 4                |

# inc = number of pairs of incompatible paths
$g_{d}^{\text{max}}$ = maximum travel time (occupation time) of paths

- Time limit for each instance = 4 hours
- Times expressed in seconds

Comparisons with the results obtained by the Heuristic Algorithm ("curr") currently used by RFI
The main goal of the experiment is to evaluate the performance of the current station topology (platforms and paths).

Extend the current “capacity” of the stations considered, by using the minimum number of platforms for the existing trains and then allowing new trains to stop at the station.

In case of congested scenarios, the goal is to find feasible solutions, or to reduce the global infeasibility (number of dummy platforms used and of trains assigned to dummy platforms).
### Computational results

<table>
<thead>
<tr>
<th>instance</th>
<th>$\pi$</th>
<th>curr</th>
<th>LP value</th>
<th>time</th>
<th>first value</th>
<th>time</th>
<th>best value</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA C.LE.</td>
<td>0</td>
<td>749012</td>
<td>234038</td>
<td>27</td>
<td>349037</td>
<td>48</td>
<td>339039</td>
<td>56</td>
</tr>
<tr>
<td>PA C.LE.</td>
<td>1</td>
<td>410139</td>
<td>10159</td>
<td>56</td>
<td>120155</td>
<td>96</td>
<td>120155</td>
<td>96</td>
</tr>
<tr>
<td>PA C.LE.</td>
<td>2</td>
<td>380182</td>
<td>10159</td>
<td>53</td>
<td>10176</td>
<td>86</td>
<td>10176</td>
<td>86</td>
</tr>
<tr>
<td>PA C.LE.</td>
<td>3</td>
<td>11431</td>
<td>10159</td>
<td>54</td>
<td>10172</td>
<td>106</td>
<td>10172</td>
<td>106</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>0</td>
<td>745000</td>
<td>306020</td>
<td>67</td>
<td>306020*</td>
<td>115</td>
<td>306020*</td>
<td>115</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>1</td>
<td>705005</td>
<td>147069</td>
<td>192</td>
<td>147087</td>
<td>490</td>
<td>147080</td>
<td>825</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>2</td>
<td>458065</td>
<td>8116</td>
<td>274</td>
<td>8121</td>
<td>499</td>
<td>8116*</td>
<td>4617</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>3</td>
<td>336340</td>
<td>8116</td>
<td>572</td>
<td>8121</td>
<td>1057</td>
<td>8116*</td>
<td>13647</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>4</td>
<td>8692</td>
<td>8116</td>
<td>434</td>
<td>8126</td>
<td>866</td>
<td>8116*</td>
<td>1040</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>0</td>
<td>1576300</td>
<td>653204</td>
<td>122</td>
<td>808255</td>
<td>350</td>
<td>808255</td>
<td>350</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>1</td>
<td>1398330</td>
<td>373486</td>
<td>123</td>
<td>438685</td>
<td>642</td>
<td>438685</td>
<td>642</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>2</td>
<td>1197485</td>
<td>128896</td>
<td>199</td>
<td>148867</td>
<td>542</td>
<td>148867</td>
<td>542</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>3</td>
<td>838235</td>
<td>8885</td>
<td>216</td>
<td>8924</td>
<td>656</td>
<td>8924</td>
<td>656</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>4</td>
<td>11290</td>
<td>8877</td>
<td>198</td>
<td>8912</td>
<td>880</td>
<td>8912</td>
<td>880</td>
</tr>
</tbody>
</table>

**Different values of the dynamic threshold $\pi$ (max allowed conflict time):**

- $\pi = 0$: simultaneous occupation of incompatible paths is forbidden
- $\pi = g_d^{\text{max}}$: simultaneous occupation of incompatible paths does not affect feasibility, but affects only the quadratic part of the objective function
### Minimization of the number of dummy platforms

<table>
<thead>
<tr>
<th>instance</th>
<th>$\pi$</th>
<th>curr</th>
<th>LP</th>
<th>first</th>
<th>best</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>value</td>
<td>time</td>
<td>value</td>
</tr>
<tr>
<td>PA C.LE.</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>26</td>
<td>2*</td>
</tr>
<tr>
<td>PA C.LE.</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>PA C.LE.</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>10</td>
<td>0*</td>
</tr>
<tr>
<td>PA C.LE.</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0*</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>39</td>
<td>2*</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>97</td>
<td>1*</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>70</td>
<td>0*</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>22</td>
<td>0*</td>
</tr>
<tr>
<td>GE P.PR.</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0*</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>214</td>
<td>3</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>48</td>
<td>2*</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>39</td>
<td>2*</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>60</td>
<td>0*</td>
</tr>
<tr>
<td>BA C.LE.</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0*</td>
</tr>
</tbody>
</table>
often the capacities of the stations are bottlenecks of the railway system: even the small, daily, and unavoidable delays of some trains can spread heavily onto the whole system:

- the goal is a platforming plan yielding a high throughput of trains in the station while limiting the spread-on delays

- **Classical Robust Optimization (RO)** is an approach to optimization under uncertainty.
  
  Ben Tal and Nemirovski, *Mathematical Programming*, 2002,
  Bertsimas and Sim, *Operations Research*, 2004,
RO keeps the original (nominal) objective function and guarantees that the solution is feasible not only for the nominal scenario but also for all the possible scenarios, by listing these scenarios as a mathematical program, through uncertainties in the data defining the problem constraints.

RO has the clear disadvantage to be over conservative, since it focuses on finding a solution that is feasible for all scenarios and may be of very bad quality.

The main drawback of RO is that it does not consider the possibility of changing the nominal solution within the operations to adapt it to the scenario that is occurring in practice.
The Recoverable Robustness approach

(Liebchen, Luebbecke, Moehring, Stiller, "Robust and On Line Optimization", Ahuja, Moehring, Zarolagis, eds, Springer Verlag, 2009)

combines the notion of recoverable algorithm, that is used to adapt the nominal solution to the actual scenario, and the implicit representation of the list of scenarios as a mathematical program, with no assumption on the probabilities associated with these scenarios.

Classical Robust Solutions for platforming need to have excessively large time buffers between each pair of trains using a common resource (a platform or a path).

In a Recoverable Robust Platforming the buffers are cautiously distributed in the system to ensure that the total delay stays below a certain threshold in every likely scenario.
 Recoverable Robustness is very well suited for platforming.

It is possible to define a tractable model, which can be tackled by existing solving techniques for the original optimization problem, obtained from the nominal (deterministic) model:

1) by adding a variable $D$ (global delay) to the nominal objective function (to be minimized),
2) by imposing additional linear constraints.

Approach proposed in Caprara, Galli, Stiller, T., EU Project ARRIVAL, 2009.
Computational Results

- Real world instances Palermo Centrale and Genova Porta Principe from Rete Ferroviaria Italiana.

- Seven **time windows** in a day:
  - **nom** refers to solutions optimized for the deterministic TPP
  - **RR** refers to recoverable robust solutions,
  - **D** is the maximum propagated delay in minutes over all the scenarios with at most 30 minutes of seminal disturbances.

For each time window the recoverable robust solutions are able to assign the **same number of trains** as the nominally optimal solutions.
### Results for Palermo Centrale

<table>
<thead>
<tr>
<th>time window</th>
<th># trains not platformed</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: 00:00-07:30</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: 07:30-09:00</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: 09:00-11:00</td>
<td>0</td>
<td>487</td>
<td>6</td>
<td>355</td>
<td>143</td>
<td>131</td>
<td>26.90</td>
</tr>
<tr>
<td>D: 11:00-13:30</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: 13:30-15:30</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: 15:30-18:00</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: 18:00-00:00</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>
## Results for Genova Porta Principe

<table>
<thead>
<tr>
<th>time window</th>
<th># trains not platformed</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: 00:00-06:00</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: 06:00-07:00</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: 07:00-08:00</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: 08:00-09:00</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: 09:00-10:00</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: 10:00-11:00</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: 11:00-12:00</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>
RAILWAY CREW PLANNING

- We are given a planned timetable for the train services (actual journeys with passengers or freight, and the transfers of empty trains or equipment between different stations) to be performed every day of a certain time period.

- Each train service is split into a sequence of trips (duty elements, train tasks): segments of train journeys which must be served ("covered") by the same crew (driver, conductor) without interruption.

- Each trip is characterized by:
  - departure time, departure station,
  - arrival time, arrival station,
  - additional attributes.

- Each daily occurrence of a trip has to be covered by a crew.
RAILWAY CREW PLANNING (2)

- Each crew (drivers, conductors) performs a roster: sequence of trips whose operational cost and feasibility depend on several rules laid down by union contracts and company regulations (cyclic for “long” time periods).

- The problem consists of finding a set of rosters, covering every daily occurrence of each trip in the given time period, so as to satisfy all the operational constraints with minimum cost (minimum number of crews).

- Very complex and challenging problem due to both the size of the instances and the type and number of operational constraints.

- In the Italian Train Operator Company (“Trenitalia - Ferrovie dello Stato FS”): about 8000 trains and 25000 drivers (largest problem involves about 5000 trips).
Papers on crew planning

- Caprara, Monaci, T., CASPT 2002.
- Ens, Jiang, Krishnamoorthy, Owens, Sier, Annals of O.R., 2004
The overall problem is approached in two phases: (current practice adopted by most TOs)

1. **CREW SCHEDULING:**
   - the short-term schedule of the crews is considered, and a convenient set of pairings “covering” all the trips is constructed.
   - Each pairing (duty) represents a sequence of trips to be covered by a single crew within a given time period overlapping at most $L$ consecutive days (i.e. 2 days or 24 hours).

2. **CREW ROSTERING:**
   - The pairings selected in phase 1 are sequenced to obtain the final rosters.
   - Trips are no longer taken into account in an explicit way, but determine the attributes of the pairings which are relevant for the roster feasibility and cost.
• The **Crew Rostering Phase** considers each depot separately, since a roster cannot include pairings associated with different crew home depots.

• **Main objective:** minimization of the global number of crews needed to cover all the daily occurrences of the trips in the given period (i.e. the global “length” of the rosters).

• In **urban mass-transit** applications the crew rostering phase plays a minor role, since the corresponding constraints are rather weak and the number of crews is easily determined from the solution of the crew scheduling phase:
  - crew rostering is aimed at **balancing the workload** among the crews as evenly as possible;
  - the objective of the crew scheduling phase calls for the minimization of the number of working days (required crews).
• In **railway applications** considerable savings can be obtained through a clever sequencing of the pairings obtained in the crew scheduling phase.

• The **objective of the crew scheduling phase** has to take into account the **characteristics of the pairings** selected, and their implication in the subsequent rostering phase.

• **Feedback** between the two phases (dynamic updating of the crew scheduling costs).
INTEGRATION OF PAIRING AND ROSTERING OPTIMIZATION

• The SCHEDULING OPTIMIZATION (SO) and the ROSTERING OPTIMIZATION (RO) phases can be joined together in an iterative way to obtain a better overall solution.

• Both the SO and the RO phases are kept, but:
  - the selection of the pairings in SO is driven by the objective function of RO;
  - each time a new candidate set of pairings is found in SO, RO is executed to check if the overall incumbent solution can be updated.
EXPERIMENTAL RESULTS

- Standard ("Old") and Integrated ("New") Crew Planning Systems compared on a set of real-world instances provided by Trenitalia (Ferrovie dello Stato).

- PENTIUM 3 GHz.

- "OLD" SYSTEM
  - 1500 seconds for the PO phase
  - 300 seconds for the RO phase

- "NEW" SYSTEM
  - overall time limit = 1800 seconds
## CHARACTERISTICS OF THE INSTANCES

<table>
<thead>
<tr>
<th>Instance</th>
<th># trips</th>
<th># depots</th>
<th># pairings</th>
<th>time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESTRE</td>
<td>121</td>
<td>1</td>
<td>1,024,448</td>
<td>28</td>
</tr>
<tr>
<td>MILAN</td>
<td>502</td>
<td>1</td>
<td>874.416</td>
<td>105</td>
</tr>
<tr>
<td>VERONA</td>
<td>86</td>
<td>1</td>
<td>484.139</td>
<td>16</td>
</tr>
<tr>
<td>BZ-TS-UD</td>
<td>118</td>
<td>3</td>
<td>457.021</td>
<td>18</td>
</tr>
<tr>
<td>A-CH-F-D-I-1</td>
<td>91</td>
<td>13</td>
<td>796.771</td>
<td>23</td>
</tr>
<tr>
<td>A-CH-F-D-I-2</td>
<td>309</td>
<td>15</td>
<td>497.847</td>
<td>62</td>
</tr>
<tr>
<td>Instance</td>
<td>OLD SYSTEM</td>
<td></td>
<td></td>
<td>NEW SYSTEM</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td># weeks</td>
<td>LB</td>
<td># pair.</td>
<td>time</td>
</tr>
<tr>
<td>MESTRE</td>
<td>11</td>
<td>11</td>
<td>37</td>
<td>198</td>
</tr>
<tr>
<td>MILAN</td>
<td>50</td>
<td>50</td>
<td>145</td>
<td>1014</td>
</tr>
<tr>
<td>VERONA</td>
<td>8</td>
<td>8</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>BZ-TS-UD (3 depots)</td>
<td>11</td>
<td>11</td>
<td>30</td>
<td>17</td>
</tr>
<tr>
<td>A-CH-F-D-I-1 (13 depots)</td>
<td>12</td>
<td>12</td>
<td>28</td>
<td>138</td>
</tr>
<tr>
<td>A-CH-F-D-I-2 (15 depots)</td>
<td>49</td>
<td>49</td>
<td>127</td>
<td>201</td>
</tr>
<tr>
<td>total</td>
<td>141</td>
<td>1583 secs</td>
<td>128</td>
<td>2079 secs</td>
</tr>
</tbody>
</table>