Improving Real-Time Train Dispatching: Models, Algorithms and Applications*

Dr. Andrea D’Ariano

*TRAIL PhD Thesis downloadable from http://www.darenet.nl
PhD research project

**Aim:** Development of novel traffic management systems for a precise and effective railway traffic regulation in terms of:
1. Punctuality increase
2. Energy saving

**Tool:** Flexible rail operations, models and algorithms for train sequencing, routing and speed coordination

**Application:** Recover real-time timetable disturbances such as multiple delays and blocked tracks
Other parties (un)directly involved in the project

Delft University of Technology:

TRAIL Research School:

Dutch Infrastructure Manager ProRail:

Dutch Foundation “Next Generation Infrastructures”:

Università degli Studi di Siena:

Dresden University of Technology:
Presentation outline

- Problem introduction
- Dispatching support system
- Railway traffic optimization
- Assessment of flexible operations
- Evaluation of green wave policy
- Summary and coming research
Railway traffic management

Scientific issues:
• Reliability and punctuality of railway services
• Effectiveness of traffic control in case of disturbances

Means:
• Timetabling (off-line)
  ➢ Design of feasible schedules
  ➢ Robustness against perturbations
• Traffic monitoring
• Train dispatching (on-line)
  ➢ Manual
  ➢ Automatic
Railway traffic regulation

Automatic:
- According to scheduled arrival/departure times and itineraries (no delay, no problems, no action)
- Route setting at junctions (Dutch: ARI)

Dispatching techniques:
- Station dwell and running time extensions (*re-timing*)
- Holding/overtaking trains (*re-ordering*)
- Alternative routes (*re-routing*)
- Cancellation of routes and/or connected services
Inter-area traffic management and Francesco Corman’s research...
Local dispatching practice

Only in case of disturbances...

Three basic rules (ARI):

- **If** there is a conflict between trains running to the same track: The planned order is maintained;
- **If** there is a conflict between trains running to different tracks: The train that has claimed its route first, will go first (FCFS);
- **When** trains are outside a predefined time-window (usually 3 or 5 minutes) the dispatcher may act according to his knowledge, experience and a list of what-if scenarios.
Research motivation

Actual limits of the dispatching practice:

- Computer support often limited to graphical interfaces and automatic route setting systems (e.g. ARI);

- Dispatchers usually do not have precise information of the future evolution of train traffic and the chosen actions may be sub-optimal [Kauppi et al. CTW 2006];

- The delay propagation is unpredictable by traffic controllers, specially in case of complex rail networks, high density traffic, severe disturbances.
No advanced dispatching support tool exists to reschedule train movements during operations.

In fact, there is still a lack of:

- **Precision**: Models and algorithms must include the variability of train dynamics and must respect specific railway constraints
- **Robustness**: Existing dispatching systems are able to provide viable solutions only for small instances and simple disturbances
- **Efficiency**: Near optimal dispatching solutions can be computed only if global information is considered when computing orders, routes and times of the trains running in the studied area
- **Quickness**: The development of novel optimization algorithms must include the constraints due to limited computation times
State-of-the-art: Proposed improvements

Our advanced dispatching support tool is designed in order to satisfy the necessary operational requirements as follows:

- **Precision**: We formulate the problem and relevant constraints as a combination of alternative graphs and blocking time theory.

- **Robustness**: We compute feasible solutions even for large and complex instances, and various types of disturbances.

- **Efficiency**: We make use of the alternative graphs proprieties in order to develop innovative retiming, reordering and rerouting algorithms based on global information.

- **Quickness**: We assess the computational complexity of our problem formulation when varying the network structure and density, the level of disturbance, the period of traffic prediction.
Presentation outline

- Problem introduction
- Dispatching support system
- Railway traffic optimization
- Assessment of flexible operations
- Evaluation of green wave policy
- Summary and coming research
General architecture of a dispatching system
Data Loading:

- Actual speed and position of all trains at their entrance in the studied area;
- Infrastructure status (such as track layout, signals, switches and speed limits);
- Timetable data;
- Rolling stock characteristics;
- Any additional significant real-time information.
**Disruption Recovery:**
- Discards infeasible routes from a list of prioritized routes;
- Assigns the feasible route with highest priority to each train;
- Emergency timetable to be adopted when dealing with situations of strong disorder.
ROMA* dispatching support system (3)

Real-Time Railway Traffic Optimization:

- Solves the conflict detection and resolution problem by means of alternative graphs and with strict time limits of computation;
- Modifies train routes & orders;
- Reduces delay propagation;
- Improves use of infrastructure capacity at network scale.
Train Speed Coordination:

- Adjusts the speed profiles of the running trains (safe headway distances for any pair of trains);
- Checks feasibility according to the blocking time theory (i.e., UIC norm 406);
Presentation outline

- Problem introduction
- Dispatching support system
- Railway traffic optimization
- Assessment of flexible operations
- Evaluation of green wave policy
- Summary and coming research
Railway Traffic Optimization module

- **Infeasible Schedule**
  - Train (Re)Scheduling
  - Feasible Schedule
  - Rerouting Alternatives?
    - No Rerouting or Time Limit Reached
      - Optimal Orders
    - Possible Improvements
      - Train Rerouting
        - New Routes
        - CDR algorithms: Local Search (LS), Tabu Search (TS)

**CDRFR** algorithms:
- Heuristics (e.g. FCFS, AMCC)
- Branch and Bound (B&B)

**CDR** algorithms:
- Local Search (LS)
- Tabu Search (TS)

*Conflict Detection and Resolution with Fixed train Routes*
Initial assumptions for the CDRFR problem

- The sequence of block sections for each train is prescribed
- The running time of a train on a block section is an operation
- Each operation performs a running time, depending on train speed
- Each block section can host at most one train at a time
Primary delay

Station A
Station B
Station C
Station D
Station E
Station F

Current time
Time

Conflicts
Delay
Primary delay

Consecutive delay

Station A
Station B
Station C
Station D
Station E
Station F

Current time
Time
Alternative Graph (AG) formulation

Alternative graph formulation
[Mascis & Pacciarelli EJOR 2002]

Blocking time separation
[Hansen & Pachl Book 2008]
Illustrative example (1)

CDRFR formulation of a small example with three trains

Each alternative pair is used to order two trains on a block section
Illustrative example (2)

Optimal CDRFR solution computed by the B&B algorithm

A conflict-free deadlock-free schedule is a complete consistent selection $S$
Optimal solution to the compound CDR problem

A new route for TA and a new complete consistent selection S are shown.
**Approach:** Branch & Bound (optimal or near-optimal solutions)

**Objective function:** Minimization of the maximum consecutive delay at each relevant point [D’Ariano et al. EJOR 2007]

**Branch & Bound method:**
Intermediate nodes are partial solutions (LB), final nodes are complete solutions (UB).
Branching: if LB > UB then a solution is not considered.
Initial heuristics

**FCFS** (First Come First Served) dispatching rule:
Give precedence to the train arriving first at the block section.

**FLFS** (First Leave First Served) dispatching rule:
Give precedence to the train that would leave the block section first.

**AMCC** (Avoid Most Critical Completion time) heuristic:
Choose the arc companion of the most critical for all the unselected alternative pairs, and continue until a complete selection is built or a positive length cycle (deadlock) is detected in AG.
Branch and bound algorithm

Branching rule: Choose the most critical unselected alternative pair and branch on this pair.

Hybrid search strategy: Alternate three repetitions of the depth-first visit with the choice of the open node of the search tree with smallest lower bound among the last nine open nodes.


Implications rules: General network topology and graph proprieties
Implication rules

$$((i,j),(h,k)) \in A$$

$$l^S(j,i) + a_{ij} > 0 \Rightarrow (h,k) \text{ implied by } S$$

$$r_i^S + a_{ij} + d_j^S > UB \Rightarrow (h,k) \text{ implied by } S$$

$J$: set of operations, $c$: operation, $\{c\} \cup J$ must be processed on the same machine

$$\min_{j \in J \cup \{c\}} r_j + \sum_{j \in J \cup \{c\}} p_j + \min_{j \in J} d_j > UB \quad \Rightarrow c \text{ must be processed after } J$$

if $$l^S(c,h) + a_{hk} + \max_{j \in J} \{l^S(k,j) + p_j\} > 0 \Rightarrow (h,k) \text{ is forbidden}$$
We initially propose a train routing optimization algorithm based on a local search (LS) procedure [D’Ariano et al. TS 2008] as follows:

- A move is to change one route and its evaluation is to solve the associated CDRFR problem;

- At each iteration the best (local) move is taken from a set of neighbours of a current CDRFR solution;

- When all local moves are non-improving, a local minimum is reached and the procedure stops.
Neighbourhoods

**Initial approach:** Complete neighbourhood

**Smart approach:** It is well known that a job shop scheduling solution can be improved by changing the critical path $C(S)$ related to the current graph selection $S$ only [Balas OR 1969].

We use of the critical path concept to design focused neighbourhoods.

Backward $B(S)$ and forward $F(S)$ ramified critical paths follow.
Illustrative example of ramified critical paths

\[
C(S) = T_{A}^{1} \rightarrow 10 \rightarrow T_{B}^{2} \rightarrow 10 \rightarrow T_{B}^{3} \rightarrow 10 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 10 \rightarrow 14 \rightarrow 10 \rightarrow \text{out} -131
\]

\[
B(S) = T_{A}^{1} \rightarrow 10 \rightarrow T_{B}^{2} \rightarrow 10 \rightarrow T_{B}^{3} \rightarrow 10 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 10 \rightarrow 13 \rightarrow 10 \rightarrow 14 \rightarrow 10 \rightarrow \text{out} -131
\]

\[
F(S) = T_{A}^{1} \rightarrow 10 \rightarrow T_{B}^{2} \rightarrow 10 \rightarrow T_{B}^{3} \rightarrow 10 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 10 \rightarrow 13 \rightarrow 10 \rightarrow 14 \rightarrow 10 \rightarrow \text{out} -131
\]
Tabu search algorithm

The ramified critical paths are well focused on reducing the maximum consecutive delay but are not always opt-connected.

An novel **tabu search** (TS) algorithm [D’Ariano et al. TRpB 2008] escapes from local minima by taking a non-improving move and then forbidding the inverse move for a given number of iterations.

Another technique to escape from local minima is based on **restarts** (i.e., performing a few moves regardless they are good or bad).
Test case: Utrecht-Den Bosch dispatching area

- Utrecht-Den Bosch railway network (50 km long, including 21 station platforms)
- 40 running trains per hour (timetable 2007)
- Rolling stock connections are located in Zaltbommel and Den Bosch stations
- Rerouting is performed in stations and corridors (356 local routes)
Results on the compound CDR problem (1)

Percentage of maximum consecutive delays for four ROMA configurations

<table>
<thead>
<tr>
<th>Average Results (in seconds)</th>
<th>Default</th>
<th>Routing</th>
<th>Optimization*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Delay</td>
<td>Delay</td>
<td>Delay</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>Avg</td>
<td>Tot</td>
</tr>
<tr>
<td>ARI</td>
<td>489.4</td>
<td>66.9</td>
<td>0.6</td>
</tr>
<tr>
<td>B&amp;B</td>
<td>279.8</td>
<td>50.4</td>
<td>2.1</td>
</tr>
</tbody>
</table>

*Routing Optimization by the local search algorithm
Results on the compound CDR problem (2)

performance handling perturbations

- Local Search
- Restart *
- Complete *
- Hybrid 1 *
- Hybrid 2 *
- Hybrid 3 *

Perturbations are multiple train delays

Disruptions are tracks which are blocked

performance handling disruptions

- Local Search
- Restart *
- Complete *
- Hybrid 1 *
- Hybrid 2 *
- Hybrid 3 *

*Routing Optimization by the tabu search algorithm
Presentation outline

- Problem introduction
- Dispatching support system
- Railway traffic optimization
- Assessment of flexible operations
- Evaluation of green wave policy
- Summary and coming research
Railway Dynamic Traffic Management (RDTM*)

- Flexible use of platforms island: The timetable defines a *set of feasible platform tracks* for each train and for each station.
- Flexible sequencing: The timetable defines a *partial order* of trains at overtakes and junctions, the actual order being defined in real-time.
- Flexible timetable: Exact train arrival/departure times are replaced by *time windows* of [minimum, maximum] arrival/departure times.

*[Schaafsma TRAIL PhD Thesis 2001]*
RTDM: Research background

- 2003 –2004 Feasibility study and simulation
- 2005 –2007 Prototype RDTM
- 2007: First implementation of routing and sequencing flexibility at the Schiphol bottleneck
- Agreement with train companies operating at Schiphol about how managing train sequences (previous agreement was that a delayed train cannot obstacle on-time trains)
- New passenger information system (static and dynamic info)
- Introduction of RDTM functions into the Dutch railway traffic control system “Procesleiding”
Passenger information system

Static information

Dynamic information
RTDM: Exploring its potential

- Flexible sequencing (as implemented up to now): First Come First Served (FCFS) rule

- Timetable flexibility: Not implemented but two timetables:
  1. A working timetable (for dispatchers and drivers) with time windows of [min, max] arrival/departure time
  2. A public timetable (for passengers) with maximum arrival time and minimum departure time

- In this talk we explore the benefits of advanced scheduling algorithms in combination with flexible timetables
Flexible timetable

Time windows

Station A
Station B
Station C
Station D
Station E
Station F

Time
Flexible timetable

Station A
Station B
Station C
Station D
Station E
Station F

Current time
Time

Conflicts
Delay
Flexible timetable

Current time

Time

Station A
Station B
Station C
Station D
Station E
Station F

Delay
Real-time conflict resolution
Test case: Dispatching area around Schiphol

- Schiphol railway network (40 km long, around Amsterdam international airport)
- 54 running trains per hour (highly used capacity, dense traffic)
- Some block sections are used in double direction (nearby HfdD and Asdl)
Comparison of scheduling algorithms

FCFS fails 10/300 times

FLFS fails 14/300 times

AMCC fails 23/300 times, (when it finds a solution this is often optimal)

Truncated B&B computes proven optimal solutions in 297/300 cases within 2 min. Optimal solution always found within 30 seconds
Results on flexible timetables

![Graph showing flexibility and delays for different scheduling methods: AMCC, FCFS, FLFS, and B&B.]

- **AMCC**: FCFS reduction 16%
- **B&B**: FCFS reduction 13%, B&B reduction 32%
- **FCFS** and **B&B** show significant reductions in delays compared to AMCC.
Presentation outline

- Problem introduction
- Dispatching support system
- Railway traffic optimization
- Assessment of flexible operations
- Evaluation of green wave policy
- Summary and coming research
Traffic Management policies - WIC

With a WIC (Wait In Corridors) policy, trains are allowed to wait both in stations and along the lines between stations. Thus, they can meet yellow (and even red) signal aspects (according to the three-aspect Dutch signalling system).
Traffic Management policies - GW

With a GW (Green Wave) policy, trains are allowed to wait only at their scheduled platform stops.

Each train may depart from a station only if this is able to reach the next station by running at its scheduled speed (i.e., with green signal aspects only, in absence of conflicts).
Blocking time theory

a) Blocking time graph

- Sight & reaction time
- Approaching time
- Running time
- Clearing time
- Switching time

Distance

Block section length

Time

$t_i$

$t_j$

$p_{\mu(i)}$

$p_i$

$p_{\mu(j)}$

$p_j$

$\beta$

$a_{ij}$
Blocking time theory – WIC formulation

a) Blocking time graph

b) Alternative graph with the WIC policy

Distance Time

Sight & reaction time
Approaching time
Running time
Clearing time
Switching time

Block section length

\[ t_i \]
\[ t_j \]

\[ a_{ij} \]

\[ t_{\mu(i)} \]
\[ p_{\mu(i)} \]
\[ t_i \]
\[ p_i \]
\[ t_{\sigma(i)} \]

\[ t_{\mu(j)} \]
\[ p_{\mu(j)} \]
\[ t_j \]
\[ p_j \]
\[ t_{\sigma(j)} \]
## Blocking time theory – GW formulation

### a) Blocking time graph

- **Distance Time**
  - Sight & reaction time
  - Approaching time
  - Running time
  - Clearing time
  - Switching time

- **Time**
  - Blocking time train 1
  - Blocking time train 2

- **Block section length**

### b) Alternative graph with the WIC policy

- **Switching time**
- **Sight & reaction time**
- **Approaching time**
- **Running time**
- **Clearing time**

### c) Alternative graph with the GW policy

- **Switching time**
- **Sight & reaction time**
- **Approaching time**
- **Running time**
- **Clearing time**

**Formulas**:

1. Blocking time graph
   - \( t_{\mu(i)} \) to \( t_{i} \) and \( t_{\sigma(i)} \)
   - \( p_{\mu(i)} \) to \( p_{i} \)
   - \( p_{\mu(j)} \) to \( p_{j} \)
   - \( a_{ij} \)

2. Alternative graph with the WIC policy
   - \( t_{\mu(i)} \) to \( t_{i} \) and \( t_{\sigma(i)} \)
   - \( p_{\mu(i)} \) to \( p_{i} \)
   - \( p_{\mu(j)} \) to \( p_{j} \)
   - \( a_{ij} \)

3. Alternative graph with the GW policy
   - \( t_{\mu(i)} \) to \( t_{i} \) and \( t_{\sigma(i)} \)
   - \( p_{\mu(i)} \) to \( p_{i} \)
   - \( p_{\mu(j)} \) to \( p_{j} \)
   - \( a_{ij} \)
Illustrative example

Time 0

Stop

Extra delay due to train speed adjustments
Preliminary results: Energy versus Delay

Computation times:
- Schiphol network: ~35 sec WIC, ~100 sec GW
- Utrecht-Den Bosch network: ~2 sec WIC, ~1 sec GW
Presentation outline

- Problem introduction
- Dispatching support system
- Railway traffic optimization
- Assessment of flexible operations
- Evaluation of green wave policy
- Summary and coming research
Main achievements

**ROMA** is a laboratory dispatching support tool applicable to:

- Precisely forecast railway traffic and delay propagation
- Pro-actively support dispatchers in case of disturbances

**Some of the innovative scientific contributions:**

- A new model based on alternative graphs & blocking time theory
- Fast and effective scheduling and rerouting algorithms for the real-time management of complex and dense railway networks
- Assessment of flexible operations and traffic management policies
- Development of advanced train speed coordination strategies
Current/future research directions

**ROMA** applications and new features:

- Algorithms for feasibility recovery and quick response to delays
- Develop a rolling horizon approach for real-time traffic control
- Link the traffic optimizers with microscopic simulation tools
- Study the implementation of new tracks and signalling systems
- Railway traffic management in other European networks
- Traffic control policies for passenger and train point of views
- Robustness to stochastic disturbances and network disruptions
- Integration of railway planning and traffic control tools
Further steps towards its practical application

• Demonstration/evaluation of effects of the proposed dispatching solutions by a microscopic traffic simulator (e.g. FRISO)

• Establishment of a comprehensive closed-loop railway traffic monitoring and control system that requires the transmission of all the relevant information during operations (e.g. by automatic train detection data, GPS systems or other ICT)

• Implementation of a user-friendly interface by means of easy to manipulate effectiveness indicators (e.g. regarding the schedule evaluation or the minimization of energy consumption)

dariano@dia.uniroma3.it