Seamless integration of Microsimulation and Ant Colonies: A new tool for automatic timetable generation

Nicola Coviello, Giorgio Medeossi, Thomas Nygreen
Presentation outline

1. Strategic network planning in Norway
2. Jernbanedirektoratet’s research on automatic timetabling
3. The Multi-Objective Automatic Timetable Generator
4. The timetable planning process with the MOACO
5. Integration into the Treno planning suite
6. Timetable generation and simulation: live demo
Strategic network planning

Constraints

Market

Cost–benefit analysis

Transport model

Concept

Iterate?

Implement?

Timetable

Rolling stock

Infrastructure
Motivation

Strategic timetabling (and timetabling in general) is very time-consuming work, requiring skilled timetable planners.

A medium-sized strategic timetabling-project normally involves months of timetabling, and a large project takes years.

Timetabling is a mix of high-level decision-making and low-level modelling, identifying possible partial solutions, constraint-checking and conflict-solving.

The more of the low-level work we can automate, the better we can use our limited human resources, finding even better strategies in less time.
Previous studies

2014* Jernbaneverket SINTEF

2018–2019 Jernbaneverket SINTEF

2021 Jernbanedirektoratet trenolab Université Gustave Eiffel

* See Lamorgese, L., Mannino, C. and Natvig, E., 2017. An exact micro–macro approach to cyclic and non-cyclic train timetabling. Omega, 72, pp. 59–70.
Modelling the timetabling process

A timetable can be built by a sequence of a base pair of actions:

<table>
<thead>
<tr>
<th>#</th>
<th>Action</th>
<th>Leading criteria</th>
</tr>
</thead>
</table>
| 1  | The selection of the next train to be scheduled, including the decision whether to schedule or not an optional train | • Trains’ priority  
• Conflict probability w.r.t. to already scheduled trains |
| 2  | The scheduling of a certain train group within a timetable already populated by previously set train-paths | • Conflicts  
• Travel time  
• Stability  
• Energy consumption |

Pairs of Actions 1 and 2 are repeated, until a path is designed for each train to be scheduled.
Then the process re-starts, paths are scheduled another time, taking advantage of previously gathered «experience».
A Multi-objective perspective

Timetable KPIs can be used to «measure» and compare timetables.

- Total travel time
- Total energy consumption
- Stability
- Weighted number of scheduled trains
- (Number of residual conflicts)

More than one feasible timetable can satisfy a given service concept on a given infrastructure, depending on the KPIs to be prioritized.

User is provided with a Pareto Optimal Set of feasible timetables, i.e. those timetable which better exploit available capacity.

![Diagram showing relationship between total travel time and number of trains]
ACO fundamentals

The problem is represented by a Construction Graph. Each node is a solution component. At each iteration, a set of agents (ants) build solutions on the CG.

A solution is incrementally constructed by selecting a path or a clique on the CG.

A new node is stocastically chosen with a probability which depends on:

- An heuristic factor (can be statically or dynamically computed)
- A pheromonal factor, which models the learned desirability of that node

Pheromone is laid at the end of each iteration on by the best performing ants.

Search in following iterations is guided towards a neighbourhood of the best solutions found so far.
Infrastructure model

For each used rolling stock:
- Min tech. run times
- Min tech. headways
- Max energy consumption
Timetable model

The service concept

[Diagram showing a time-space model with stations Heimdal, Melhus, and Støren]
Timetable model

The service concept

<table>
<thead>
<tr>
<th>Station</th>
<th>Arrival time admissibility range</th>
<th>Departure time admissibility range</th>
<th>Optional stop</th>
<th>Min stop time</th>
<th>Max stop time (in case of stop)</th>
<th>Min run time</th>
<th>Max run time</th>
<th>Min/Max stop time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heimdal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Melhus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Støren</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A two-layers MOACO

- Layer 1: which train is to be scheduled (or not) next

- Layer 2: how it is actually scheduled. In each location: station, track, stop/pass, stop time. In each line section: run time.

Multi-colony architecture:
- A colony (set of ants) is defined for each active objective, each colony having its own pheromone trails
- Colony’s ants focus mainly on their «particular» objective
- Best solutions update the so far obtained POS
- Pheromone of each colony is updated by the POS’s solutions which better performed w.r.t. the colony’s main objective
Layer 1 construction graph

3 movable groups, 1 optional group

Heuristics:
• Trains’ priority
• Conflict probability with already scheduled ones

Clique pheromonal strategy
Layer 1 exploration

How many additional trains?

How much conflictive with already scheduled trains? (Estimation)

Scheduling in Layer 2
A solution (clique) on layer 1
Layer 2 construction graph

The time-expanded graph

START

Station 1

Station 2

Station 3

END

(Station track; Stop or pass)

DEPARTURE events

ARRIVAL events

END
Run and stop times

Station 1

Station 2

Station 3

space

Time

Min Run Time
Max Run Time

Min Stop Time
Max Stop Time

(Station track; Stop or pass)

START

END
Conflict management

(Station track; Stop or pass)

Station 1

Station 2

Station 3

Space

Time

START

END
Conflict management
Conflict management

No conflict-free path exists!
Conflict management

Only edges belonging to minimum-conflict paths are activated
MILP Refinement

Given a MOACO solution:
• Produces a conflict-free solution (if feasible)
• Further improves the value of the objective functions

Thanks to:
• Time as continuous variable
• Tolerances on the strict periodic grid for each train group
Integration in the planning suite and process

3 tools perfectly integrated: no import/export of files

1. Prepare service concept
2. Run RT calc
3. Run ACO
4. Optimal TTs
5. Edit timetable
6. Simulation
2. Estimate running and headway times in one click

Once the service concept is defined, all inputs for the ACO are calculated on the microscopic model and pushed to the ACO tool.

**RUNNING TIME & BLOCKING TIMES** (default itinerary)

**TIME LOSSES & BLOCKING TIMES** (alter. itineraries, stops)

**HEADWAY TIMES** (based on blocking times) for all trains on all line sections

<table>
<thead>
<tr>
<th>St. A</th>
<th>St. B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. Microscopic simulation

trenissimo is a new synchronous simulation tool designed to make railway simulation more intuitive, accurate, and effective.

Accurate

• Dispatching-driven
• *Explicit* representation of driver behaviour
• Delays and dwell time from real data
• *Explicit* representation of signalling systems

Fast

• Separation of animation and calculation:
• Stochastic simulation on Multi-core computers.
• Command line execution, no windows open
• Parallel execution on multiple computers

Efficient

• Automatic creation of microscopic routing from macro timetable
4000 km of tracks
2400 trains per day
24 hours of operations
simulated in
6 minutes
6. Microscopic simulation

Trenissimo is a new synchronous simulation tool designed to make railway simulation more intuitive, accurate, and effective.

- **Accurate**
  - Dispatching-driven
  - Explicit representation of driver behaviour
  - Delays and dwell time from real data
  - Explicit representation of signalling systems

- **Fast**
  - Separation of animation and calculation:
    - Stochastic simulation on Multi-core computers.
    - Command line execution, no windows open
    - Parallel execution on multiple computers

- **Efficient**
  - Automatic creation of microscopic routing from macro timetable
Conclusions & Outlook

- Working prototype of MOATG tool
- Integrated in state-of-the-art commercial suite
- Under large-scale, real—life testing

- Create&test alternative scenario within one day
- On Bergen line, 3 infrastructure x 6 service concepts tested in 8 hours (>250 optimal timetable generated)

Next steps
- Further improvement of algorithm performance
- Develop methods for bulk analysis of timetables
- Increase the number of KPIs
- Real-life use