Towards Automated Capacity Planning in Railways

Schüpbach, Kaspar\textsuperscript{2}, Caimi, Gabrio\textsuperscript{1} and Jordi, Julian\textsuperscript{1}

\textsuperscript{1}Schweizerische Bundesbahnen SBB
\textsuperscript{2}ELCA Informatik AG

Abstract

As part of the \textit{SmartRail 4.0} program, SBB is focusing with the project TMS (\textit{Traffic Management System}) on algorithmically supported, optimized and integrated capacity planning. In this contribution, we present preliminary results and learnings out of the current proof of concept study.

Keywords: Automated Capacity Planning, Timetabling, Service Intention

1 Introduction

Swiss Federal Railways (Schweizerische Bundesbahnen, short SBB) is pushing ahead digitization and automation of railway planning and operations. Customers are to benefit from higher capacities, less disturbances, better radio communication, improved customer information and lower overall costs. Railway infrastructure utilization is to be increased by shorter headway times and more precise planning. For this purpose, SBB has launched the «SmartRail 4.0» program along the following principles.

- Algorithmically supported, optimized and integrated capacity planning
- Advanced control systems for railway operations
- New generation of digital interlocking systems
- Significant reduction in quantity and diversity of signaling systems
- Network-wide roll-out of the ETCS cab signaling
• Increased data transmission capacity
• Highly available and precise tracking of trains

The SmartRail 4.0 program is organized in 4 principal streams: ETCS-interlocking (ES, from German term ETCS-Stellwerk), Localization, Connectivity and Security (LCS), Automatic Train Operation (ATO) and Traffic Management System (TMS).

In the context of SmartRail 4.0, the TMS-stream strives to reach the following goals:

• Integrated and automated planning
• Automation of the operations centers. The employee develops himself from user to manager of the system.
• Enabling of efficient, real-time and precise automation and control of train movements and speeds.
• Precise coordinated remote control of departure, driving and arrival of trains.

The opportunities for the railway system behind these goals are pointed out in [8].

The program clearly aims for an evolutionary approach towards automation of the planning and operation systems, in order to realize improvements step by step. During transition, it is crucial to take the human factor and the interaction between manual and automated processes into account. Certain roles will have to prepare the inputs and have to understand and post-process computational results. Particularly in the long-term planning process with many commercial and political aspects, the human factor will remain still important for long time, even if also at this stage algorithmic decision support will be important.

The key factor for the success of TMS is to find the right approach which enables on the one hand to have a strong algorithmic performance to solve big instances of the size of Switzerland, and on the other hand to enable a continuous integration in the current processes, with particular focus on the man-machine interaction.

In this contribution, we present preliminary results and learnings out of the current proof of concept study, focusing on the goal ‘Algorithmically supported, optimized and integrated capacity planning’ from the SmartRail4.0 program.
2 Proof of Concept Study

In this contribution, we present preliminary results and learnings out of the current proof of concept study.

First purpose of our study is a stepwise approach to the relevant problem formulation for the new planning paradigm based on service intentions. Content of the service intention and relevant business rules per planning horizon are elaborated in collaboration of SBB railway experts and its IT department.

Second purpose is to evaluate, compare and refine algorithmic solution techniques. In order to facilitate the evaluation of several approaches, we develop a capacity planning framework which provides scenarios, rule checking and visualizations which are independent of the solution algorithm.

Third purpose is the development of intuitive user interfaces and interaction possibilities to support the new planning paradigm. This includes the design of relevant performance indicators and visualizations for service intentions and resulting capacity plans.

2.1 Railway Capacity Planning

Railway capacity planning is about planning the utilization of the railway infrastructure. In this step, the infrastructure manager verifies whether and plans how the intended railway services can be coordinated on the network. Additional to train trips, the capacity plan includes track utilization by shunting, by parked rolling stock as well as reduced capacity due to construction and maintenance sites.

Capacity planning is based on the service intention as an agreement between infrastructure manager and train operating companies [9, 3]. The service intention contains passenger-relevant requirements as the list of train runs with given commercial operation points, given commercial time slots, minimum stopping times and connections. It further contains other capacity-relevant information of the train runs such as rolling stock properties, which affect duration of track occupation.

The capacity planning problem described here is a counterpart to the railway line planning problem, where passenger demand is translated into train lines with frequencies and connections. Line planning is a negotiation process with many stakeholders from train
operating companies, infrastructure manager and authorities. Result of line planning is the service intention. Capacity planning can support the line planning process by validating whether the resulting service intention can be transformed into a feasible capacity plan on given infrastructure.

The goal of capacity planning is to find a plan fulfilling all requirements in the service intention on the available railway infrastructure. This is in contrast to the current planning paradigm, in which each new or changed requirement is directly translated in an adapted plan without formulating the functional requirement in terms of a service intention.

Keeping track of the requirements allows to optimize capacity utilization for each situation separately, whenever service intention or track availability differ, thereby taking advantage of the degrees of freedom allowed by the service intention. Algorithmic support allows to create exact capacity plans for each day and hour of operation, while the future job of the capacity planner will be to focus on designing and negotiating producible, flexible and stable service intentions.

Note that the line plan will typically contain a large fraction of periodic trains (passenger trains repeated once or twice per hour), resulting in periodically repeated requirements in the service intention. There is no requirement, however, for the resulting capacity plan to be periodic. Even the trains with periodically repeated constraints may have a different schedule and route each time. This allows to optimize each individual hour of each operation day individually, while the commercial times communicated towards the passengers remain periodic and easy to remember. As remarked in [2], periodicity has a big value towards the customer, but better plans are possible when omitting constraints for periodicity and symmetry. This is particularly valid, when an important part of the requirements are non-periodic, such as construction intervals and cargo trains.

2.2 Infrastructure Topology Model

In a first approach to algorithmic capacity planning we are focusing on scheduling train trips with predefined routes within given time slots. The resulting schedule needs to be conflict-free with respect to occupations of infrastructure resources.

Part of the goals of the study is to find the right level of topology detail required to produce relevant capacity plans in reasonable computation time. The reasonable level of detail
may vary depending on the planning horizon considered. Potentially, an adaptive level of topology detail can be used, where more details are used in bottleneck areas operated close to capacity limit, and more details are added into the plan the closer operation day comes.

In order to test various topology levels, we are using a generic resource occupation model which allows for both detailed and aggregated resources. The predefined routes per train are given as a sequence of route sections, each with defined minimum trip time and a list of resources occupied on the section. A resource is a generic topology element which can be occupied by a train along its run. It usually corresponds to a route section, e.g. a signaling section, a station track where a train stops or also a longer distance tracks between two major stations. A resource can either be of type headway resource or of type blocking resource:

**Blocking Resources** are the more detailed variant as used to model blocking time stairways. For every pair of two route sections of two different train runs occupying the same blocking resource, the occupation intervals cannot overlap. Differently stated, the first train needs to leave the occupying route section before the second train can enter.

**Headway Resources** are the approximate variant. For every pair of two route sections of two different train runs occupying the same headway resource in the same driving direction, the time difference between the two occupation start times need to be at least the headway time. The same holds, for the occupation end times. Result is, that the second train can in this variant already enter the resource while the first one has not yet left. Headway resources can be used to model long track sections consisting of multiple aggregated signaling block sections. If the driving direction of two consecutive trains on headway resources is opposite, then occupation intervals may not overlap, and the same condition as for blocking resources (see above) holds.

Generic resource definition allows for comparison of microscopic, mesoscopic and macroscopic models. Typical topology models described in the literature are the macroscopic, mesoscopic and microscopic levels as shown in Figure 1.

**Microscopic Topology** is the most detailed level. Separate blocking resources are introduced for each signaling section.
**Mesoscopic Topology** is the intermediate level. Track sections between stations are aggregated into one headway resource. Station tracks are modeled in detail as a blocking resource. Motivation is that station tracks usually have only space for one train at the time, while distance tracks between stations accommodate multiple trains running in the same direction, given that headway time requirements are fulfilled.

**Macroscopic Topology** is the most approximative level. Station tracks are not modeled as resources, thereby assuming infinite capacity in stations. Track sections between stations are aggregated into one headway resource, same way as in mesoscopic level.

![Figure 1: Illustration of microscopic, mesoscopic and macroscopic topology models.](image)

### 2.3 Algorithmic Framework

We have built a framework as illustrated in Figure 2 which allows experimenting with various algorithmic solution approaches and comparison of results. The scenarios (consisting of service intention and train routes), scenario validation, solution validation and visualization components are independent of the solution algorithm.
3 Specific Model and Preliminary Results

We are currently experimenting with a mixed integer linear program (MILP) and a genetic algorithm (GA) formulation. In this section, we describe the model in detail by means of the MILP formulation.

3.1 Description of mathematical model

We use an alternative graph formulation similar to the one described in [4]. The nodes in the graph represent discrete events, in our case the relevant events are the start/end events for track sections (defining resource occupations) by each train together with departure and arrival events for each train stop. A solution to the train scheduling problem is to find event times for each node in the graph, fulfilling the following conditions induced by the arcs of the graph.

- Directed arcs represent time dependencies between events. For each arc, the minimum required time difference (time at destination node minus time at source node) over the arc is defined.
- Directed arcs are used to model minimum travel times over track sections, minimum stopping times at stations and minimum connection times between arrival of one train and departure of the next.
- Alternative arcs pairs are pairs of directed arcs, each with a minimum required time difference. Only for one of the two arcs of an alternative arc pair the required time difference needs to be fulfilled.

Alternative arc pairs are used to enforce minimum headway times for pairs of track sections of two different trains occupying a common headway resource. Also, alternative arc pairs are used to enforce disjointness of occupation intervals for pairs of track sections of two different trains occupying a common blocking resource.

Nodes can have required lower and upper limits on the event times (time slots). The lower time limit is treated as hard constraint. The upper limit is a soft constraint, its violation is penalized in the objective function.

Solution to the alternative graph problem consists of event times for each node such that the required time differences over directed arcs resp. alternative arc pairs are fulfilled, and event times are within the given time slots.

The MILP formulation for this alternative graph problem has the following variables and constraints:

<table>
<thead>
<tr>
<th>Table 1: MILP Model Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbol</td>
</tr>
<tr>
<td>$t_e \geq 0$</td>
</tr>
<tr>
<td>$d_e \geq 0$</td>
</tr>
<tr>
<td>$p_{(a,b),(c,d)} \in {0,1}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: MILP Model Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>constraint</td>
</tr>
<tr>
<td>$t_b - t_a \geq \tau_{a,b}$</td>
</tr>
<tr>
<td>$t_b - t_a + M(1 - p_{(a,b),(c,d)}) \geq \tau_{a,b}$</td>
</tr>
<tr>
<td>$t_d - t_c + M p_{(a,b),(c,d)} \geq \tau_{c,d}$</td>
</tr>
<tr>
<td>$t_e \geq l_e$</td>
</tr>
<tr>
<td>$t_e - d_e \leq u_e$</td>
</tr>
</tbody>
</table>

The alternative arc constraints are so-called big-M coefficient, for which $M$ is a coefficient large enough to always switch off one of the two constrains per pair, depending on the value of the binary precedence variable.
The objective function is to minimize the weighted sum of delays with respect to the given upper time limits $d_e$ with delay factor $v_e$.

$$\min \sum v_e d_e$$

### 3.2 Model Improvements

This far, we have experimented with two major improvements to our initial MILP model as described in 3.1: a hard upper bound on the allowed delay per event and a preprocessing algorithm that determines route sections where no crossing or overtaking is possible and hence precedence decisions are dependent.

#### 3.2.1 Maximum Allowed Delay

The first idea is similar to the *heuristic bound tightening* approach discussed in [5]. We introduce an upper limit $\delta$ on the allowed delays per event. For each $d_e$ variable, we enforce $d_e \leq \delta$. The event time $t_e$ consequently needs to fulfill $l_e \leq t_e \leq u_e + \delta$, in case lower and upper time limits are given for event $e$. The introduction of the maximum delay limit $\delta$ allows to formulate the hard upper time limit $t_e \leq u_e + \delta$.

For events without time limits, the lower limits from previous events of the same train run can be propagated forward along arcs (minimum trip or stop time). Lower time limit $t_e \geq l_e$ is propagated along arc $(e, f)$ with required time difference $\tau_{e,f}$ to yield a lower time limit for $t_f$.

$$t_f \geq t_e + \tau_{e,f} \geq l_e + \tau_{e,f}$$

Similarly, the upper bounds can be propagated backwards.

We can then use the (propagated) time slots to reduce the number of potentially colliding resource occupation intervals. In many cases the time slots already rule out conflicts. In such cases the corresponding alternative arc pairs can be omitted and the number of binary variables reduced in the MILP formulation.
3.2.2 Dependent Precedence Variables

Our initial MILP model contains precedence variables for each pair of route section with common resource occupation for each train pair. This results in a large number of binary precedence variables for each train pair and potentially in a large search space for the MILP solver.

In this improvement we use a preprocessing step to detect continuous sequences of route sections with common resource occupations per train pair. On such sequences, no overtaking or crossing can take place and the ordering of trains is constant over the sequence.

Over such a sequence with constant ordering, the MILP Model can be tightened by introducing precedence dependency constraints. For the set of precedence variables \( p_0, \ldots, p_s \) for the given constant order sequence and the given train pair, we introduce the following constraint

\[
p_0 = p_1 = \cdots = p_s
\]

Note that these additional constraints cut off only infeasible solutions, thereby reducing search space for the solver without reducing the solution space for the model.

3.3 Preliminary Computational Results

We evaluate the effects of the model improvements described in 3.2 on a set of three scenarios. These scenarios all contain the same service intention (i.e. the same list of trains that are to be run, together with their start and end points and desired time slots) but differ in the granularity of the underlying topology used to model resource occupation and conflict detection. Namely, we consider (see Figure 1 for details on the topology models)

- a *macroscopic* scenario (i.e. trains in stations do not occupy any resources and two stations are connected by a single headway resource),
- a *mesoscopic* scenario (same as macroscopic, but trains in a station occupy a single blocking resource),
- a *microscopic* scenario, where the track topology and occupation times are modeled in detail by blocking resources.

The service intention for the scenarios consists of 80 trains on the lines Pfäffikon - Sargans
and Uznach - Näfels, which cross in the common station of Ziegelbrücke. The line is located in the eastern part of Switzerland and has about 80 km of track length.

### 3.3.1 Effect of Maximum Allowed Delay

Figure 3 illustrates the possible gains in the solution time by setting the maximum allowed delay for the trains to a low value.

In this case, all scenarios do admit a solution with zero delay (i.e. all trains can be scheduled within their desired time slots). Thus, even with maximum allowed delay set to zero, the scenarios remain feasible, and in these examples, optimality is achieved roughly two orders of magnitude faster than with a value of 4 hours. For example, the microscopic scenario on average takes about 1 minute to solve to optimality with a maximum allowed delay of 4 hours. However, with the parameter set to 0 minutes, it is solved in roughly a second.

![Figure 3: Effect of the parameter maximum allowed delay on the solve duration.](image-url)
3.3.2 Effect of Dependent Precedence Variables

For the following comparison, we again take the three reference scenarios and set the parameter Maximum Allowed Delay to 30 minutes.  

Figure 4 illustrates the effect of this improvement on the solution time. One can see that for the macroscopic scenario, the effect is minimal. This is not surprising, since in a macroscopic model trains can overtake in every station. For the mesoscopic scenario, the effect on the computation time is already non-trivial. In addition, preliminary tests with larger scenarios suggest that the gains increase the larger the scenarios are (i.e. the number of trains and the lengths of their paths). For the microscopic scenario, finally, the gain is quite large.

Clearly, the more detailed the topology, the more precedence variables are dependent on one another, when two trains share a common path. Therefore, this improvement can eliminate more binary variables from the microscopic model than it can from the mesoscopic model. This explains why the gain in computation time is largest for the microscopic scenario.

![Figure 4: Effect of the parameter dependent precedence variables on the solve duration.](image)

More exhaustive tests have shown that the two improvements combine rather well. That is, both for large and small settings of “maximum allowed delay”, adding the Dependent Precedence Variables improvement further reduces the required solution time considerably. We limit ourselves to one example here, in order to keep the presentation uncluttered.
4 More Models to be examined

In parallel to the approach described in this paper, we also want to examine other models for our challenges. Currently, to our knowledge, no model is known to be able to solve the full railway capacity planning problem, but many models and algorithms tackle parts of it and solve some sub-problems in efficient and elegant ways. These approaches sometimes can be combined, but sometimes are completely different and not compatible.

Our idea in this first phase of the program is to test different strategies and models, and compare the performances of several approaches, with the goal of combining the most promising ones and implementing them into the new production systems.

The goal is to handle the high complexity and the large size of the real instances with the appropriate separation in sub-problems and/or the use of heuristics methods. As in each heuristic, there is no optimality guarantee. However, for our purpose it is much more important to find solutions of reasonable quality quickly, compared to having proven optimality gaps.

We plan to experiment with several of the following approaches:

- Adaptation of the MILP-formulation by fixing a subset of binary variables. This corresponds to some functional decisions (e.g. train sequence) which is fixed in advance based on experience or other functional reasoning. The same effect can be obtained by adding additional constraints to the MILP-formulation based on functional reasons.

- Time Slot-Heuristic: a simple heuristic can be developed starting from the time slot of the service intention. For each precedence variable the train with the earlier begin of the time slot will be given priority. This is a sort of first-come-first-served strategy, which is certainly not optimal but should allow very fast computation times.

- Sequential Routing: Groups of trains are planned in sequence and not all simultaneously. The planned trains in a sequential step will get a fixed combinatorial structure for the following steps, but still can be shifted in time. This approach has been studied in [7] with first very promising results also for instances of large size.

- Local Search: Starting with a feasible solution an approach based on a local search meta-heuristic can be applied. The criteria about the definition of the neighborhood for the local search can be defined based on functional factors like geography, train
category, time slots, connections, or others. This approach enables also parallel computing using different criteria.

- Rolling Horizon Approach: Like in the sequential routing the instance will be solved step-by-step and the result of the previous step will be fixed in its combinatorial structure for the following step. Here, the steps are based on the considered time horizon, following the principles of Model Predictive Control. Every step considers a defined time horizon (e.g. 3 hours) and in the found solution all planning information about the first part of the time horizon (e.g. 30 minutes) will be fixed. Then the time horizon will be extended by the same amount of time and iterated. This approach has been already implemented with success in [6, 1] for the microscopic planning of a local region like the main station area of Bern. It is still an open question if the approach is also powerful for the network wide planning question that we are facing here.

5 Conclusions and Further Steps

We have presented first steps towards automation of the timetable generation process in the applied context of SBB. The presented work is part of the strategically relevant program SmartRail 4.0 at SBB which is about digitization of the railway of the future.

The results presented in this paper are only the very first step in a multi-year program with ambitious goals and still many open questions. This open questions will be addressed with a succession of several proof of concept studies, where we want to validate the application of different approaches and technologies in practice.

The presented very preliminary results are not yet a real contribution to the theoretical and academic knowledges about the topic. Nevertheless, this first step is not only important from a practical perspective, but even more from the point of view of the integration of academic world into the practical application with the prototyping and the proof of concept done with real data and inside the operative IT landscape of SBB.

Furthermore, the program SmartRail 4.0 with the sub-program TMS addresses topics which can be of relevance also for other railway companies and initiatives. With this paper we want to share our intentions, plans and first results and invite community and partners to work together on the challenges of automated railway traffic management.
Literature


Schüpbach, Caimi, Jordi

Authors

Schüpbach, Kaspar
PhD in Operations Research. ETH Zürich, Department of Mathematics, 2012. Several years of experience as Algorithmic Expert and IT Business Analyst in Railways and Logistics.

Caimi, Gabrio
PhD in Operations Research on optimization of dense railway networks. ETH Zürich, Department of Mathematics, 2009. Several years of experience in different positions of railway infrastructure companies (BLS and SBB).

Jordi, Julian
PhD in Mathematics, ETH Zürich, 2010. Since then working as a Business Analyst in the Traffic Operations and Dispatching group of SBB IT.