Introducing an Optimization Model for Timetable Based Railway Network Design

T. Sander[a,*], K. Nachtigall[a]

[a] TU Dresden, Chair of Traffic Flow Science, Dresden, Germany
tim.sander@tu-dresden.de, karl.nachtigall@tu-dresden.de
[*] Corresponding author

Extended abstract submitted for presentation at the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

April 1, 2022

Keywords: Railway Network Design, Railway Timetabling, Railway Capacity, Mixed-Integer Linear Programming

1 INTRODUCTION

Many western-European railway infrastructure operators plan the development of their networks according to demands given by a long-term timetable. While the railway timetabling process on its own has been thoroughly studied and optimized (refer to chapters five and six of Borndörfer *et al.* (2018) or Caimi *et al.* (2017) for an overview) and railway network design has been the topic of various publications as well (e.g. Spönemann (2013) or chapter three of Borndörfer *et al.* (2018)), the combination of the two still leaves room for further research. In this extended abstract, we propose an optimization model for the timetable-based railway network design problem, which integrates timetabling and network design on a macroscopic infrastructure.

2 METHODOLOGY

2.1 Problem Overview

The task is to design a railway network that allows stable operations of a set of trains while respecting the input timetable, which defines time bounds, start and destination for each train. Additionally, the timetable is refined by assigning arrival and departure times for each train and each used arc. We propose an optimization model that minimizes both infrastructure costs and trip durations while

- conserving integral flows
- respecting the given time bounds for each train
- determining a detailed routing and timing on the macroscopic infrastructure
- not exceeding line capacities by respecting train-sequence and train-type dependant minimal headway times

2.2 Modelling Approach

In the proposed model, we define an input timetable as a set of integral flows travelling from a source node to a destination node, each having time bounds on both ends that need to be respected. Each flow represents a train trip and has a train type (e.g. local, intercity, freight), on which the travel times depend. It is also possible to further specify the routing by including via-stations into the timetable. Apart from that, the trains can be routed freely through the network, which is defined as a multi-graph $G = (N, A)$ featuring nodes $i \in N$ and arcs (a) or $(i, j, tr) \in A$. Nodes represent stations or interlockings and provide the opportunity to change from one arc to another or to wait for a free time slot on the next arc. Currently, node capacities are not limited. The arcs (i, j, tr) are identified by their origin node i, their destination node j and their track number tr and represent one track of a railway line. The construction of multiple parallel tracks is possible. Each arc is considered bi-directional, which allows modeling both single-track lines as well as double or multi-track lines. The inclusion of an arc into the network, indicated by decision variable $y_{i,j,tr}$, comes at the expense of associated fixed building costs $f_{i,j,tr}$. Pre-existing tracks can be modelled by setting their building costs to zero. On each arc, train-type dependant travel times $t_{k,i,j}$ are in place and a specific matrix of minimal headway times can be assigned.

To estimate the number of parallel tracks necessary for the operation of the input timetable, minimal headway times (MHT) are used to evaluate the consumed capacity on each track. Since these headway times and as such the railway capacity in general depend on different factors, the optimization model features various possibilities to model capacity extensions, namely the inclusion of another parallel track and the reduction of travel and headway times.

The opportunity to reduce travel and headway times in the model represents changes to the railway capacity which don't come at the cost of a brand new track, e.g. updating the train control system. They are modelled by specific reduction variables r^{time} and r^{MHT}, which come with associated costs c^{time} and c^{MHT} for each unit of time reduction. These variables are strictly bound to avoid unrealistic results.

The timetabling aspect is modelled by variables $d_{k,i,j,tr}$ and $a_{k,i,j,tr}$, which define the departure and arrival times of train k while travelling on arc (i, j, tr). Whether a train is using a certain arc or not is specified by binary decision variable $x_{k,i,j,tr}$.

2.3 Optimization Model

The model features an objective function that minimizes both infrastructure costs (for building tracks and reducing trip or headway times) and trip times. The minimization of trip times uses the difference between arrival and departure time for each train $\sum a_k - \sum d_k$, which includes both travel times and dwelling times at nodes.

$$min \sum f_a y_a + \sum c^{time} r^{time} + \sum c^{MHT} r^{MHT} + \sum a_k - \sum d_k \tag{1}$$

The objective function (1) is subject to three different sets of constraints, each responsible for a specific aspect of the optimization. The first block deals with the network design aspect and features a flow conservation constraint, a linking constraint that makes sure, that trains are only travelling on arcs included in the network and a track sequence constraint, which assures that the parallel tracks on a line are built in ascending order.

The second block deals with the timetabling aspect. It comprises constraints that make sure, that the difference between arrival and departure times on an arc equals the travel time minus the optional reduction variable. There are also constraints to ensure, that the time bounds given in the input timetable are not violated. Last but not least, the timetabling block features constraints that enforce that a train doesn't leave a station before it arrives there.

The minimal headway times are enforced in the last set of constraints. Here, an additional decision variable z_{i,j,tr,k_1,k_2} is introduced, which is set to one if trains k_1 and k_2 are running on the same track and if train k_1 is running before k_2. In this case, the departures of the two trains have to be separated at least by the headway time defined for the arc (i, j, tr) and the combination of train types. The enforcement of headway times is modelled using a Big-M Constraint, which is en- or disabled by the decision variable z. The headway time constraint set exists twice, once for trains using the same track in the same direction and once for trains running on the same track in opposite directions.

2.4 Solution Approach

Since the number of variables becomes very large and the model is quite complex, a two-stage solution approach has been developed. In the first stage, only the x and y-variables are considered and many constraints dealing with timetabling and headway times are relaxed. They are replaced with two constraints that ensure time bounds and capacities in a more general way. The time bounds are enforced by a constraint that makes sure, that the sum of travel times of each train doesn't exceed the time window given by the input timetable. Instead of the minimal headway times, a basic worst-case capacity measure is introduced. It limits the number of trains running on a track by assuming that the longest possible headway time, which occurs on a track if the fastest train type follows the slowest, has to be respected between all pairs of trains. With these constraints in place, the routing and a basic network are calculated in the first optimization step. Since the time bounds might require train crossings between stations and therefore at least one track per direction, the result from the first stage is altered to make sure, that on each line at least two tracks are constructed. This basis network features more arcs than necessary and will be refined in the second step, once the more detailed and favourable capacity estimation is used. The second stage introduces the timing variables a and d, the reduction variables r^{time} and r^{MHT} and the headway time variables z. Since these are only created for the network which resulted from the first stage and not for all possible arcs, their number and with them the calculation time of the second stage can be greatly reduced by this two-stage approach. The second stage is implemented as a new optimization model which also discards the x and y-variables that were not used in the first stage's solution. This has proven to be beneficial for the performance instead of simply adding the additional variables and constraints to the existing model.

3 CASE STUDY

To test the model, prove its functionality and evaluate its performance, a small case study based on data from the German nation-wide timetable concept, the Deutschlandtakt, has been created. The network consists of 110 nodes and 149 lines, on which at most 5 parallel tracks can be constructed. Input timetables with different amounts of trains have been tested. The model has been implemented in Python 3.8 and solved with Gurobi 9.1.2 on a Lenovo Thinkpad T490 Laptop featuring an Intel i7-8565U CPU and 16 GB of RAM. A short overview of some computational results as well as data for the model size can be found in table 1. A typical result can be seen in figure 1, with the network after the first stage on the left and after the second stage on the right-hand side.

Table 1 – *Computational results*

Trains	St.1 - Vars	St.1 - Cons	St.1 - Time[s]	St.2 - Vars	St.2 - Cons	St.2 - Time[s]
22	38717	9790	0,83	6035	14806	22,55
40	76985	25342	253,32	18336	29637	20.913,91
60	127105	51891	27,3	25488	37901	2616,82
92	280033	154254	407,54	82685	111614	no Sol. (24 h)

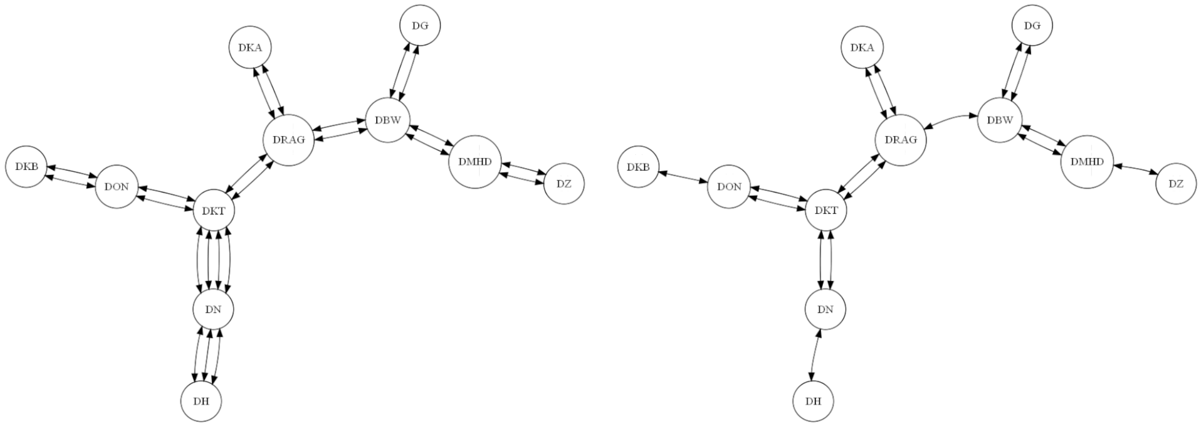

Figure 1 – *Exemplary network after first (left) and second stage*

It is worth noting, that the computation times shown in table 1 are not only dependent on the train count, but also on various factors not included in the table, e.g. the type of trains, the trip length and how the trains spread over the network.

4 OUTLOOK

The presented model provides the base for further extensions in various directions. On one hand, the model is going to be expanded in a way that allows the consideration of the robustness of the network against timetable changes. Various methods including worst-case robust and stochastic optimization with demand uncertainties are currently evaluated for this purpose. One promising way to model network robustness is by including a set of optional trains and performing various tests with them, e.g. by introducing a trade-off between infrastructure cost and penalties for non-travelling trains or by creating scenarios by randomly choosing a certain number of optional trains that become mandatory if chosen.

On the other hand, the model is going to be extended to better capture the specifics of railway operations, e.g. by including mandatory buffer times, introducing capacity restrictions to ensure stable operations or considering basic measures for node capacity restrictions. Apart from functional changes, the computational results and especially the failure to calculate a solution for the second step of the fourth test case show that performance improvements are necessary. To reduce optimization time, both aggregation and decomposition algorithms will be evaluated and implemented in the future.

References

Borndörfer, Ralf, Klug, Torsten, Lamorgese, Leonardo, Mannino, Carlo, Reuther, Markus, & Schlechte, Thomas (eds). 2018. *Handbook of Optimization in the Railway Industry*. Vol. 268. Springer Verlag.

Caimi, Gabrio, Kroon, Leo, & Liebchen, Christian. 2017. Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, **6**(4), 285–312.

Spönemann, Jacob Christian. 2013. *Network design for railway infrastructure by means of linear programming*. Ph.D. thesis, Hochschulbibliothek der Rheinisch-Westfälischen Technischen Hochschule Aachen, Aachen.

5 ACKNOWLEDGEMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project ID 432300662