HIERARCHICAL DECOMPOSITION METHODS FOR PERIODIC RAILWAY TIMETABLING PROBLEMS

Sabrina Herrigel (corresponding author)
ETH Zurich, Institute for Transport Planning and Systems (IVT)
Wolfgang-Pauli-Strasse 15
8093 Zurich, Switzerland
Telephone: +4144 632 7519
sabrina.herrigel@ivt.baug.ethz.ch

Marco Laumanns
IBM Research – Zurich
Säumerstrasse 4
8803 Rüschlikon, Switzerland
Telephone: +4144 724 8342
mlm@zurich.ibm.com

Andrew Nash
Vienna Transport Strategies
Vienna, Austria
andy@andynash.com

Ulrich Weidmann
ETH Zurich, Institute for Transport Planning and Systems (IVT)
Wolfgang-Pauli-Strasse 15
8093 Zurich, Switzerland
Telephone: +4144 633 3350
weidmann@ivt.baug.ethz.ch

5234 Words, 7 Figures, 1 Table

Submission Date: November 15th 2012, Zurich

ABSTRACT

Today many European railway networks are operating near capacity. Developing timetables for these dense and often highly congested networks is becoming increasingly difficult. Several algorithmic approaches for solving timetabling problems have been developed in recent years, but the problem size, computational complexity and lack of transparent interfaces for planners slow down adoption of these approaches in practice. This research proposes an iterative method based on train hierarchies to solve large periodic timetabling problems. The proposed method adds a new group of trains to the schedule in each step of the process while holding trains added in previous steps fixed within a specified time interval. Using a case study with real-world data, the influence of the number of decomposition steps and time interval on computation time and timetable quality is analyzed. The results show that setting parameters to a compromise between the extremes of a purely sequential or a purely simultaneous timetable planning approach is very effective at reducing computation time while still providing optimal or close to optimal timetables.

1 INTRODUCTION

Railway networks are increasingly important for providing transportation services for goods and passengers. Sustainable and cost-efficient railway transport benefits society, economy and environment. Recent growth in railway traffic has strained network capacity, and constructing new railway infrastructure is expensive and disruptive. Together this means that the current railway capacity be used as efficiently as possible.

Over the last two decades railway scheduling has been intensively studied in the academic world and many models and algorithms have been developed. These new tools help to increase the efficiency of railway operations and planning. Especially important are algorithms supporting timetable and infrastructure planning. Using these algorithms in computer-supported timetable planning allows planners to compare different timetable concepts and infrastructure extensions more quickly than when this work was done by hand. However, the inherent complexity of timetable design continues to limit their computational practicability to problems of moderate size. In particular, the automatic generation of a complete timetable for an entire network remains a challenge.

There are two basic approaches for timetable construction: sequential and synchronous planning. Sequential timetable planning means planning a single train, considering that train path fixed, and then planning the next train. In contrast, synchronous planning means that all the train paths are planned simultaneously in a single step. Given human cognitive limits, manually constructed timetables are often planned using the sequential approach, train by train. In this approach, planners look for possible additional slots in a partially fixed timetable. Several software tools have been developed to automate this process and are currently used to help plan timetables [6, 7]. This approach of fixing train paths sequentially is also used in so-called asynchronous railway simulation frameworks.

Several automated synchronous timetable planning algorithms have also been developed [9], [10], [17]. They have the advantage of considering a timetabling problem as one mathematical problem and therefore do not lose potential (including optimal) solutions through early, maybe disadvantageous, train path fixations. Unfortunately, high computation times for large instances still hinder the integration of synchronous planning algorithms into planning software used in practice.

The benefit of sequential timetable planning is a significant reduction in computing time compared to synchronous timetable planning. But, since trains are scheduled one-by-one and schedules of previously planned trains are held fixed, many possible combinations of different train paths – and therefore potential solutions – are overlooked. Furthermore, the algorithms may not even be able to find an existing solution. On the other hand, synchronous timetable planning is difficult for complex networks with many train operations. It would be ideal to have an approach that combines the benefits of both sequential and synchronous planning.

In this paper, we develop an approach to timetable planning that combines sequential and simultaneous planning for a special type of schedule: the periodic timetable. This multi-step approach, called hierarchical decomposition, reduces complexity by sequentially planning timetables for groups of trains rather than individual trains and increases quality by allowing previously planned train schedule times to vary somewhat as the next group of trains is planned.

The first step in this approach consists of dividing train services into groups (often based on service quality, for example stopping patterns and speed). The groups often reflect railway priority (e.g. high priority train groups are planned before lower priority groups).

Next a timetable for trains in the first group is planned using synchronous methods. The paths for these trains are fixed within a given time window around the timetable computed by the synchronous model, and then a timetable for the next set of trains is sought. As the timetable for this next set of trains is computed, the previously planned train paths can be moved within the time window to increase the ability to create good paths for the next set of trains. The process is repeated until all the groups have been planned (or it is impossible to add another train to the schedule).

The method was developed by adding a scaling parameter to a formal generic hierarchical timetabling scheme. The scaling parameter allows users to adjust the level of simultaneousness. This scaling parameter is the time window within which train paths can vary and essentially specifies the degree of freedom given to adjusting previously planned train paths when planning lower-priority trains. Using a case study for planning a periodic timetable for a part of the Swiss railway network, we investigate how the time window parameter and the number of train groups influences computation time and solution quality.

The next section provides more background on periodic timetables and the Periodic Event Scheduling Problem (PESP) model for computing periodic timetable schedules. Section 3 describes solution methods for the PESP. Section 4 describes the generic hierarchical timetabling scheme developed, and Section 5 investigates how the time window parameter impacts schedule quality using this scheme for different priority groups. Finally, Section 6 presents conclusions and recommendations for further research.

2 THE PESP MODEL

This research focuses only on periodic timetables. In a periodic timetable the pattern of trains repeats after each period. This type of timetable is easier for customers to remember and also reduces the complexity of schedule planning [8]. Instead of performing computations for a whole day, the main planning process is reduced to planning only one representative hour (or period). This one hour, repeated consecutively over a whole day, serves as a basic concept for the daily timetable. Small adaptions such as removing a train service in off-peak hours and adding individual runs for freight trains are handled after the rolling out of this basic hour.

Periodic timetables have been used in the Netherlands, Switzerland and German for many years and are gradually being introduced in other European railways. Switzerland's comprehensive introduction of periodic timetabling beginning in 1982 has been one of the main factors behind the country's strong passenger growth. The systematization of periodic timetabling has led to the introduction of integrated, fixed interval nodes for major stations of Switzerland's railway network. This means that trains arrive at stations just before the hour (or half-hour) and depart just after the hour, giving passengers easy connections between all lines. Switzerland's success is a good example, and many other European countries are starting to systematize their national passenger railway services as well. Given this trend, we develop our approach for periodic timetables. Using the concept of partial periodic PESP [4] it is possible to adapt the same ideas to non-periodic timetabling problems.

The periodic event scheduling problem (PESP) has been shown to be a useful model for cyclic timetable planning several times in practice and is therefore also used in this research. The PESP as a general scheduling framework was introduced by Serafini and Ukovich in 1989 [20] and was used to model train timetables by Shrijver and Steenback [19] for the first time five years later. The PESP allows users to plan a set of specified train lines with desired functional requirements and restrictions given by the infrastructure. The next sections describe the data needed to construct a PESP model and then explain how this information is represented in the model.

Necessary Data to Construct a PESP Model

Macroscopic Topology

The first step in developing a PESP model for railway timetable planning is to define the level of detail for the (macroscopic) infrastructure model by specifying a set of stations, line junctions and crossing points. The elements of this set constitute the *nodes* of the macroscopic infrastructure. A node is needed for each station where (i) train sequences are allowed to change, (ii) train lines end, or (iii) we want to offer connections for passengers. Additional nodes must be placed at important line junctions outside of stations, where trains can change tracks (e.g., to overtake), or where the number of parallel tracks changes. It is possible to vary the level of detail in the macroscopic topography depending on how accurately junctions and other infrastructure are modelled. The nodes of the macroscopic topology are connected by *edges* that represent the tracks connecting two consecutive nodes. Normally station capacities and track topologies inside a station are not included in this macroscopic topology. They are considered in a consecutive, more detailed and computational step performed locally [22, 3]. This two-level approach has been successfully applied several times [18], [2], [14], [21]. In this paper concentrate on the first (macroscopic) part of this two-level approach.

Trains

Once the macroscopic topology has been defined it is possible to define the train services. For every train service we want to schedule, we have to fix a path through the macroscopic topology and define conditions on the corresponding train's driving behavior. A sequence of macroscopic nodes defines the route from the starting station to the final station. For every edge on this path we have to fix the track used by each train as well as a lower and upper time limit the train needs to move over the corresponding track. Similarly, it is also necessary to define a time interval specifying bounds for the dwell times at nodes where trains stop. In this case bounds are set to provide a minimal time for passengers to board and alight, and a maximum reasonable time this train can wait in the station. If a node just stands for a junction or a station where the train does not stop the lower and upper dwell time bounds can be set to zero. It is also possible to define turnaround time conditions for every train at its final station as well as time restrictions on the minimal and maximal total driving time over a longer path through the network. To refer to a train service, we use a number $t_l \in \mathbb{Z}$ called train service number.

Simplified Safety System

In addition to defining the topology and the train services, it is necessary to define a simplified version of the safety system in the model to prevent train overtakings and crossings. This is done by introducing headway constraints between every pair of train using the same track. These headway constraints can be train and track specific.

Additional Functional Requirements

In addition to the macroscopic infrastructure constraints, there are some further options to describe functional requirements for the set of train services being modeled. For example, it is possible to specify lower and upper bounds for connections between two trains in a station to achieve travel time targets for passengers making connections between the trains. It is also possible to create a passenger friendly distribution of trains serving the same route using frequency constraints, and it is of course possible to constrain any train departure and arrival time to a desired time interval.

The Period Length

Before starting to construct a periodic timetable, it is necessary to fix the period length, which is denoted T. Often the period is 60 minutes ($T = 60$) or 3600 seconds ($T = 3600$), if planning must be done with a finer time discretization. It is also possible to choose a time period smaller or larger than one hour. The important characteristic is that a constructed timetable can be repeated without conflict between two periods. These criteria must be part of the PESP model.

Formulating the PESP Model

Once the data has been defined it is possible to formulate a mathematical model to solve a macroscopic periodic timetabling problem satisfying all the infrastructure restrictions and functional requirements. The aim of this timetabling problem is to find feasible departure and arrival times for every train at every node of its route through the macroscopic infrastructure. These departure and arrival times are called the *events* of the periodic event scheduling problem and constitute the set V.

More formally, a macroscopic timetable is a function $\pi : V \rightarrow [0,T)$ uniquely assigning a time $\pi(v_i) =: \pi_i \in [0,T)$ to every event $v_i \in V$ in our model. Thus, for every departure and arrival event v_i we introduce a variable π_i which can be considered the decision variable for the problem of identifying the departure and arrival times being sought in the model.

Each condition on the desired timetable mentioned in Table 1 can be described as a constraint between two event times π_i and π_j as a minimal and maximal time which has to pass between the two corresponding events v_i and v_j. We thus introduce a PESP constraint between two event times π_i and π_j as a constraint $a_{ij} \in A$ with three parameters: $a_{ij} = (l_{ij}, u_{ij}, type_{ij})$, where $l_{ij} \leq u_{ij}$ describe the lower and upper bound for the difference between the two event times π_i and π_j and $type_{ij} \in \{cTrip, cDwell, cHead, cFreq, cConn, cSlot, cDep\}$ describes the type of constraint explained in Table 1. We distinguish between these different types of constraints to have a better overview for modeling. From a mathematical point of view they do not differ. A constraint a_{ij} is satisfied if and only if

$$l_{ij} \leq (\pi_j - \pi_i) \mod T \leq u_{ij}, \qquad (1)$$

where the modulo operator is important to express the periodicity requirement.

TABLE 1: Summary of the Different Constraint Types

cTrip	For every train, constraints of type *cTrip* are given between every departure event and the subsequent arrival event modeling the minimal and maximal driving time this train needs for the corresponding distance.
cDwell	Every trip constraint, except the last one to the final station, is followed by a dwell constraint between the arrival and departure event of a train in the same station (node) specifying bounds for the dwell time at this node.
cHead	With constraints of type *cHead*, a simplified safety system is modeled. For every pair of trains using the same track, a headway constraint is introduced which ensures that the time passing between the corresponding arrival and departure at the beginning and the end of this track are separated by at least the headway time h. Since the order of trains is normally not determined at this point, a constraint with lower bound h and upper bound $T-h$ is used for every pair of train using the same track.
cFreq	If there are trains with a smaller period $\frac{T}{F}, F > 1, F \in \mathbb{N}$ than T, this train has to be modeled F times and frequency constraints fixing an exact distance between every departure event of these trains at each station have to be added.
cConn	To model a desired connection, a constraint between the arrival event of the first train and the departure event of the second train in the corresponding station requiring the allowed transfer time is added.
cSlot	To fix a certain time event in a desired interval, an additional node, called "zero vertex", is introduced. The zero event will take place at the time 0 independently of every train, so constraints between this zero vertex and the desired event are added to constraint its time.
cDep	To separate two trains serving a similar offer by at least l_{ij} time units and at most u_{ij} time units, constraints of type *cDep* are used.
cOverall	Constraints of type *cOverall* are used to fix minimal and maximal total travel times for a train from its first station to the end destination. This can be used to set required buffer times.

A PESP model can be visualized as a directed graph (Figure 1). The top part of the figure shows four train pairs running between Luzern and Ebikon during the specified period (one suburban train, two regional express trains and one intercity train). The middle part of the diagram shows the simplified topography and the bottom part shows how the schedule and topography is represented in the PESP model.

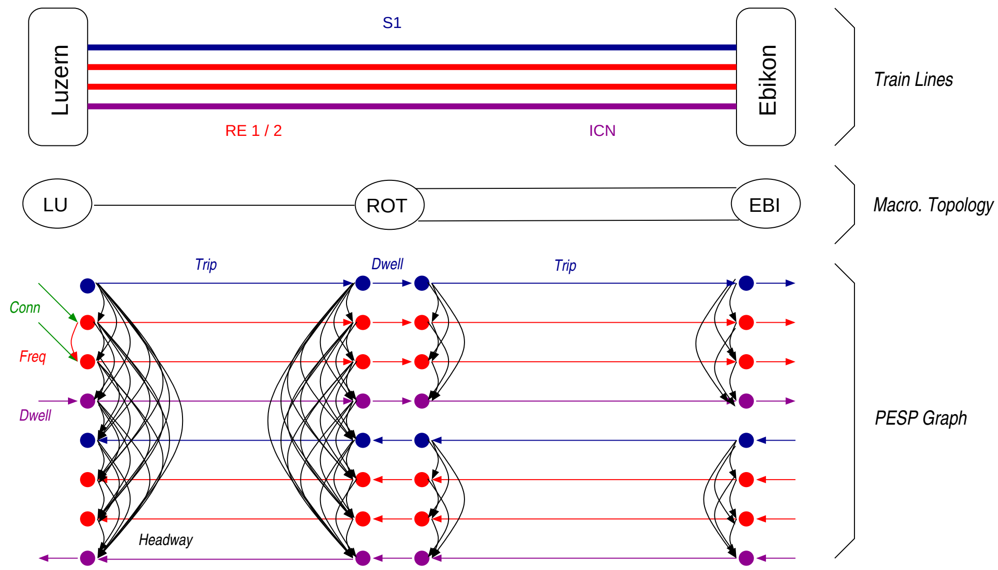

FIGURE 1: Example of a PESP graph for the route Luzern to Ebikon with a change from a single track line to a double track line in Rotsee and four trains running in both directions.

Every event $v_i \in V$ defines a vertex and every constraint $a_{ij} \in A$ a directed edge from vertex v_i to vertex v_j. To every edge a_{ij} we assign a closed interval describing the time which should pass between the time of event v_i and the time of event v_j.

3 SOLUTION METHODS

The PESP is proven to be a very hard problem (NP-hard, [20]). This means that in the worst case, computation times grow exponentially with the problem size (unless $P = NP$,[5]). Nevertheless, the development of algorithms to solve the PESP has been progressing for several years.

There are two main methods for solving a PESP model: constraint programming and mixed-integer programming [13]. The first method, constraint programming, together with the PESP model is core element of the Dutch railway schedule planning software DONS (Design of Networks Schedules) [9] and is also used for TAKT, a German program system [15]. The second method, mixed-integer programming, has been used in practice for smaller projects, for example a new timetable construction for the underground metro train system of Berlin [11] but has not yet been used on a large scale timetable planning project.

In the mixed-integer programming approach, integer and continuous variables are used to describe the PESP model in a system of linear inequalities, which is completed by an objective function for optimization. These models can then be solved by branch-and-cut methods [1] implemented in state-of-the-art commercial solvers. Using continuous time variables, direct optimization capability and the possibility of adding further constraints not directly modeled in the PESP graph this solution method has a number of advantages over constraint programming and is therefore used in this research.

The straightforward mixed-integer linear programming formulation, called “Classical MILP formulation” uses two types of decision variables: continuous variables π_i for every event time and integer variables q_j for every constraint $a_j \in A$ to model the periodicity. Every departure or arrival time π_i of the desired periodic timetable must have a value between 0 and the total time period T $(0 \leq \pi_i < T, \forall v_i \in V)$. Moreover, these event times have to satisfy every condition of the PESP model, which we can formulate directly as in the PESP model. The only difference to equation (1) is the replacement of the modulo operator $\mod T$ by $p_a \cdot T$, allowing addition of an arbitrary integer multiple of the time period T to ensure periodicity.

Several possible objective functions for this problem have already been examined including the minimization of total travel time, maximizing a specific timetable robustness measure, or minimizing rolling stock [2]. As described in Section 5 this research used minimization of total travel time as an objective function, summing up the activity durations associated with each trip, dwell and connection constraint. The classical MILP can therefore be stated as follows:

$$
\begin{array}{lll}
\text{Minimize} & f_{obj}(\pi) & \\
\text{subject to} & l_a \leq \pi_i - \pi_j + p_a T \leq u_a & \forall a \in A \\
& 0 \leq \pi_i < T & \forall v_i \in V \\
& p_a \in \mathbb{Z} & \forall a \in A
\end{array}
$$

Besides the classical MILP formulation, which directly follows from constraints of the PESP model, there exists a second formulation called cyclic MILP formulation [16, 10]. The cyclic formulation turned out to be more efficient for several test cases [10].

Instead of directly considering all time events π_i as decision variables, time differences between every pair of event times connected over a PESP constraint are used. These new variables $x_a := \pi_i - \pi_j$ for every PESP constraint $a_{ij} \in A$ are called tension variables in analogy to electrical networks. To satisfy every PESP constraint, each tension variable x_a has to lie between its corresponding edge bounds $(l_a \leq x_a \leq u_a)$. In addition to these continuous tension variables, integer variables are again necessary to require a periodic timetable. Similar to an electric circuit, the directed sum over all tension variables along a cycle in the PESP graph, modulo the total time, has to be zero. Thus, for every cycle C in the PESP graph, $\sum_{a\in C^+} x_a - \sum_{a\in C^-} x_a = Tq_C$, where C^+ and C^- denote the set of every positively oriented and negatively oriented edge along cycle C, and q_C is an integer variable called cycle periodicity variable for cycle C.

$$
\begin{array}{lll}
\text{Minimize} & f_{obj}(x) & \\
\text{subject to} & l_a \leq x_a \leq u_a & \forall a \in A \\
& \displaystyle\sum_{a\in C^+} x_a - \sum_{a\in C^-} x_a = Tq_C & \forall C \in \mathscr{C}_B \\
& x_a \in \mathbb{R}^+ & \forall a \in A \\
& q_C \in \mathbb{Z} & \forall C \in \mathscr{C}_B
\end{array}
$$

Since there are exponentially many cycles in a general graph, it would be necessary to introduce the same number of additional constraints and integer variables. However, it has been shown that it is sufficient to require this cycle constraint for every cycle out of an integer cycle bases $\mathscr{C}_B$ of the PESP graph (a much smaller subset than the set of every cycle) [12] to ensure that every cycle satisfies the cycle constraint. For a general graph there exist a huge set of possible integer cycle bases. Fortunately, they do not influence the result of our timetable construction, but they have a significant impact on computation time. The choice of a best cycle basis to this problem is still an unresolved research question. Empirically it has been shown that an integral cycle basis constructed out of a minimum spanning tree (with respect to the edge spans as weight) leads to good performance and therefore this approach is also used in this research. The objective function can be translated from the classical MILP formulation. To simplify notation it will from here on be referred to as $PESP(V,A,\mathscr{C}_B, f_{obj}(x))$.

4 HIERARCHICAL DECOMPOSITION METHOD

Motivated by planning practice and manual timetable construction, our goal is to study the influence of hierarchical train prioritization in timetable construction via the periodic event scheduling problem. We define priority classes by dividing the whole set of train services (trips) into $p > 1$ disjoint groups. In every iteration, one group of train services is added to the scheduling problem while constraining previously scheduled train runs to a certain time interval of size t_w around their scheduled time from the previous iteration. Different variations of train service partitions and interval sizes t_w were tested to reveal dependencies between these two parameters, computation time, and the quality of the obtained timetables. This section defines the hierarchical decomposition method in more detail and specifies the algorithm. The results will be analyzed in Section 5.

Let T_l be the set of all train service numbers contained in a considered PESP model with event set V and constraint set A. To partition T_l into p different groups for prioritization, we define a function $prio : T_l \to \{1,\dots,p\}$ assigning a priority value $prio(t_l)$ to every train service number $t_l \in T_l$.

In the first iteration, only train services of priority 1, i.e., the set $prio^{-1}(1)$, are scheduled. Since a reduced set of trains is being considered only the events and constraints corresponding to these trains need to be considered. Let V_1 be this reduced set of events, containing all events corresponding to train services of $prio^{-1}(1)$, plus the zero time event. The reduced constraint set A_1 then is defined as the set of constraints contained in A whose end nodes both belong to V_1 (edges of the induced subgraph over V_1).

To compute a first partial timetable, it is necessary to solve $PESP(V_1,A_1,\mathscr{C}_B|_{A_1}, f_{obj}(x)|_{A_1})$, where C_B and $f_{obj}(x)$ are the standard cycle basis, and the objective function is minimization of total travel time (as specified in the previous section), reduced to the smaller domain of tension variables corresponding to A_1.

Let $\Pi^{(1)}$ be the set of all event times $\pi_k^{(1)}$ in the solution of this first reduced problem. To fix this result within a time interval of size t_w around this solution, new slot constraints are introduced for every time event $v_k \in V_1$ as

$$slot_k^{(1)} : a_{0k}^{(1)} = (\pi_k^{(1)} - \frac{t_w}{2}, \pi_k^{(1)} + \frac{t_w}{2}, slot)$$

and all slot constraints are collected in $A^{(1)}_{fix}$. If the first partial timetable problem has no solution, the entire timetabling problem is infeasible.

In every further iteration i, $1 < i \leq p$, the previous event set V_{i-1} is enlarged by adding all events corresponding to train services of the ith priority group $prio^{-1}(i)$ and define A_i again by the set of constraints of A with both end nodes in V_i. We add the slot constraints $A^{(i-1)}_{fix}$ to the current constraints and solve $PESP(V_i, A_i \cup A^{(i-1)}_{fix}, \mathscr{C}_B|_{A_i \cup A^{(i-1)}_{fix}}, f_{obj}(x)|_{A_i})$. As in the first iteration, after computing a partial timetable, a new set of slot constraints for every time event $v_k \in V_i$, denoted by $A^{(i)}_{fix}$, is constructed in every further iteration. If a partial $PESP$ is infeasible, we stop the iteration. In Figure 2 a flow chart diagram summarizes this steps for one fixed time interval t_w.

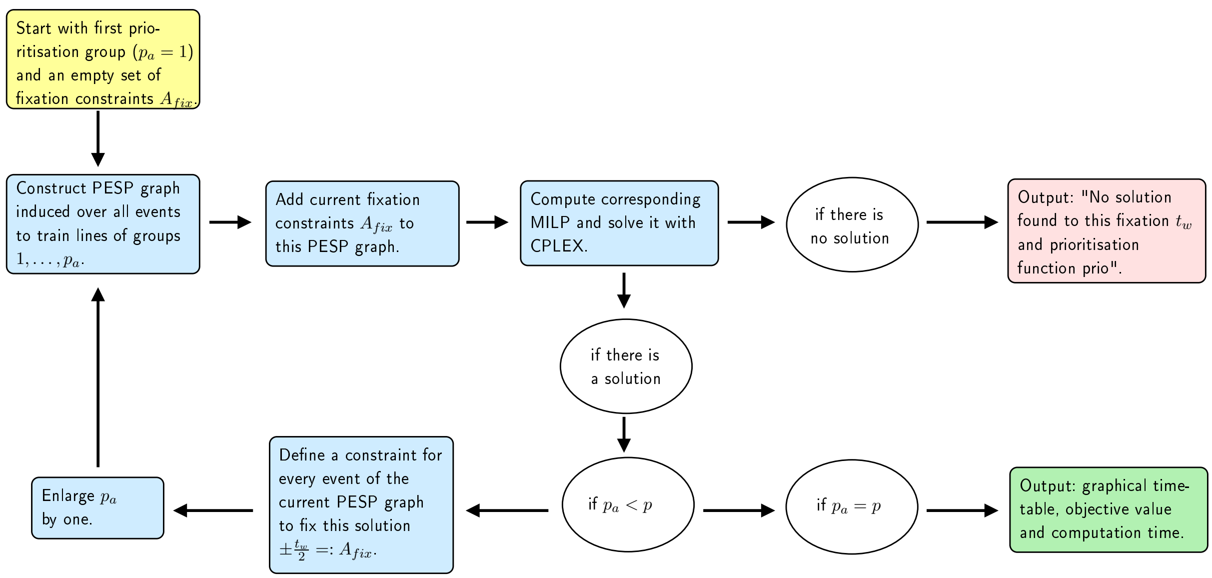

FIGURE 2: Flow chart for one hierarchical iteration for a fixed parameter t_w and a given set of priorization groups.

The main control parameter of the algorithm is the interval length t_w, which controls the degree of freedom of higher priority trains when scheduling the lower priority trains. The parameter can vary from a complete fixation ($t_w = 0$) to complete freedom ($t_w = T$). In the latter case, the hierarchical decomposition problem equals the original complete PESP problem.

Enlarging parameter t_w starting from zero up to $tw = T$, we therefore obtain a transition from a strictly sequential planning scheme as one extreme to a fully simultaneous planning as in the standard PESP. Selecting a suitable t_w value is crucial for the algorithm running time and the solution quality. In particular, a too small t_w might create infeasibilities during the iterations even if the original simultaneous PESP is feasible. Thus, in case of infeasiblity, a simple remedy would be to restart the algorithm with a larger t_w.

5 COMPUTATIONAL RESULTS

The decomposition method described above was tested using a timetabling problem from central Switzerland containing the three cities Luzern, Zug, and Arth-Goldau, reaching up to the main corridor Olten-Lenzburg in midland and to lake of Zurich containing Pfäffikon SZ and Thalwil.

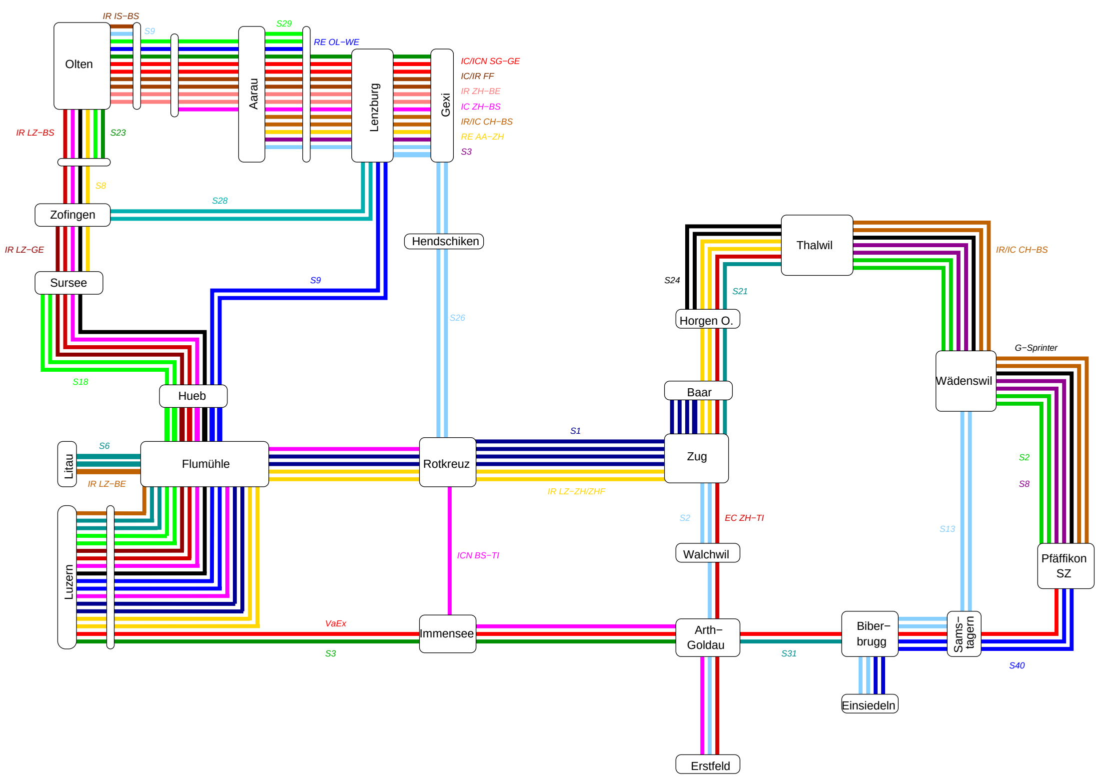

FIGURE 3: Line map of the model

The data used to build this model were taken from the timetabling software used at the Swiss Federal Railways (SBB). Out of over 7500 passenger trains running every day on Switzerland's railway network, all periodic train services using the infrastructure of the case study region were extracted. Driving paths, minimal trip and dwell times as well as all commercial stops are read from the data set. Fixing of upper bounds for the dwell times is a crucial point to allow train crossings and overtakings. In our model crossings are allowed and automatically determined by the algorithm in all stations where crossings are allowed in today's timetable. Signal headways are directly extracted from the data for all distances and train types. They range from 1.75 minutes for very frequently used infrastructure elements up to 6 minutes for sparsely used tracks. Altogether the case study region contains 159 stations and 113 periodic trains per hour. Out of these 159 stations 68 had to be included in the macroscopic topology because of varying track numbers and signal headways, as well as crossing points and ending train services.

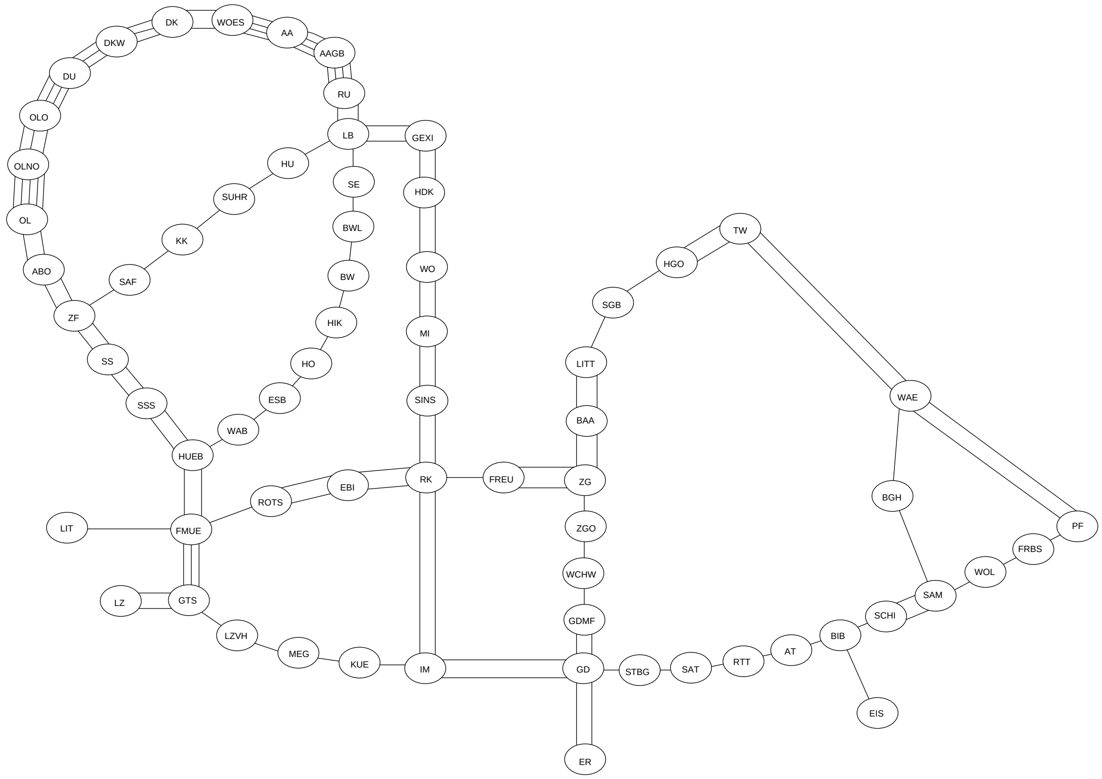

FIGURE 4: Macroscopic topology of the model

The resulting PESP model contains 1483 events and 6214 constraints. There are 23 connection constraints modeling high priority connections between trains of the chosen model. Furthermore, functional requirements such as time distances between trains with similar commercial offers are automatically taken from today's timetable and are included in the model via 108 frequency and separation constraints. Using overall constraints, a minimal buffer time of 7 percent is required for all train trips. The objective function minimizes the total travel time (sum of all used trip, dwell and connection times) of all trains passing the considered region and has its optimum at 3109 minutes.

The remainder of this section describes the computational results for the example and discusses in particular the influence of the two decomposition parameters *prio* and t_w. All computations were performed using a compute server with 2x2 Intel X5650 CPUs each with six cores and the MIP solver of IBM ILOG CPLEX* Optimizer (Version 12.4) with an optimality gap of 0.5%. The evaluation tested every even value for the parameter t_w (train fixation interval), starting from a complete fixation ($t_w = 0$) up to a weaker fixation of +/- 16 minutes ($t_w = 32$). The allocation of prioritization groups for the hierarchical decomposition is illustrated in Figure 5. As a first priority group all fast trains are taken. Then all regional passenger trains are automatically distributed in a predefined number of groups corresponding their geographical position in the model. The number of groups (p) varies from three groups up to seven and is part of our analysis regarding computation time and quality.

The hierarchical iteration as described in Section 4 and illustrated in Figure 2 is done for every even fixation value from $t_w = 0$ to $t_w = 32$ and for every groups size $p = 3$ to $p = 7$. Figure 6 gives an overview on the results. The first graphic of Figure 6 shows dependencies of the size of fixation interval t_w and computation time. As expected, a very strong sequential planning (small t_w) has much faster computation times than more synchronous plannings with interval sizes of half an hour. The computation time does not grow monotonously since enlarging the fixation interval allows the model to find new timetable variants from one group to the next, which in some cases can accelerate the solution processes. A more fascinating result is the influence of the number of prioritization groups. Although there are more MIPs to solve for a larger number of groups, the solution process is faster for $p = 5, 6, 7$ than for $p = 3, 4$.

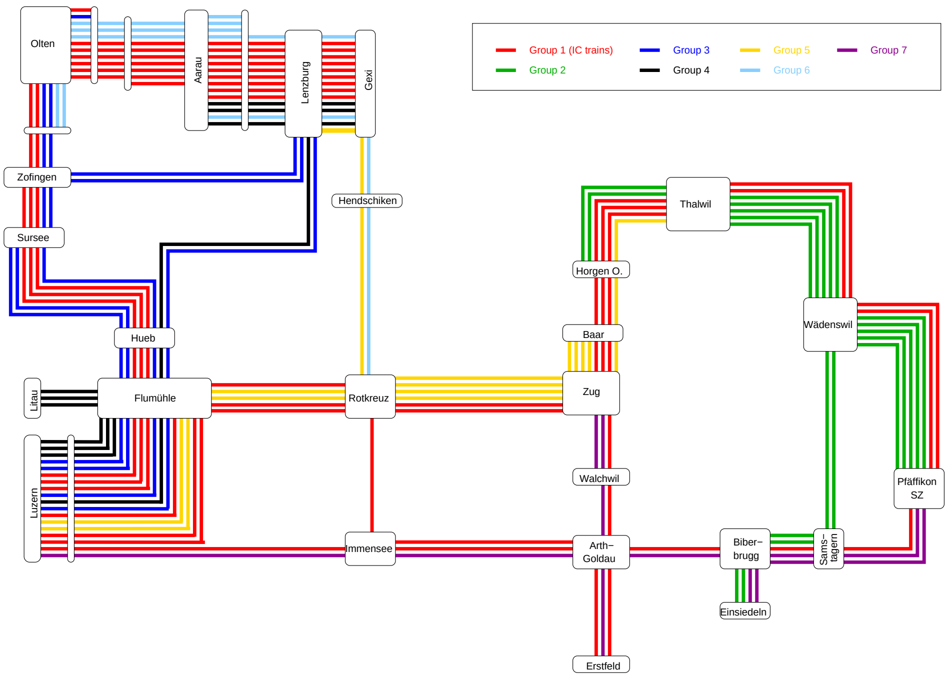

FIGURE 5: Example for the allocation of prioritization groups for $p = 7$.

Comparing these computation times of the first graphic with the corresponding objective values in the second graphic, the results show the problem of complete sequential planning. For very small fixation intervals t_w the algorithm ended in infeasiblity. It was not possible to add all prioritization groups without moving train departure and arrival times of already included trains. In the case of $p = 3$ a first feasible timetable could be found for a fixation interval of size $t_w = 4$ with total travel time (objective value) of 3168 minutes. Enlarging the number of prioritization groups the first occurrence of a feasible time table moves more and more to larger fixation intervals. Up from an interval size of 18 minutes also the last computation series found a feasible timetable. Observing the trend of objective values comparing to the fixation interval size (t_w) the solution quality of our timetables, measured with total travel time, grows for larger values of t_w. Interesting here are also the good results for $p = 6$ and $p = 7$ starting from the first occurrence of a feasible timetable. Up from a fixation interval of size 15 minutes, all objective values stay at a certain level of quality not exceeding 0.5 percent of the optimal total travel time.

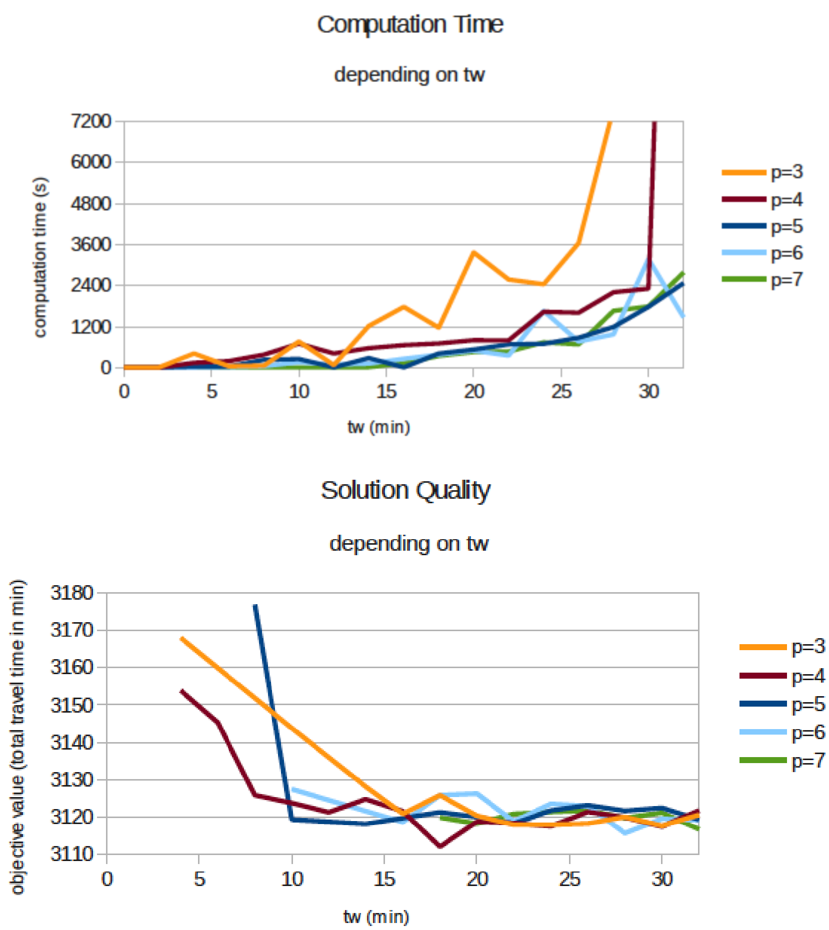

FIGURE 6: Computation times and objective values refining the prioritization function from $p = 3$ up to $p = 7$

For a better comparison of our new solution method for the PESP model to the complete synchronous solution process we defined a straightforward method stringing together all computations for one prioritization group starting with $t_w = 0$ and enlarging t_w until a solution satisfying our expectations was found. The result is shown in Figure 7 and shows clear advantages to the complete synchronous solution process. All hierarchical decomposition methods for all prioritization groups from $p = 3$ to $p = 7$ find a first feasible timetable in less than 10 minutes, while the original, completely synchronous method finds a first feasible solution only after 39 minutes. Furthermore, the optimization process was much faster in all hierarchical decomposition methods than in the complete synchronous method. After three hours computation time, the original method still hasn't reached a total travel time comparable to the hierarchical decomposition methods. Again, the prioritization groups for $p = 5, 6, 7$ show better results than using only three prioritization groups.

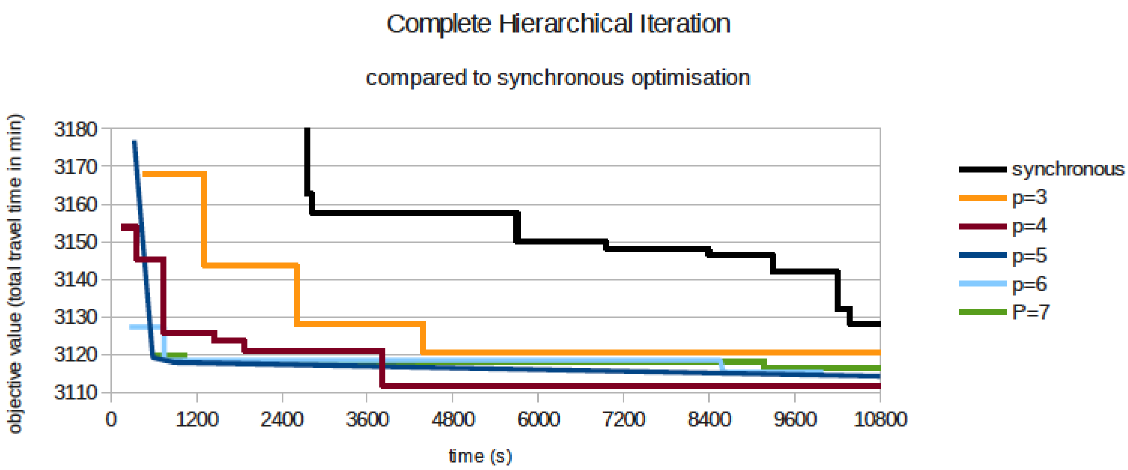

FIGURE 7: Comparison of the new hierarchical decomposition method to the original synchronous optimization process.

To find a good number and adequate choice of prioritization groups it is worth considering the different computation times for all MIP during one iteration. If there is one MIP having a remarkably higher computation time over different sizes of fixation intervals then the whole decomposition method can be improved by further partitioning the corresponding prioritization group. On the other hand, if there is one MIP, especially the first one, with a very short computation time it can be beneficial to add some further trains to this prioritization group. The first prioritization group often can include many more trains than the remaining groups.

6 CONCLUSION

This paper introduces a hierarchical decomposition method to solve instances of the periodic event scheduling problem (PESP) via its mixed integer linear programs (MILP) formulation. Train services are partitioned into different priority groups and introduced into the scheduling problem step by step, where solutions of previous steps are only fixed within a specified time interval. The introduction of these two parameters (number of prioritization groups and time interval) allows a continuous transition from fully sequential timetable construction up to a fully simultaneous planning of the whole problem (i.e. the original MILP solution approach). Therefore the discovery of a feasible timetable, if there one exists, is ensured.

The method was tested on a model describing a time tabling situation over 159 stations in central Switzerland and shows promising results. Using the hierarchical decomposition method as a heuristic method to solve the PESP can improve computation times while optaining optimal or close to optimal schedules. For all considered sizes of priority groups the new method found first feasible solutions faster and showed a faster optimization process than the corresponding complete synchronous solution process.

The results of this paper also show that strong sequential timetable construction, without backward iterations, can quickly end with infeasible subproblems, as soon as there is no large capacity surplus. Using an intermediate time fixation interval size of about 10 to 30 minutes seems a suitable and robust choice, and the resulting computation times and solution quality are remarkably good.

Decomposition in general plays an important role in improving algorithms for timetabling. Interesting areas for future research include further examinations of the proposed train hierarchical decomposition methods for national problem instances. Another is to consider more deeply the relationship between its parameter settings and the corresponding mixed-integer linear program.

* IBM, ILOG, and CPLEX are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies.

7 REFERENCES

[1] D. Bertsimas and Weismantel R. *Optimization over integers*. Athena Scientific, 2005.

[2] G. Caimi. *Algorithmic decision support for train scheduling in a large railway network*. PhD thesis, ETH Zurich, 2009.

[3] G. Caimi, F. Chudak, M. Fuchsberger, M. Laumanns, and R. Zenklusen. A new resource-constrained multi-commodity flow model for conflict-free train routing and scheduling. *Transportation Science*, 2010. Accepted for publication.

[4] G. Caimi, M. Laumanns, K. Schüpbach, S. Wörner, and M. Fuchsberger. The periodic service intention as a conceptual framework for generating timetables with partial periodicity. *Transportation Planning and Technology*, 34(4):323–339, 2011.

[5] St. Cook and P. Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.

[6] P. Hachemane. *Evaluation de la capa-cit de rseaux ferroviaires*. PhD thesis, Ecole polytechnique fdrale de Lausanne, Switzerland, 1997.

[7] D. Hauptmann. *Automatische und diskriminierungsfreie Ermittlung von Fahrplantrassen in beliebig grossen Netzen spurgeführter Verkehrssysteme*. PhD thesis, IVE, Universität Hannover, 2000. In German.

[8] G. Hürlimann. *Die Eisenbahn der Zukunft - Automatisierung, Schnellverkehr und Modernisierung bei den SBB 1955 bis 2005*. PhD thesis, Universität Zürich, 2006.

[9] L. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maroti, A. Schrijver, A. Steenbeek, and R. Ybema. The New Dutch Timetable: The OR Revolution. *INTERFACES*, 39(1):6–17, 2009.

[10] C. Liebchen. *Periodic Timetable Optimization in Public Transport*. PhD thesis, TU Berlin, 2006.

[11] C. Liebchen. The first optimized railway timetable in practice. *Transportation Science*, 42(4):420–435, 2008.

[12] C. Liebchen and L. Peeters. Integral cycle bases for cyclic timetabling. *Discrete Optimization*, 6(1):98–109, 2009.

[13] T. Lindner. *Train Schedule Optimization in Public Rail Transport*. PhD thesis, TU Braunschweig, June 2000.

[14] M.A. Odijk and I.H.A Van den Berg. DONS: Computer aided design of regular service timetables. In T.K.S. Murthy, C.A. Brebbia, B. Mellitt, G. Sciutto, and S. Sone, editors, *Computers in Railways IV*, volume 2, pages 109–116, 1994.

[15] J. Opitz. *Automatische Erzeugung und Optimierung von Taktfahrplänen in Schienenverkehrsnetzen*. PhD thesis, TU Dresden, 2009.

[16] L. Peeters and L. Kroon. A cycle based optimization model for the cyclic railway timetabling problem. In S. Voß and J.R. Daduna, editors, *Proceedings Computer-Aided Scheduling of Public Transport (CASPT 2000)*, volume 505, pages 275–296. Springer, Berlin, 2001.

[17] T. Schlechte. Railway track allocation - simulation and optimization. Technical report, Konrad-Zuse-Zentrum für Informatikdienste Berlin, 2011.

[18] T. Schlechte, R. Berndörfer, B. Erol, T. Graffagnino, and E. Swarat. Micro-macro transformation of railway networks. *Journal of Rail Transport Planning & Management*, 1:38–48, 2011.

[19] A. Schrijver and A. Steenbeek. Dienstregelingontwikkeling voor Railned: Rapport CADANS 1.0. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam, the Netherlands, 1994. In Dutch.

[20] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM J. Disc. Math.*, 2(4):550–581, 1989.

[21] B. Sewcyk, A. Radtke, and G. Wilfinger. Combining Microscopic and Macroscopic Infrastructure Planning Models. In *Proceedings of the 2nd International Seminar on Railway Operations Modeling and Analysis*. TU Delft, Leibniz Universität Hannover, TU Braunschweig, 2007.

[22] P. J. Zwaneveld. *Railway Planning – Routing of trains and allocation of passenger lines*. PhD thesis, Erasmus University Rotterdam, 1997.