Netw Spat Econ (2009) 9:25–46
DOI 10.1007/s11067-008-9091-6

Design of a Railway Scheduling Model for Dense Services

Gabrio Caimi · Dan Burkolter · Thomas Herrmann · Fabian Chudak · Marco Laumanns

Published online: 23 November 2008

Abstract We address the problem of generating detailed conflict-free railway schedules for given sets of train lines and frequencies. To solve this problem for large railway networks, we propose a network decomposition into condensation and compensation zones. Condensation zones contain main station areas, where capacity is limited and trains are required to travel with maximum speed. They are connected by compensation zones, where traffic is less dense and time reserves can be introduced for increasing stability. In this paper, we focus on the scheduling problem in condensation zones. To gain structure in the schedule we enforce a time discretisation which reduces the problem size considerably and also the cognitive load of the dispatchers. The problem is formulated as an independent set problem in a conflict graph, which is then solved using a fixed-point iteration heuristic. Results show that even large-scale problems with dense timetables and large topologies can be solved quickly.

G. Caimi (✉) · M. Laumanns
Institute for Operations Research, ETH Zurich, 8092 Zürich, Switzerland
e-mail: caimig@ifor.math.ethz.ch

M. Laumanns
e-mail: laumanns@ifor.math.ethz.ch

D. Burkolter
SMA und Partner AG, 8050 Zürich, Switzerland
e-mail: d.burkolter@sma-partner.ch

T. Herrmann
R+R Burger und Partner AG, 5401 Baden, Switzerland
e-mail: herrmann@rrag.ch

F. Chudak
D-Wave Systems Inc., V5C 6G9 Burnaby, British Columbia, Canada
e-mail: fchudak@dwavesys.com

Keywords Timetable · Railway network decomposition · Conflict-free scheduling · Time discretisation · Itinerary reduction · Independent set problem · Fixed point iteration heuristic

1 Introduction

Railway traffic in Switzerland, as well as in many other countries, has increased considerably for both passenger and freight transportation during the last few years, and this trend is expected to continue. Construction of new tracks, though, is very expensive and hardly possible in many city centers. The capacity of the existing network must therefore be better utilised to meet the customer demand for an enlarged offer.

When increasing the density of the timetable, scheduling trains becomes more and more difficult as the chosen schedule not only has to meet safety restrictions, but should also minimise propagation of delays. An automatic generation of conflict-free timetables in reasonable time can be very helpful in order to evaluate several alternative timetables. Therefore, the interest in automatically generating railway timetables has increased over the past years. In particular, the Swiss Federal Railways (SBB), major operator of railway infrastructure in Switzerland, is currently investing efforts into the development of efficient methods for generating and operating railway schedules (Laube et al. 2007; Lüthi et al. 2007).

Usually, the strategic timetable generation is done in two steps:

(i) In the first step, with the help of origin-destination matrices or other evaluations of the passenger demand, an offer of train services with lines and frequencies is developed to meet the customer needs. We call this offer *train service intention*, since at this point it is not known whether this offer is realisable. A train service intention consists of train lines and frequencies, specifying the customer-relevant information such as stop stations, interconnection possibilities, and rolling stock. For detailed information we refer to Laube and Mahadevan (2008).

(ii) In a second step, the feasibility of a service intention is checked by trying to generate a feasible schedule. If this is possible, a schedule is provided as proof of feasibility, otherwise both steps have to be reiterated until a feasible service intention is found.

Our research focuses on the second step, the construction of a timetable for a given train service intention. We concentrate on the creation of detailed train schedules in which both, an itinerary through the railway topology and passing times, have to be determined for each train. In this way we can guarantee that the provided timetable runs conflict-free, i.e., assuming no delays, all trains can run exactly as planned without creating safety conflicts, and no rescheduling due to resource conflicts becomes necessary. The creation of detailed train schedules is relevant for both strategic and tactical timetable generation. It guarantees feasibility of the corresponding service intention in the long-term case, whereas in the short term it enables the operation of a conflict-free timetable.

The problem of finding detailed train schedules for each train is accentuated in major stations with many incoming an outgoing lines, where connections with short transfer times must be provided. As a consequence, trains tend to arrive and leave during a short peak interval and the solution space of feasible routing assignments is more constrained. In contrast, there are less parallel tracks and much less switches in rural regions, resulting in a considerably smaller number of potential itineraries. Due to the lower traffic density in rural regions, time reserves can be introduced. We therefore propose a decomposition of the whole network into *condensation zones* and *compensation zones*, which can be treated with different models and algorithms according to their distinct properties. The decomposition and the approach to coordinate the different zones are introduced in Section 2. We then focus on the problem of scheduling trains in condensation zones, which are identified as the critical zones in the network. For a given train service intention and an arbitrary subset of boundary conditions, we present a generic model and an algorithm to create conflict-free train schedules in Section 3. Section 4 presents results for the test cases of Berne and Lucerne in Switzerland, and Section 5 concludes with a summary and outlook for further research.

Related work (see Huisman et al. 2005 for a survey) reports two principal approaches to the problem of finding a schedule for a whole railway network: one abstracting from the detailed track topology and the other considering partial detailed topologies of the network. Of particular interest in the first case is the Periodic Event Scheduling Problem (PESP), in which a set of cyclic events is modelled via cyclic time window constraints (Serafini and Ukovich 1989). The PESP allows to model large railway networks in an aggregated way to produce draft timetables. Often these draft timetables only include arrival and departure times at major stations on the scale of minutes. PESP enables to schedule trains in a relatively large railway network (such as the Netherlands, see Odijk 1996), but the exact train routing on an aggregated level has to be known a priori and the safety system is only roughly modeled using headway times. However, PESP solutions do not guarantee timetable feasibility on a detailed level. In particular, PESP, as well as other approaches (e.g., Carey 1994), assumes infinite capacity in station regions.

In the second approach, the detailed topology of a local region, typically a main station area, is taken into consideration for producing conflict-free timetables (Bourachot 1986; Carey and Carville 2003) or checking the feasibility of a given macroscopic timetable (Caimi et al. 2005; Zwaneveld et al. 1996).

Effort to integrate these approaches for a conflict-free scheduling of a whole railway network have remained rare. To our knowledge, the only project that addressed this question is the Dutch project DONS in collaboration with the Dutch Railways. Kroon and Zwaneveld (1995) and Schrijver and Steenbeck (1994) presented a two-level approach for a decision support system to create conflict-free timetables for the Dutch Railways. In the upper level, the train service intention is known, as well as an aggregated railway topology. The module CADANS supports the generation of cyclic hourly draft timetables using the PESP model. In the lower level, the timetable generated by CADANS is checked only by considering the running times over the relevant track sections and the release time, but the blocking times are not checked to be conflict-free. This module is called STATIONS and its model and algorithms are presented in Zwaneveld et al. (1996, 2001). The approach seems to be very interesting for our aims, but it does not take into consideration the different properties of condensation and compensation zones, whose distinct characteristics might be exploited more effectively with disparate scheduling policies. Moreover, additional optimisation potential exists in the interface between the zones, in particular for utilising the compensation zones for buffering against delays.

2 Network decomposition approach

The railway network is built to enable mobility of the population and depends on population density and geographical properties. Within conurbations, both the topology of the railway network and the layout of train lines are typically complex. This often requires many switches or the construction of level-free crossings around main stations to enable connections among all directions. In contrast, railway networks in rural regions usually consist only of singular lines that connect cities with a limited number of parallel track sections (1 or 2) and only few switches.

Furthermore, train traffic on a railroad network is typically not homogenous. In urban regions there are various commuter trains serving all parts of the conurbation with high frequency, long distance trains, and also some freight trains. The majority of these train services travel to the main station and cause heavy traffic in their proximity, additionally aggravated by the numerous intersections to serve all the possible directions. On the other hand, in lightly populated areas the few lines are served by local trains with low frequencies, passed by long distance trains, and also used by freight trains.

Consequently, we distinguish two major segments within the railway topology: the *condensation zones*, a relatively small area (radius up to 15 km) where the railway topology is quite complex and train frequencies are high, and the *compensation zones*, which have simple topologies with lower train frequencies. Figure 1 illustrates a possible subdivision into zones for the central part of the Swiss railway network.

2.1 Scheduling by network separation

In order to account for different timetabling paradigms and to handle large scheduling problems, we propose a new scheduling design, which first decomposes the problem geographically into condensation and compensation zones. Different policies for generating train schedules are then applied to the two zones according to their properties. In particular, time reserves are moved

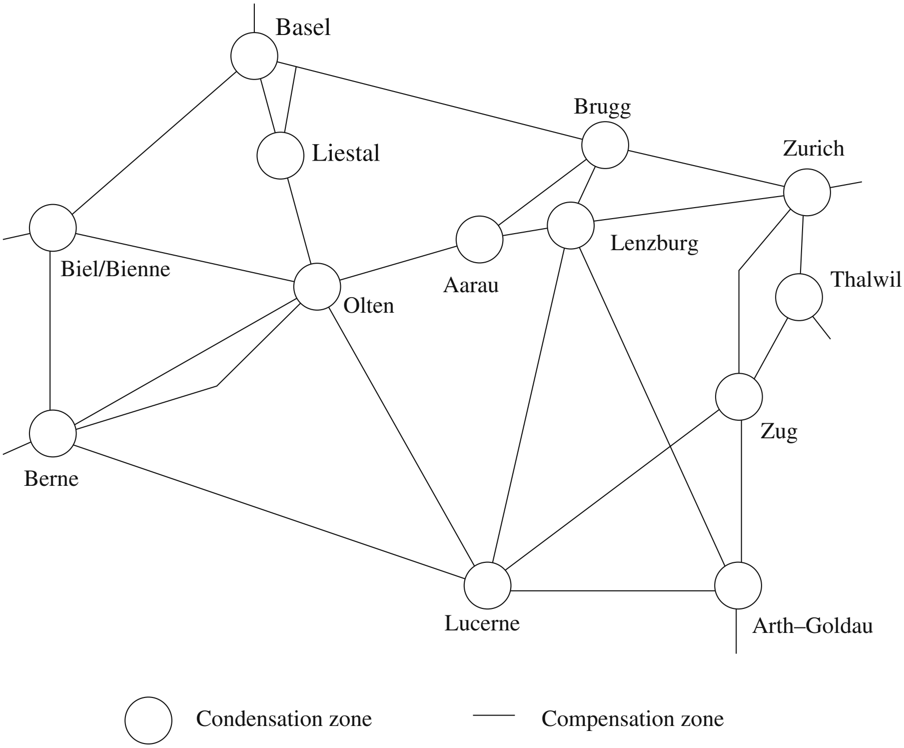


Fig. 1 Possible representation of the Swiss railway central part divided in condensation and compensation zones

from the condensation zones to the compensation zones, where more capacity is available.

The zones are connected by *portals* that comprise a certain number of parallel tracks. The portals are the interfaces between the zones, since travelling through portals is the only possibility to go from one zone to the next. Once the boundary conditions at these interface points are fixed, each zone can be treated independently without affecting other zones.

2.2 Condensation zones

In condensation zones the track topology is complex, and many different itineraries to travel between portals and platforms exist. Here, an appropriate assignment of exactly one of these itineraries to each train is crucial.

The typical layout of a main station region consists of stretches of several relatively long parallel tracks (typically 1–3 km) without switches, leading into different directions. These stretches are connected by switch regions which allow reaching all platforms and directions. The important fact is that the switch regions are short (usually up to 500 m) relative to the stretches and are,

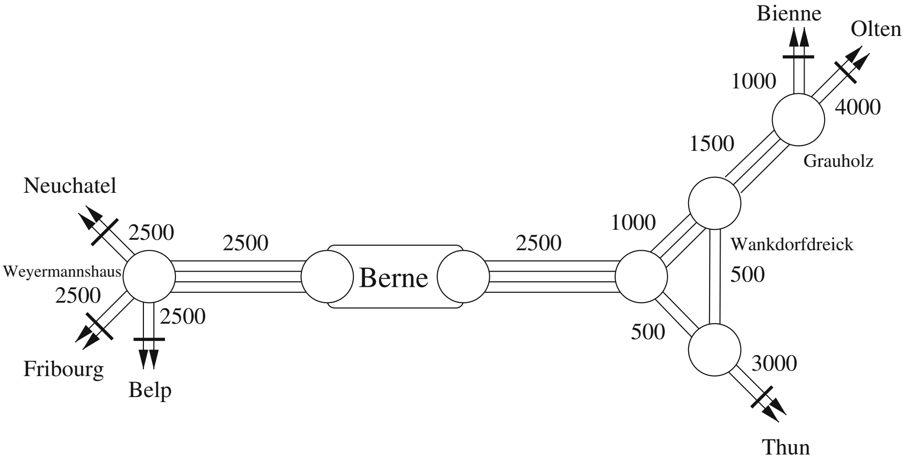
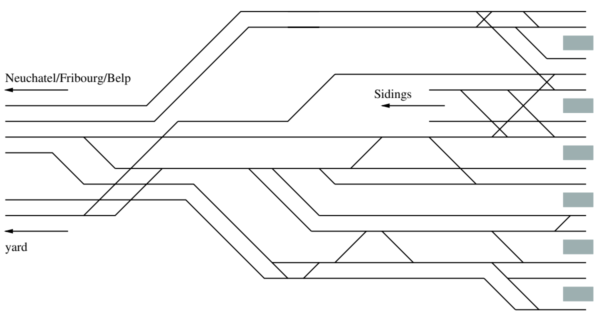


Fig. 2 Sketch of the condensation zone Berne. Approximate distances given in meters

therefore, passed quickly by trains. Figure 2 shows a rough layout of the track topology for the station region of Berne. The most complex switch regions lie east and west of the station just in front of the platforms. Their topologies are shown in the Figs. 3 and 4, respectively.

As such an area is expected to have a high traffic density, bottleneck resources should be made available as soon as they are not needed anymore. Thus, trains are required to travel through the condensation zone with full speed, i.e., no time reserves in the trip are included. In contrast to the compensation zones, the speed profile is therefore no longer free. It is thus sufficient to assign one passing time per train at a specific location (e.g., at the portal or at the platform). All the passing times within the condensation zone can then be derived, once the itinerary has been fixed. However, some reserves will remain integrated in the dwell and turn-around times at the platform of the stations. Additionally, there is a small tolerance (typically 15 s) in the speed profile to deal with imprecisions in travel behaviours.

Fig. 3 Switch region topology in the west of the Berne main station



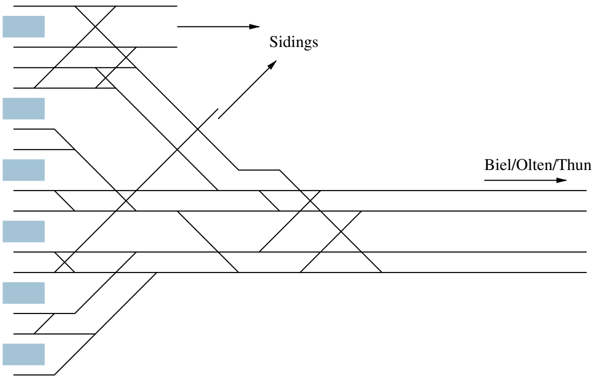


Fig. 4 Switch region topology in the east of the Berne main station

Under a denser service intention, the dispatchers will face more pressure and must respond to schedule disruptions faster than now. In Herrmann (2005) it was shown that for a possible condensed schedule, the dispatchers would have to react four times as often as nowadays to resolve disruptions. Thus, the adoption of a new scheduling paradigm for the condensation zones also contributes simplifying the dispatchers' work.

2.2.1 Time discretisation

In order to gain additional structure in the schedule we divide condensation zones into switch regions and stretches connecting two switch regions, as illustrated in Fig. 2. We then introduce a new policy in the switch regions based on a time discretisation. As a consequence, train entries into switch regions are restricted to certain time intervals. Trains must travel as quickly as possible through the switch regions in order to occupy the least capacity possible. Note that due to the time discretisation, trains cannot travel at their maximal technical speed. If necessary, their speed profiles are slightly adapted to accommodate the discretisation in a way that a minimal number of intervals are needed to pass the condensation zone. As a consequence, the duration of the run through the condensation zone is the same whichever route is chosen, since the difference in travel time (few seconds) are compensated by the time discretisation.

More formally, we introduce a time raster which describes the time intervals to enter the corresponding switch region.

Definition 1 (Time raster) Let K be a switch region and $T_K, \tau \in \mathbb{R}$ be real parameters with $0 \leq T_K < \tau$. A **time raster** for a switch region K is the partition of the time line in intervals of length τ with phase start in T_K. We call the interval

$$(T_K + (k-1) \cdot \tau, T_K + k \cdot \tau], \quad k \in \mathbb{N}$$

the k^{th} interval. We name τ the **interval length** of the raster and T_K the **phase** of the raster.

We measure the travel times inside a switch region according to this definition. If the travel time of a train z_i through K is $\Delta^K t_i$ seconds, we quantify the minimal number of intervals needed for travelling through the region K by

$$p_i^K(\tau) := \left\lceil \frac{\Delta^K t_i}{\tau} \right\rceil$$

where $\lceil x \rceil = \min\{n \in \mathbb{Z} \mid x \leq n\}$. The train run must use exactly $p_i^K(\tau)$ intervals, otherwise unnecessary loss of capacity is incurred. Notice that any path through the switch region will be allocated for the whole interval and may be therefore used at most once in each interval. Thus, we enforce utilisation of a minimal number of intervals by each train.

Definition 2 (Discretised timetable) Let τ be the interval length and $p_i := p_i^K(\tau)$ be the number of required intervals for train z_i to travel through the switch region K. A **discretised timetable** with interval length τ is a timetable for which a time raster for each switch region K with the following properties exists:

$$\forall \text{ trains } z_i \, \exists k_i \in \mathbb{N} : \quad \left[t_i^{in}, t_i^{out}\right] \in (T_K + (k_i - 1) \cdot \tau, \, T_K + (k_i - 1 + p_i) \cdot \tau],$$

where t_i^{in}, t_i^{out} are the entry and the exit time of train z_i in the switch region K.

The journey for train z_i inside the switch region begins at interval k_i and takes exactly p_i intervals. Figure 5 shows a discretised timetable both in the classical way and in a dedicated representation that shows the switch region

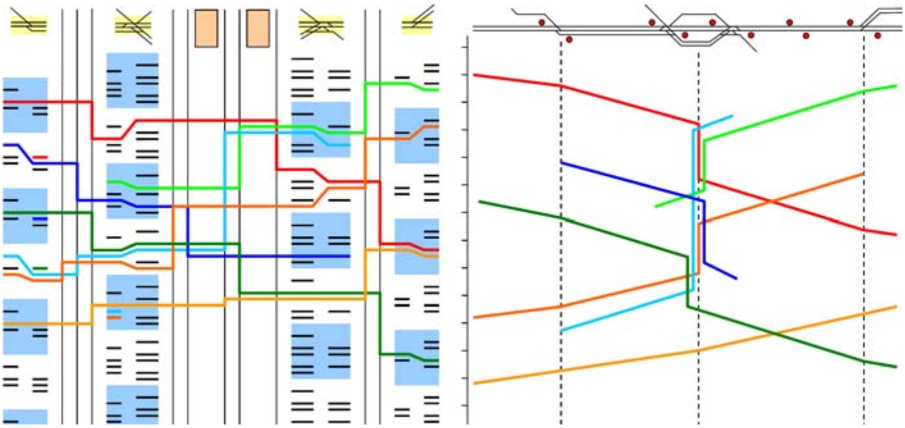


Fig. 5 Representation of a discretised timetable including track occupation (*left*) versus the classical time-space diagram (*right*). From Roos (2006)

and corresponding time raster, as well as the itinerary through the railway topology. In the figure, τ is chosen equal to the headway such that it is possible to use the same path in the switch region directly in the next time interval. Notice that in order to be able to shift train runs without having to change speed profiles, τ is the same for all switch regions in a condensation zone and each choice of $\tau > 0$ is possible. A discretised timetable with a very small τ (e.g., $\tau = 1$ s) essentially matches a standard schedule.

Detection of conflicts becomes simpler in a coarse-grained discretised schedule. The dispatchers gain better comprehensibility of the schedule and can therefore quickly decide in case of delays. However, an appropriate choice of T_K and τ is crucial to avoid too much capacity loss and thereby guarantee a good quality of the discretised timetables. The values of T_K can be chosen to minimise the time loss in the travel times on the stretches between two switch regions, whereas the choice of τ should achieve a compromise between large values to simplify the dispatcher's work and small values to minimise time loss.

Simplification of the dispatcher's work is a criterion that is very difficult to quantify. A potential indicator could be the number of distinct scheduling alternatives for each train during operations, which is approximately proportional to $1/\tau$. In this work, we test some different values for τ and the results shown in Section 4 suggest that a choice of τ equal to the headway time provides good performance. For T_K, values provided by practitioners from SBB (Roos 2006) are used for this case study.

2.3 Compensation zones

A compensation zone connects two condensation zones and consists of a line with a simple track topology—usually one or two parallel tracks with some small stations and few switches. Hence the number of possible itineraries connecting two portals is small. The routing for each train is often known a priori by introducing simple and widely used policies, such as the utilisation of each track for only one direction or the separation of the tracks between freight and passenger traffic. Moreover, minor stations enable overtakings between faster and slower trains.

The traffic density in this zone is less and the running time reserves to increase timetable stability can be introduced without creating capacity bottlenecks. In general, generous time reserves lead to a better punctuality of the railway services. However, large time reserves also cause longer travel times. Therefore, a compromise must be found. In Switzerland it is current practice to plan 7% running time supplement for passenger trains and 11% for cargo trains as rule of thumb (see also Kroon et al. 2005), without differentiating between compensation and condensation zones.

It is reasonable to assume that a time reserve close to these values satisfies the needs of stability and performance of the railway operator. On the other hand, not fixing the travel time inside the compensation zone a priori leaves more flexibility for scheduling within the zone and simplifies the fixing of the travelling times at the portals, which is fundamental for the coordination

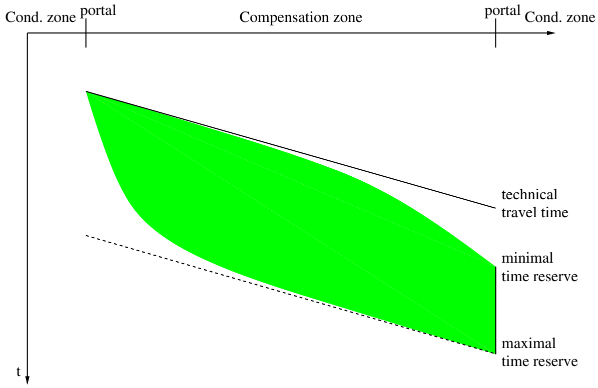


Fig. 6 Flexibility of the speed profile in a compensation zone. The *shaded zone* represents the feasible space for scheduling the train

between zones. Consequently, the speed profile should be chosen freely within certain bounds, as illustrated in Fig. 6.

2.4 Coordination among the zones

Besides scheduling the different zones individually, we have to coordinate them in order to generate a conflict-free train schedule for the entire railway network. Since the speed profile in the condensation zone is fixed, the train speed at the portal is also fixed. To assure the same speed at the portals for all itineraries inside the compensation zones it suffices to locate the portal sufficiently far from the last switch coming from the compensation zone to enable the train to reach the desired speed at the portal. Thus, the only decision variables at the portals are the passing times and tracks, which represent the boundary conditions between the zones. Once time and track are specified, each zone can be treated independently, and the union of the different locally feasible solutions provides a globally conflict-free train schedule. The sole assignment necessary on a global level is their coordination by setting the boundary conditions at the portals.

3 Model and algorithm for condensation zones

Due to the different properties of the two zones, the scheduling problem is addressed with two different models and algorithms. In this section, we present the model and algorithm to solve the train scheduling problem in a condensation zone in detail.

3.1 Conflict graph model

An algorithm to solve the train scheduling problem in a condensation zone is given in Carey and Carville (2003), which propose a heuristic approach similar to the manual methods adopted by train planners at British railways. A general model for the train routing problem, based on an input timetable, is given in Caimi et al. (2005) and Zwaneveld et al. (1996), where two different algorithms for its solution are proposed. In his PhD thesis, Zwaneveld (1997) proposes an extension of the routing problem in order to consider also deviations of planned arrival and departure times.

Our goal in this paper is to solve the timetabling and the routing problem in a condensation zone simultaneously. For this problem we propose a more general conflict graph modelling similar to Zwaneveld (1997), combined with a heuristic solution method based on a fixed point iteration, as in Caimi et al. (2005). First, we show the basic modelling if an input timetable is given. Then the extensions to the model to include also the timetabling decisions are presented.

Given is a set Z of n trains, each having a set $R_i = \{r_{i_1}, \ldots, r_{i_{m(i)}}\}$, $i = 1, \ldots, n$, of $m(i)$ possible routings connecting its entry and leaving portal within a condensation zone and passing through all minor stations where it is supposed to stop. Since the timetable is given, it is possible to calculate non-compatible routes of different trains, i.e., routes that share some track segment at the same time including safety time. Two conflicting routes are denoted $r_{p_q} \not\leftrightarrow r_{u_v}$; analogously, $r_{p_q} \leftrightarrow r_{u_v}$ implies that the routes q and v of trains p and u respectively are compatible. The set of all pairs of conflicting routes is called conflict set C. Additionally, all routes of the same train are by definition "in conflict", since only one route for each train is needed. The conflict set is described by

$$C = \left\{ (r_{i_k}, r_{j_l}) \mid i = j \ \vee \ r_{i_k} \not\leftrightarrow r_{j_l} \right\}.$$

Note that this model accommodates any safety system (both fixed and moving block systems) for conflict determination.

A feasible solution to the routing problem is a set $\mathcal{R}$ of routes r_{p_q} such that each train receives a route, i.e., $|\mathcal{R}| = n$, and the chosen routes are conflict-free, i.e., $r_{p_q} \leftrightarrow r_{u_v}$ $\forall\, r_{p_q}, r_{u_v} \in \mathcal{R}$. An instance of this problem can be visualised as a graph in which the node set is composed of the elements of $R := R_1 \cup \ldots \cup R_n$ and the edges correspond to the elements of C. This graph is called a conflict graph and has a special structure since all routes of the same train form a clique in the graph. An independent set in this graph, i.e., a set of nodes such that no two chosen vertices are connected by an edge, corresponds to a conflict-free set of routes. Note that if the instance of the routing problem is feasible, i.e., all trains can be routed, then a maximum independent set in the conflict graph includes one node from each train clique.

Zwaneveld et al. (1996) show that the train routing problem is $\mathcal{NP}$-complete with respect to the infrastructure (topology) and the train service intention, i.e., the number of train itineraries through the station area. More precisely, it is shown that the train routing problem is $\mathcal{NP}$-complete by a reduction from the satisfiability problem (SAT), if each train has at least three different routing possibilities.

In order to combine timetabling and routing in one step, the model must be generalised. We are given a train service intention instead of a timetable.

Thus, all trains have a certain time flexibility for their runs, i.e., instead of fixed passing times at portals, the passing times are relaxed to be within some time interval, which defines their time frame. Additionally, some train connections must be provided. Recall that a time discretisation for train departures/arrivals in switch regions of condensation zones is introduced. Therefore, each train time frame in the service intention implies a discrete set of departure/arrival times, denoted by $T_i = \{t + k \cdot \tau \mid k = 0, \dots, \sigma_i - 1\}$, where σ_i is the number of possible time assignments for train i, defining the slot for train i. A conflict graph can be built as follows:

1) For each train its set R_i of routes is multiplied by σ_i (the number of its possible departure/arrival times), resulting in a set of time/routes

$$R_i^* = \{r_{i_j}^t \mid r_{i_j} \in R_i,\ j = 1 \dots m(i),\ t \in T_i\}, i = 1, \dots, n.$$

Consequently, $R^* := R_1^* \cup \dots \cup R_n^*$ denotes the set of all possible time/routes assignments and corresponds to the vertex set of the conflict graph.

2) Conflicts, i.e., violated safety restrictions, between all train routes and each departure/arrival time have to be determined, which results in the conflict set

$$C^* = \left\{ \left(r_{i_k}^t, r_{j_l}^u \right) \mid i = j \ \vee \ r_{i_k}^t \not\leftrightarrow r_{j_l}^u \right\}.$$

3) Finally, additional conflicts due to broken connections must be introduced, i.e., if the train service intention includes a connection between two trains then routes with departure/arrival times that do not enable the connection are in conflict. Let denote A the set of all conflicts due to broken connections.

4) The resulting conflict graph G is composed of the vertex set R^* and the edge set $E := C^* \cup A$.

Figure 7 illustrates an example of a conflict graph with different route/time assignments, where the clique restrictions connecting nodes of the same train are omitted for clarity. Each possible route/time assignment is represented by a node. Conflicting assignments correspond to edges that connect the two respective nodes. A possible set of compatible routes is represented by

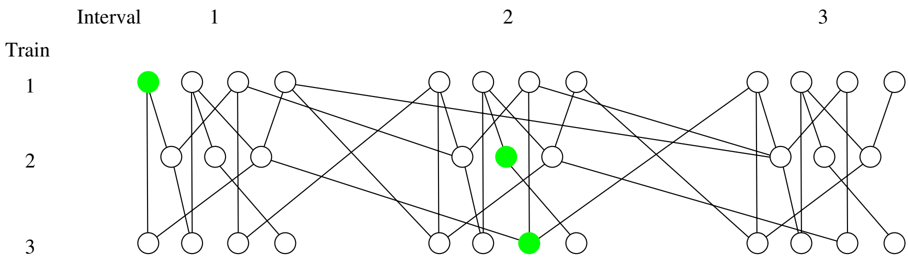


Fig. 7 Conflict graph describing the routing and scheduling possibilities for three trains through a network in a discretised timetable

the filled vertices. The size of the graph increases considerably due to the multiplication of the train routes as a consequence of the time discretisation. Therefore, the graph has to be reduced before searching for an independent set, which corresponds to a schedule for the condensation zone. The reduction of the possible itineraries is described in Section 3.3.

3.2 Fixed point iteration algorithm

The train scheduling problem can be formulated as an Integer Linear Program (ILP). Let

$$x_{ij}^t = \begin{cases} 1 & \text{if the time/route } r_{i_j}^t \text{ is assigned to train } i \\ 0 & \text{otherwise} \end{cases}$$

Recall that $m(i)$ denotes the number of different routes train i may take, n the number of trains, and σ_i the number of slots, i.e., the number of the discrete sets of departure and arrival times. With the definition of the x_{ij}^t the train scheduling problem is formulated as follows:

Problem 1 (Train scheduling problem, ILP Formulation)

$$\max \quad \sum_{i=1}^{n} \sum_{t=0}^{\sigma_i - 1} \sum_{j=1}^{m(i)} x_{ij}^t \tag{1}$$

$$\text{s.t.} \quad \sum_{t=0}^{\sigma_i - 1} \sum_{j=1}^{m(i)} x_{ij}^t \leq 1 \qquad \text{for all } i = 1, \ldots, n \tag{2}$$

$$x_{ik}^t + x_{jl}^u \leq 1 \qquad \text{for all } r_{i_k}^t \not\leftrightarrow r_{j_l}^u \tag{3}$$

$$x_{ij}^t \in \{0, 1\} \tag{4}$$

Equation (2) assures that only one vertex per clique induced by a train service is used, hence each train is assigned to a slot and route once. Equation (3) assures that no conflicting pair of routes is in the chosen set of vertices. Recall that these constraints include safety restrictions as well as broken connections. The objective function Eq. (1) maximises the number of vertices satisfying these constraints, i.e., the cardinality of the independent set. Note that the underlying safety rules are only used to determine whether two routes are conflicting or not. An assignment of 0 or 1 to all variables x_{ij}^t respecting all constraints in Problem 1 provides a conflict-free schedule for all trains in the train service intention, for which the timetable and designated routes for all trains are decided.

In condensation zones the number of switches is usually large, thus the number of itineraries to reach a point A from B is large as well, resulting in large sets R_i. The number of vertices per train is $\sigma_i \cdot |R_i| = |R_i^*|$. In Zwaneveld et al. (1996), a Branch-and-Cut algorithm is proposed to solve the problem.

Since our instances have a higher conflict density, we choose an algorithm for which it was shown that it can handle large dense problems well (Burkard 2000). In order to find a maximum independent set for Problem 1, an algorithm specially developed to solve Constrained Semi-Assignment Problems is adapted (Cochand 1993). The basic idea of this heuristic is to use a continuous relaxation of the Boolean decision variables and then iterate, starting from an interior point, towards an extremal point, which corresponds to a feasible assignment. The main advantages of using this adapted heuristic for solving the train scheduling problem are that it allows the clique structure of the graph to be efficiently exploited and that different solutions can be found by simple randomisation procedures.

For each x_{ij}^t a new variable p_{ij}^t is introduced with $p_{ij}^t = 1$ if train i is assigned to its j-th route at slot t and $p_{ij}^t = 0$ otherwise. Allowing all values $p_{ij}^t \in (0, 1)$ Algorithm 1 is conducted.

The intuition behind the iteration step (Eq. (6) in Algorithm 1) is as follows. The interpretation of the p_{ij}^t as probabilities for choosing assignment $r_{i_j}^t$ for train i can be thought of as a Bayesian interference process.

The expressions $(1 - (p_{kl}^u)^s)$ can be interpreted as the probability of not choosing $r_{k_l}^u$ for the assignment of train k. The probability of selecting $r_{i_j}^t$ for train i is penalised by all conflicting assignments to $r_{i_j}^t$. If such a conflicting

Algorithm 1 Fixed Point Iteration to find an Independent Set in n-clique graphs

Input: G with vertex set $V = \{v_{ij}^t \mid v_{i1}^0, \ldots, v_{im(i)}^{\sigma_i - 1}$ build a clique for $i = 1, \ldots, n\}$, edge set $E \subseteq \{(v_{ij}^t, v_{kl}^u) \mid i \neq k\}$, and maximum number of iterations S.
Output: A set of vertices $I \subset V$ such that vertices v_{ij}^t and v_{kl}^u, $i \neq k$, belonging to I are not connected by an edge. I is an independent set of size n or $I = \emptyset$.

Initialisation:

For every v_{ij}^t assign a value $(p_{ij}^t)^0$ such that

$$0 < (p_{ij}^t)^0 < 1 \quad \text{and} \quad \sum_{t=0}^{\sigma_i - 1} \sum_{j=1}^{m(i)} (p_{ij}^t)^0 = 1 \qquad \forall\, i \in \{1, \ldots, n\} \tag{5}$$

Iteration:

While $s < S$ and $(p_{ij}^t)^{s+1} \neq (p_{ij}^t)^s$ for some p_{ij}^t do:

$$(p_{ij}^t)^{s+1} := \frac{(p_{ij}^t)^s \prod_{r_{k_l}^u \leftrightarrow r_{i_j}^t} (1 - (p_{kl}^u)^s)}{\sum_{f=0}^{\sigma_i - 1} \sum_{g=1}^{m(i)} (p_{ig}^f)^s \prod_{r_{k_l}^u \leftrightarrow r_{i_g}^f} (1 - (p_{kl}^u)^s)} \tag{6}$$

$$i \in \{1, \ldots, n\} \quad j \in \{1, \ldots, m(i)\} \quad t \in \{0, \ldots, \sigma_i\}$$

$$s = s + 1 \tag{7}$$

Randomisation:

Randomly choose a vertex $\hat{v}_{ij}^t$ for each clique i according to their final distribution probabilities p_{ij}^t.

assignment has a high probability of being selected, then its influence on the penalty is larger. The probability of selecting a time/route r_{ij}^t is also adjusted by the probability of not choosing the alternative time/routes of the same train itinerary due to the clique structure of the graph. Thus, a feedback exists that increases the probability of choosing a likely assignment, which considerably accelerates convergence.

The denominator preserves $\sum_{f=0}^{\sigma_i-1}\sum_{g=1}^{m(i)}\left(p_{ig}^f\right)^s = 1$ for all cliques i. Without this normalisation, the probabilities would rapidly tend to zero. In theory, attractive fixed points $(p_{ij}^t)^{s+1} = (p_{ij}^t)^s$ for all i, j, and t of the iteration correspond to solutions of the scheduling problem, assuming that a solution exists. Yet, non-attractive fixed points which do not meet all restrictions may exist. This has been shown for a more general setting in Cochand (1993).

Choosing the initial distribution The p_{ij}^t can be seen as probabilities that train i chooses route j at slot t (see above). As there is no obvious reason to prefer one assignment over another, the uniform distribution could be chosen for the initialisation phase, i.e., $p_{ij}^t = \frac{1}{\sigma_i m(i)}$.

This choice has a drawback: recall that the iteration itself is deterministic. At the end of the iteration phase, the result is a probability distribution over the routes for each train. By distributing the starting probabilities uniformly, the algorithm is deterministic up to the end of the iteration phase. However, having a wide variety of solutions is preferred over a sparse range of solutions. Therefore, we choose the initial values of p_{ij}^t randomly to receive a large variation in the output.

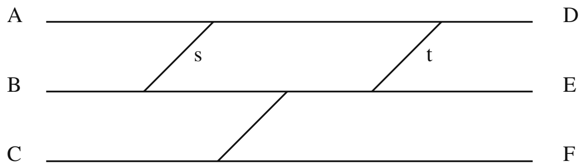
Number of iterations In practice, due to rounding off in computer arithmetic, the denominator in Eq. (6) might vanish. Due to the limited precision of computing, fixed points not corresponding to solutions occur. Therefore, the iteration is stopped after a small number of steps. Empirical evidence showed that after a few hundreds of iterations good "trends" in the distribution of the $(p_{ij}^t)^S$ are reached. Hence the randomised rounding procedure can be started early and the probability of success is still high—provided that a solution exists.

3.3 Itinerary reduction

As described in Section 3.1, a reduction of the allowed itineraries could become necessary to avoid a large conflict graph and thereby improve the performance of the fixed point iteration. Zwaneveld et al. (2001) propose to delete the dominated nodes of the conflict graph, and computations showed that it was possible to reduce the conflict graph size by 90% without losing feasibility of the problem. However, this approach is very time consuming since it does not exploit the characteristics of the railway infrastructure.

We propose a two-step reduction procedure to eliminate ineffective routing possibilities by using properties of the railway topology. It allows for very fast computation times by applying the following policy to reduce the itineraries.

Fig. 8 Example of a dominated path. The link BD via s does not conflict with the link CE and dominates therefore the link via t



Switch region

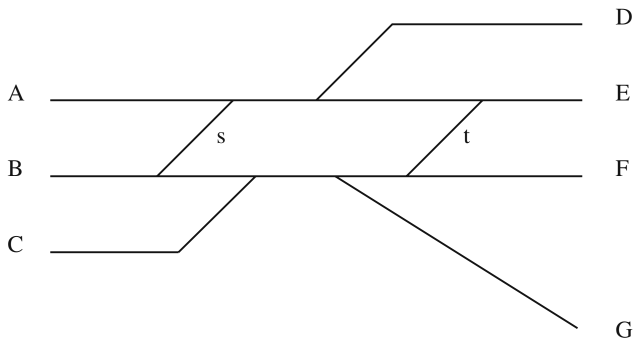
Path	AD	BD_s	BD_t	BE	CD	CE	CF
AD	-	1	1	0	1	0	0
BD_s	1	-	-	1	1	0	0
BD_t	1	-	-	1	1	1	0
BE	0	1	1	-	1	1	0
CD	1	1	1	1	-	1	1
CE	0	0	1	1	1	-	1
CF	0	0	0	0	1	1	-

Conflict Matrix

For each switch region, only a restricted number of paths connecting the entry and exit points of the region are allowed. To choose the most appropriate paths, we introduce a Boolean conflict matrix that shows whether two paths in the switch region are in conflict, i.e., whether it is possible to assign both paths simultaneously or not, similar to those described in Pachl (2002). Each row of the matrix corresponds to a path in the switch region and each entry of the matrix is a 1 if the two paths are in conflict and 0 if they are disjoint. Figures 8, 9 and Table 1 show 3 possible conflict matrices, where entries between two paths connecting same entry and exit point are represented with a dash (correspond to a 1) in order to simplify the readability.

In a first step, we remove each path that is dominated by another path having the same entry and exit points. The rows in the conflict matrix corresponding to these paths are component-wise larger than or equal to rows of dominating paths and can be easily detected. Figure 8 illustrates an example of a dominated path. Each train route using the path BD_t as a sub-route will not be considered and no trains will be scheduled on these routes. Paths which generate exactly the same row in the conflict matrix are equivalent and one of them (e.g. the longest) can be removed. This first step reduces the number of itineraries to consider while preserving the feasibility of the original train scheduling problem.

After deleting dominated paths, a second reduction step is applied where non-dominated yet similar paths are removed as well. Let $L_{\alpha\beta} \geq 0$ be a parameter indicating the number of paths from α to β to remain after the second reduction step has been applied. If $L_{\alpha\beta} = 0$ the route from α to β is not possible; otherwise we proceed heuristically by keeping at least one path that causes the minimal number of conflicts (1 in the matrix) with the other routes. If more than one path is required, we choose additional paths such that they offer the highest number of new alternatives, i.e., paths with a maximal



Switch region

Path	AD	AE	BD	BE_s	BE_t	BF	BG	CE	CF	CG
AD	-	1	1	1	0	0	0	0	0	0
AE	1	-	1	1	1	0	0	1	0	0
BD	1	1	-	1	1	1	1	0	0	0
BE_s	1	1	1	-	-	1	1	1	0	0
BE_t	0	1	1	-	-	1	1	1	1	1
BF	0	0	1	1	1	-	1	1	1	1
BG	0	0	1	1	1	1	-	1	1	1
CE	0	1	0	1	1	1	1	-	1	1
CF	0	0	0	0	1	1	1	1	-	1
CG	0	0	0	0	1	1	1	1	1	-

Conflict Matrix

Fig. 9 Example of non-dominated similar paths. The link BE via s generates other conflicts than the link via t. We prefer BE_s as it has one conflict less

number of zeros in the position where previous chosen paths had ones. Table 1 illustrates an example of a conflict matrix for an artificial switch region, where A and B are two parallel tracks in one direction and C, D, and E are three

Table 1 Conflict matrix

Path	AC_1	AC_2	AC_3	AE_1	AE_2	BC	BD	BE_1	BE_2
AC_1	–	–	–	1	1	1	0	0	1
AC_2	–	–	–	1	0	0	1	0	1
AC_3	–	–	–	0	1	1	0	1	1
AE_1	1	1	0	–	–	1	1	1	1
AE_2	1	0	1	–	–	0	1	1	1
BC	1	0	1	1	0	–	1	1	1
BD	0	1	0	1	1	1	–	1	1
BE_1	0	0	1	1	1	1	1	–	–
BE_2	1	1	1	1	1	1	1	–	–

BE_2 is dominated by BE_1, whereas AC_2 is preferred to AC_1 or AC_3 because it causes less conflicts. As second path AC_3 is better than AC_1 since it offers more new alternatives parallel tracks in the other direction. After applying the two-step reduction procedure, dominated and similar itineraries are removed. Since each feasible combination of parallel tracks between switch regions is still considered, this method maintains the global variety of routes but avoids the consideration of too many itineraries.

However, the approach does not guarantee the feasibility of the reduced conflict graph to be maintained. By decreasing the considered number of paths connecting the entry and exit point of the switch region we increase on the one hand the risk of losing feasibility but on the other hand we decrease the size of the resulting conflict graph and hence solution time. In our experiments we have never lost feasibility even if in very dense service intentions only one path per entry and exit point was considered (all $L_{\alpha\beta} = 1$ for feasible connections, see Section 4). Further tests are needed to find good values $L_{\alpha\beta}$, depending on the track topology and the train service intention.

4 Results

The conflict graph model and the fixed point iteration algorithm were implemented and tested with real data provided by the Swiss Federal Railways (SBB) for the stations of Berne and Lucerne in Switzerland. Due to different data sources, the blocking time graphs could only be calculated exactly for Lucerne. For Berne, we approximated the safety system with headway times, 150 s for the 2003 timetable and 90 s for future condensed timetables. For the same reason of data representation, computation times for Berne and Lucerne are not comparable. The tests were conducted on a Pentium 4 with a clock speed of 2.4 GHz and 3.6 GB RAM.

The main station of Lucerne is a terminal station with 12 platforms and trains arriving from four major directions. The station region has a radius of roughly 6 km and has about 40 switches within. We tested 3 different possible train service intentions, where 8, 17, and 25 trains pass Lucerne in 30, 10, and 15 min, each having an average of 30 possible routes per time slot (maximum 130) if no itinerary reduction is applied. 20 iterations ($S = 20$ in Algorithm 1) are sufficient to find feasible solutions in every scenario. Table 2 shows the results for the test case Lucerne.

The second test case is the main station of Berne; Fig. 2 roughly illustrates the layout while Fig. 3 shows the track layout of the main station's west side. Berne is a through station with 12 platforms and trains arriving from six major

Table 2 Results for Lucerne, $\tau = 120\ s$, no itinerary reduction

Scenario	#trains n	#nodes $\|R^*\|$	#edges $\|E\|$	Graph density [%]	Solution time [s]
1	8	810	58 000	9	< 1
2	17	3300	2 030 000	19	6
3	25	3800	4 200 000	29	12

directions. The station region has a radius of roughly 10 km and has about 600 switches within. We tested a strict policy where the platform track of each train in the main station is already specified, and we can therefore treat the west and the east side of the station separately. The existing buffer in the dwell and turn-around times in the main station (at least 1 min for dwell and much more for turn-around) can be exploited to adjust departure times of trains to align the time raster in both sides of the main station. In the 2003 timetable, 19 trains pass Berne in half an hour, each with an average of 300 possible routes (maximum 1433) per inbound and outbound route if no itinerary reduction is applied.

Moreover, we tested a condensed hypothetical train service intention in which 27 trains pass Berne within half an hour. The connections are according to the concept of the *integrated fixed-interval timetable* (Liebchen 2006), where in every interval of 15 min a set of trains enters the station, enables connection between all trains, and leaves the station again. This service intention is specially constructed in order to test the limits of the system. Notice that the number of trains in the east and west part of the station are not the same because some trains switch the direction in the main station, e.g., the IC Romanshorn-Zurich-Berne-Interlaken East, which arrives from Olten and leaves the station towards Thun (see Fig. 2).

By applying the two-step itinerary reduction procedure, the number of routes is reduced by a factor of 10, the number of conflicts by a factor of 80–100, and the computation time for solving the scheduling problem is reduced by a factor of 30–1000 compared to the original computation time. Note that in the resulting conflict graph, the feasibility has never been lost, even if $L^K_{\alpha\beta} = 1$ for all entry/exit points α and β for all switch regions K. $S = 100$ iterations are needed to find feasible solutions. Table 3 shows the impact of the reduction on the problem size and the computation times. Inbound and outbound trains are calculated as different trains. For the itineraries reduction $L^K_{\alpha\beta} = 1$ for all existing paths $\alpha\beta$ in all switch regions K has been used.

Table 4 shows how the conflict graph size and computation time increase by reducing the interval length τ. Whereas the number of nodes grows linearly,

Table 3 Results for Berne (fixed timetable, $\tau = 90$ s for the condensed timetables and no time discretisation for the 2003 timetable)

Scenario	Itineraries reduced	Trains in 30′	#nodes $\vert R^*\vert$	#edges $\vert E\vert$	Solution time [s]
West 2003	No	19	1400	69 000	4
West 2003	Yes	19	110	317	0.01
East 2003	No	19	5500	740 000	100
East 2003	Yes	19	350	10 000	1
East condensed	No	32	6800	7 100 000	2400
East condensed	Yes	32	250	6700	2
West condensed	No	22	1800	300 000	30
West condensed	Yes	22	200	2500	1

Impact of the itinerary reduction on conflict graph size and computation time

Table 4 Results for Berne (West side, condensed service intention, reduced itineraries)

Interval length τ [s]	Average (std dev) time loss [s]	#nodes $\|R^*\|$	#edges $\|E\|$	Solution time [s]
120	14 (17.7)	224	3600	0.05
90	7.4 (12.4)	950	76 000	0.07
60	4.8 (8.6)	1350	150 000	3
45	4.5 (8.7)	1800	280 000	3
30	2.9 (6)	2600	590 000	8
15	1.2 (1.5)	5000	2 160 000	35

Impact of the interval length τ on travel time and computation time

the number of conflict edges and the computation time grow more or less quadratically. The average additional time needed by a train to travel through the condensation zone and to accommodate the time discretisation seems marginal compared to the total travel time (around 250 seconds on average). Therefore, the choice of an interval length $\tau = 90$ (equal to the headway time) seems to be reasonable as it has an acceptable capacity loss (tolerable until 8–10% due to the short distances) while avoiding to consider too many time alternatives.

5 Summary and outlook

We introduced a railway scheduling model based on a decomposition of the railway network into condensation and compensation zones. Condensation zones lie in the vicinity of main stations of the network with high traffic demand and limited capacity. The traffic is less dense in compensation zones which connect the condensation zones. Here, slack time can be added to train runs to increase the stability of the timetable.

In this paper, we focus on the timetabling and routing problem in condensation zones. The problem is modeled as an independent set problem, which is then solved using a fixed-point iteration heuristic. In order to speed-up the computation time, a two-step reduction of the various routing possibilities has been applied. Although the procedure does not guarantee to maintain feasibility of the scheduling problem in general, we showed that even a very restrictive application performs well on real instances. Results show that timetables for large stations and dense traffic can be generated in less than a minute. It would be interesting to test the algorithm proposed here and the alternative Branch-and-Cut approach (Zwaneveld et al. 1996) on the same instances in order to compare their performances.

Ongoing research focuses on one hand on solving the timetabling problem in compensation zones. The main question of interest here is the distribution of slack times in space and time such that the stability of the resulting timetable is maximised. On the other hand, the problem of coordination between the condensation and compensation zones must be tackled. The goal of the coordination is to find suitable boundary conditions that allow to find feasible schedules for all condensation and compensation zones in a network simultaneously.

Acknowledgements We thank the Swiss Federal Railways, Infrastructure Division for funding and providing data and in particular Dr. Felix Laube, Samuel Roos, Oskar Stalder, and Dr. Raimond Wüst for insightful discussions. Furthermore, we thank the referees for their constructive comments that helped improving the quality of this paper.

References

Bourachot J (1986) Computer-aided planning of traffic in large stations by means of the AFAIG model. Rail Int 2–18

Burkard M (2000) A continuous relaxation based heuristic for a class of constrained semi-assignment problems. Ph.D. thesis, ETH Zurich

Caimi G, Herrmann T, Burkolter D (2005) Findind delay-tolerant train routing through stations. In: Fleuren H, den Hertog D, Kort P (ed) Operations research proceedings 2004. Springer, Heidelberg, pp 136–143

Carey M (1994) A model and strategy for train pathing with choice of lines, platforms, and routes. Trans Res Part B 28(5):333–353

Carey M, Carville S (2003) Scheduling and platforming trains at busy complex stations. Trans Res Part A 37:195–224

Cochand M (1993) A fixed point operator for the generalised maximum satisfiability problem. Discrete Appl Math 46(2):117–132

Herrmann T (2005) Train routings through station areas and stability of timetables. Ph.D. thesis, ETH Zurich

Huisman D, Kroon L, Lentink R, Vromans M (2005) Operations research in passenger railway transportation. In: Statistica Neerlandica., vol 59, no 4. Erasmus Research Institute Of Management (ERIM), pp 467–497. http://ideas.repec.org/p/dgr/eureri/30002129.html

Kroon LG, Zwaneveld PJ (1995) Stations: final report of phase 1. Technical report 201, Rotterdam School of Management

Kroon L, Dekker R, Vromans M (2005) Cyclic railway timetabling: a stochastic optimization approach. In: ERIM report series research in management, no ERS-2005-051-LIS. Erasmus Research Institute of Management (ERIM), RSM Erasmus University. http://hdl.handle.net/1765/6957

Laube F, Mahadevan V (2008) Bringing customer focus into every nut and bolt of the railway: Swiss Federal Railway's path into the future. In: Proceedings of the 8th World Congress of Railway Research (WCRR). Seoul, Korea

Laube F, Roos S, Wüst R, Lüthi M, Weidmann U (2007) PULS 90 - ein systemumfassender Ansatz zur Leistungssteigerung von Eisenbahnnetzen. Eisenbahntech Rundsch 3:104–107 (in German)

Liebchen C (2006) Periodic timetable optimization in public transport. Ph.D. thesis, Technical University Berlin

Lüthi M, Nash A, Weidmann U, Laube F, Wüst R (2007) Increasing railway capacity and reliability through integrated real-time rescheduling. In: Proceedings of the 11th world conference on transport research. Berkeley

Odijk M (1996) A constraint generation algorithm for the construction of periodic railway timetables. Trans Res Part B 30(6):455–464

Pachl J (2002) Railway operation and control. VTD Rail, Mountlake Terrace. ISBN 0-9719915-1-0

Roos S (2006) Bewertung von Knotenmanagement-Methoden für Eisenbahnen. Master's thesis, Institute for Transport Planning and Systems, ETH Zurich (in German)

Schrijver A, Steenbeck A (1994) Dienstregelingontwikkeling voor Railned (timetable construction for Railned). Technical report, C.W.I. Center for Mathematics and Computer Science, Amsterdam (in Dutch)

Serafini P, Ukovich W (1989) A mathematical model for periodic scheduling problems. SIAM J Discrete Math 2(4):550–581

Zwaneveld PJ (1997) Railway planning—routing of trains and allocation of passenger lines. Ph.D. thesis, Erasmus University Rotterdam

Zwaneveld P, Kroon L, van Hoesel S (2001) Routing trains through a railway station based on a node packing model. Eur J Oper Res 128(1):14–33(20)

Zwaneveld PJ, Kroon LG, Romeijn HE, Salomon M,. Dauzère-Pérès M, Van Hoesel SPM, Ambergen HW (1996) Routing trains through railway stations: model formulation and algorithms. Transp Sci 30(3):181–194