Design of a new railway scheduling model for dense services

Gabrio Caimi[1], Dan Burkolter[2], Thomas Herrmann[3], Fabian Chudak[1], Marco Laumanns[1]

[1] Institute for Operations Research, ETH Zurich, Switzerland, e-mail: {caimig,chudak,laumanns}@ifor.math.ethz.ch

[2] SMA und Partner AG, Zurich, Switzerland, e-mail:d.burkolter@sma-partner.ch

[3] R+R Burger und Partner AG, Baden, Switzerland, e-mail:herrmann@rrag.ch

Abstract

We address the problem of generating a global detailed conflict-free railway schedule for a given set of train lines and frequencies and we propose a decomposition of the railway network into condensation and compensation zones in order to solve the train scheduling problem. Condensation zones lie in the proximity of main stations, where available capacity is limited and trains are therefore required to travel with maximum speed. On the other hand, traffic is less dense in the compensation zones, which connect the condensation zones. Thus, time reserves can be introduced to train runs in compensation zones in order to increase stability of the timetable. Each zone is then dealt with according to its specific properties.

In this paper we focus particularly on the timetabling and routing problem in condensation zones. In order to gain structure in the schedule and therefore simplify the dispatcher's work, we introduce a policy to schedule trains using a time discretisation. The problem is modeled as an independent set problem in a conflict graph, which is then solved using a fixed-point iteration heuristic. Results show that even large-scaled train scheduling problems with dense timetables and large stations can be solved within a minute.

Keywords

Timetable, Railway network decomposition, Conflict-free scheduling, Independent set problem

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 also must 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 owner of railway infrastructure in Switzerland, is currently investing effort into the development of efficient methods for generating and operating railway schedules.

The strategic timetable generation is usually done in two steps:

(i) In the first step, with the help of origin-destination matrices, 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 feasible. A train service intention consists of train lines and frequencies specifying the customer-relevant information, such as stop stations, interconnection possibilities, arrival and departure platforms at stations and rolling stock.

(ii) In a second step, the feasibility of this service intention is checked by generating a feasible schedule. If this is possible, a schedule is provided as proof of feasibility, otherwise both steps have to be repeated 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. In particular, 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 the 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 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. The minor traffic density in rural regions enables introducing time reserves. 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, that 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

Related work (see [6] 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 ([12]). 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 a minutely basis. PESP enables to schedule trains in a relatively large railway network (such as The Netherlands, see e.g. [10]), 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. PESP, as well as other works (e.g. [3]), assumes infinite capacity in main station regions.

In the second approach, the detailed topologies of a local region, typically a main station area, are taken into consideration for producing conflict-free timetables ([1], [4]) or checking the feasibility of a given rough timetable ([2], [13]).

However, 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 a Dutch project called DONS in collaboration with the Dutch Railways. In [7] and [11] a two-level approach for a decision support system to create conflict-free timetables for the Dutch Railways is presented. In the upper level, the train service intention is known, as well as an aggregated railway topology. The module CADANS, based on the PESP model, supports the generation of cyclic hourly draft timetables. In the lower level, the timetable generated by CADANS is checked for feasibility with respect to the detailed routing of trains through railway stations and the corresponding safety system. This module is called STATIONS (model and algorithms are presented in [13] and [14]). This approach seems to be very interesting for our aims, however it does not take into consideration the different properties of condensation and compensation zones, which should be addressed with different scheduling policies to cope with their distinct characteristics. 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 the geographical properties of the regions. Within conurbations, both the topology of the railway network and the layout of train lines are typically complicated. 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 local 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.

Hence, 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 complicated and train frequencies are high, and the *compensation zones*, which have simple topologies with lower train frequencies. Figure 1 illustrates the subdivision into zones for the Plateau region in Switzerland.

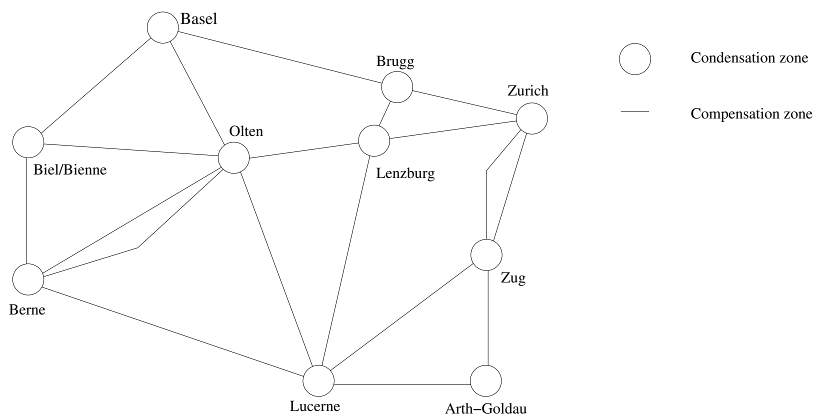

Figure 1: Possible representation of the core network of the Swiss railways divided in condensation and compensation zones.

2.1 Scheduling by network separation

In order to account for different timetable 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 removed from the condensation zones and moved 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 complicated, and many different itineraries to travel between portals and platforms exist. Here, the apt 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 in different directions. These stretches are connected by switch regions which allow reaching all tracks. The important fact is that the switch regions are short (usually up to $500\ m$) relative to the stretches and are, therefore, passed quickly by trains. Figure 2 shows a rough layout of the track topology for the station region of Berne. The most complicated switch regions lie east and west of the station just near the platforms. Their topologies are shown in the Figures 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. Therefore, trains are required to travel through the condensation zone with maximum speed, i.e., no time reserves are included. In contrast to the compensation zones, the speed profile is no longer free. It is

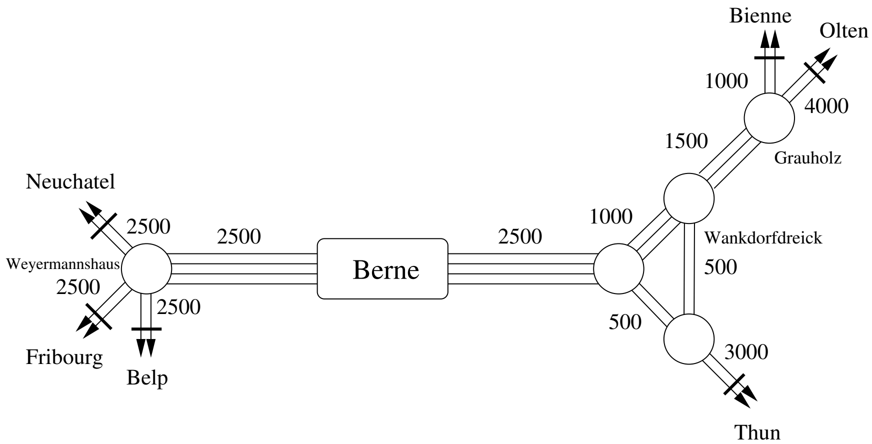

Figure 2: Sketch of the condensation zone Berne. Approximate distances given in meters.

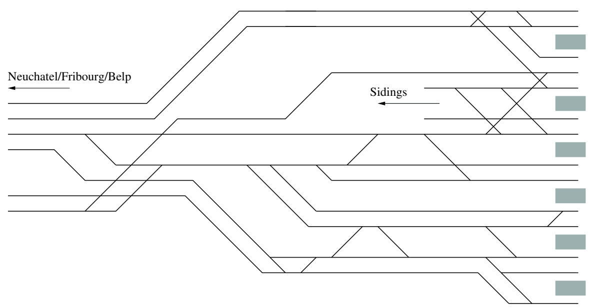

Figure 3: Switch region topology in the west of the Berne main station.

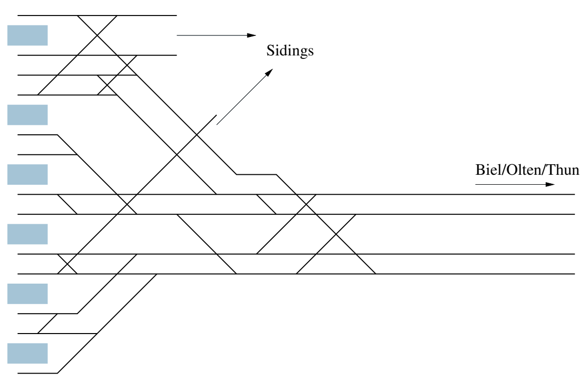

Figure 4: Switch region topology in the east of the Berne main station.

sufficient to assign one passing time per train (e.g. at the portal or at the platform) from which all the passing times within the condensation zone can be derived, once the itinerary has been fixed.

By condensing the service intention, the dispatchers will be put more under pressure and must respond to schedule disruptions faster than now. [9] shows that for a draft condensed schedule the dispatchers will have to react four times as often as nowadays to resolve disruptions. Thus, the adoption of a new scheduling paradigm for the condensation zones targets at the simplification of the dispatchers' work.

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 Figure 1. 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 use the least capacity possible. The speed profiles of the trains are adapted slightly within the condensation zones to fulfill 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 $T, \tau \in \mathbb{R}$ 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 number of intervals needed for travelling through the region K by

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

The train run must use exactly $p_i^K(\tau)$ intervals, since 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. To simplify matters, let $p_i := p_i^K(\tau)$ 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:*

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

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 inside the switch region begins at interval k_i and takes exactly p_i intervals for all trains. Notice that 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. 1 second) more or less matches a standard schedule without time discretisation.

Detection of conflicts becomes simpler in a coarse-grained discretised schedule. The dispatchers gain better comprehensibility of the schedule and can therefore make decisions quickly 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 make a compromise between a large value to simplify the dispatcher's work and small values to minimise time loss. It seems that a choice of τ close to the headway time or its fractions provides the best performance.

2.3 Compensation zones

A compensation zone connects two condensation zones and consists of one 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 train and slower trains.

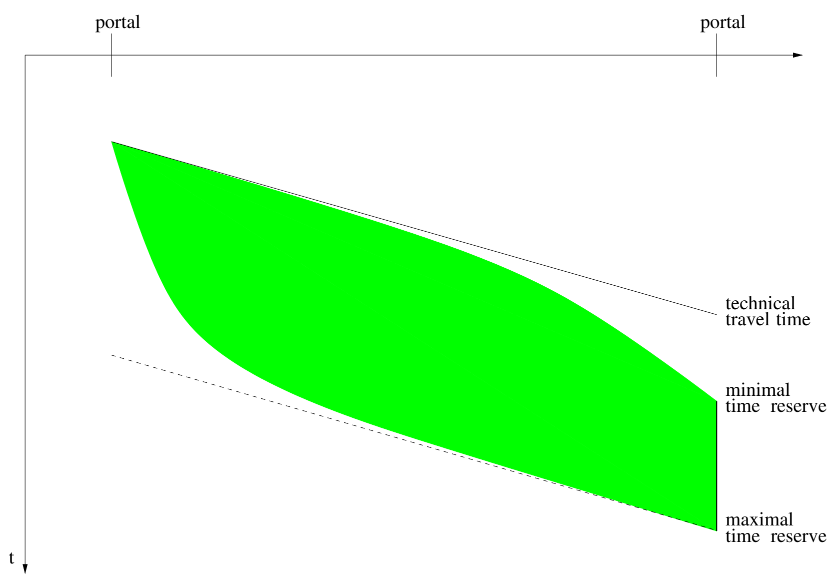

Figure 5: Flexibility of the speed profile in a compensation zone. The grey zone represents the feasible space for scheduling the train.

As this zone has less traffic density, time reserves in the running time of trains 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, high 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 [8]), 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 company. 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 between zones. Consequently, the speed profile should be chosen freely within some interval, as illustrated in Figure 5.

2.4 Coordination of the zones

To generate a conflict-free train schedule for the entire railway network, we need to coordinate the different zones. Since the speed profile in the condensation zone is fixed, train speed at the portal is also fixed. In order 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 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.

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

3 Model and algorithm for condensation zones

3.1 Conflict graph model

An algorithm to solve the train scheduling problem in a condensation zone is given in [4], which proposes a heuristic approach similar to those adopted by train planners using manual methods used by British railways. A general model for the train routing problem, based on an input timetable, is given in [2] and [13], where two different algorithms for its solution are proposed.

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 and a heuristic solution method based on a fixed point iteration, as in [2]. First, we show the basic modelling if an input timetable is given. Then the extensions to the model in order to include timetabling 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 $r_{p_q} \nleftrightarrow r_{u_v}$; 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 "in conflict", since only one route for each train is needed. The conflict set is described by

$$C = \{(r_{i_k}, r_{j_l}) \,|\, i = j \ \vee \ r_{i_k} \nleftrightarrow r_{j_l}\}.$$

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 by the elements of $R_1, \ldots, R_n$ 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 build 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 clique.

[13] shows that the train routing problem is 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 NP-complete by a reduction from SAT, if each train has at least three different routing possibilities.

In order to combine timetabling and route finding 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 the 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 $T_i = \{t + k \cdot \tau \,|\, k = 0 \ldots \sigma_i - 1\}$, 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:

- 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 $R_i^* = \{r_{i_j}^t \,|\, r_{i_j} \in R_i,\ j = 1 \ldots m(i),\ t \in T_i\}$, $i = 1, \ldots, n$.
- Conflicts, i.e. violated safety restrictions, between all train routes and each departure/arrival time have to be calculated. It results in the conflict set

$$C^* = \{(r_{i_k}^t, r_{j_l}^u) \,|\, i = j \ \vee \ r_{i_k}^t \nleftrightarrow r_{j_l}^u\}.$$

- 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.

Figure 6 illustrates an example of a conflict graph with different route/time assignments. However, 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 must be first reduced in order to find an independent set, which now corresponds to a schedule for the condensation zone. The reduction of the possible itineraries is described in Section 3.3.

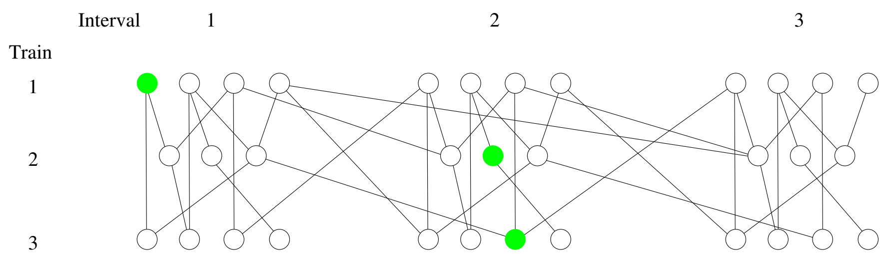

Figure 6: Conflict graph. Three trains travelling through a network. Each possible route/time assignment is represented by a node, assignments that conflict correspond to edges that connect the two respective nodes. The clique restrictions connecting nodes of the same train are omitted here for clarity. A possible set of compatible routes is represented by the filled vertices.

3.2 Fixed point iteration algorithm

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

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

Remember 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)

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

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

$$x_{ik}^t + x_{jl}^u \leq 1 \qquad \textit{for all } r_{i_k}^t \nleftrightarrow r_{j_l}^u \tag{3}$$

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

Equation (2) assures that only one vertex per clique induced by a train service is used and hence, each train is assigned to a slot and route exactly once. Equation (3) assures that no incompatible 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 (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 $\sigma_i \cdot \mid R_i \mid = \mid R_i^* \mid$. Because of this huge number of vertices and edges in the conflict graph, the Branch & Bound Algorithm described in [13] seems unapplicable. However, by accepting a probabilistic algorithm that finds solutions to feasible instances in the majority of the cases, the train scheduling problem can be solved in reasonable time.

In order to find a maximum independent set for Problem 1, an algorithm specially developed to solve Constrained Semi-Assignment Problems is adapted (see [5]). The basic idea of this heuristic is to make a continuous relaxation of the boolean decision variables and then evolve, 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 many varying solutions can be found by including randomization.

For each x_{ij}^t a new variable p_{ij}^t is introduced with $p_{ij}^t = 1$ if train schedule 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.

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 $i = 1, \ldots, n\}$ and edge set $E \subseteq \{(v_{ij}^t, v_{kl}^u) \mid i \neq k\}$
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 $I = \emptyset$.

Initialization:

Choose a maximum number of iterations S. 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:

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

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

Randomization:

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

The intuition behind the iteration step (Equation (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. $(1 - (p_{kl}^u)^s)$ can be interpreted as the probability of not choosing $r_{k_l}^u$ as 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 assignment has a high probability of being selected, then its influence on the penalty is larger. The probability of selecting a time/route 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 $\sum_{f=0}^{\sigma_i - 1} \sum_{g=1}^{m(i)} (p_{ig}^f)^s = 1$ for all cliques i. Without this normalization, 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 [5].

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 initialization 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 to 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 (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 at most some hundreds of iterations good "trends" in the distribution of the $(p_{ij}^t)^S$ are reached. Hence the randomized 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 is necessary to avoid a large conflict graph and therefore improve the performance of the fixed point iteration. [14] proposes 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 characteristics of the railway infrastructure.

We propose a reduction of the routing possibilities by using properties of the railway topology. We apply the following policy to reduce itineraries. For each switch region, only a restricted number of paths connecting the entry and exit points of the region are allowed. This number is usually set to 1 or 2. 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 simultaneously or not. In a first step we remove each path that is dominated by another path having the same entry and exit points. After deleting these paths, we choose the path that causes the minimal number of conflicts for each combination of entry/exit point. Moreover, if more than one path is required, we choose additional paths such that they offer the highest number of new alternatives. Table 1 illustrates an example of a conflict matrix for a switch region, where A and B are two parallel tracks in one direction and C, D and E are 3 parallel tracks in the other direction.

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	-	-

Table 1: Conflict Matrix. 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.

Hence, 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.

The approach does not guarantee the feasibility of the reduced conflict graph, yet the computation is very fast and can be adjusted to be more or less aggressive in the reduction. Computational results are very promising, since feasibility was never been lost for the considered instances, see Section 4 for results.

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. The tests were conducted on a Pentium 4 with a clock speed of 2.4 GHz.

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 minutes each having an average of 30 possible routes (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; Figure 2 roughly illustrates the layout. Berne is a through station with 12 platforms and trains arriving from six major directions.

Scenario	#trains	#nodes	#edges	Solution time [s]
1	8	810	58 000	< 1
2	17	3300	2 030 000	6
3	25	3800	4 200 000	12

Table 2: Results for Lucerne, $\tau = 120$.

The station region has a radius of roughly 10 km and has about 600 switches within. As the train service intention for a condensation zone already includes the designated platform in the major station, it is possible to treat the west and the east side of the station separately. In the 2003 timetable, 38 trains pass Berne in half an hour each having an average of 300 possible routes (maximum 1433), if no itinerary reduction is applied. Moreover, we tested a condensed hypothetical train service intention in which 54 trains pass Berne within half an hour. This service intention is specially constructed in order to test the limits of the system. By reducing the allowed routes, we obtain a conflict graph where the number of routes is reduced by a factor 10, the conflicts by 80–100 and the computation time by 30–1000 without losing feasibility of the problem instances. $S = 100$ iterations are needed to find feasible solutions. Table 3 shows the impact of the reduction on the problem size and the computation time. Note that due to different data representation, computation times for Berne and Lucerne are not directly comparable.

Scenario	Itineraries reduced	Trains in $30'$	#nodes	#edges	Solution time [s]
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

Table 3: Results for Berne. Impact of the itinerary reduction on conflict graph size and computation time.

Table 4 shows how the conflict graph size and computation time increase by reducing the interval length τ. Whereas the number of nodes grows linear, the number of conflict edges and the computation time grow more or less quadratically.

Interval length τ	#nodes	#edges	solution time [s]
90	950	76 000	1
60	1350	150 000	3
45	1800	280 000	3
30	2600	590 000	8
15	5000	2 160 000	35

Table 4: Results for Berne (West side, condensed service intention, reduced itineraries). Impact of the interval length τ on the computation times.

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 where available capacity is limited. On the other hand, traffic is less dense in compensation zones which connect the condensation zones. Thus, here slack time can be added to train runs in order to increase 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. Results show that timetables for large stations can be generated in less than a minute.

Ongoing research focuses on the one hand on solving the timetabling problem in compensation zones. Main question of interest here is the placement of slack times such that 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 boundary conditions, s.t. solving the train scheduling problems in all condensation and compensation zones becomes possible.

Acknowledgements

We would like to thank the Swiss Federal Railways for funding and providing data and in particular Thomas Graffagnino, Dr. Felix Laube, Samuel Roos, and Dr. Raimond Wüst for insightful discussions.

References

[1] J. Bourachot. Computer-aided planning of traffic in large stations by means of the AFAIG model. In *Rail International*. 2-18, 1986.

[2] G. Caimi, D. Burkolter, T. Herrmann. Finding Delay-Tolerant Train Routings through Stations. In *Operations Research Proceedings 2004*. 136–143, 2005.

[3] M. Carey. A model for train pathing with choice of lines platforms and routes. In *Transportation Research Part B*. 28:333-353, 1994.

[4] M. Carey, S. Carville. Scheduling and platforming trains at busy complex stations. In *Transportation Research Part A*. 37:195-224, 2003.

[5] M. Cochand. A Fixed Point Operator for the Generalised Maximum Satisfiability Problem. In *Discrete Applied Mathematics*. 46:117–132, 1993.

[6] D. Huisman, L. Kroon, R. Lentink, M. Vromans. Operations Research in Passenger Railway Transportation. In *Statistica Neerlandica*. 59(4):467-497, 2005.

[7] L.G. Kroon, P.J. Zwaneveld. STATIONS: final report of phase 1. *Report 201 of the Rotterdam School of Management*. Rotterdam, 1995.

[8] L.G. Kroon, R. Dekker, M.J.C.M. Vromans. Cyclic Railway Timetabling: a Stochastic Optimization Approach. In *ERIM Report Series Research in Management*. Rotterdam, 2005.

[9] T. Herrmann. Stability of Timetables and Train Routings through Station Regions. *PhD Thesis, ETH Zurich*. 2006.

[10] M. Odijk. A constraint generation algorithm for the construction of periodic railway timetables. In *Transportation Research Part B*. 30(6):455-464, 1996.

[11] A. Schrijver, A. Steenbeck. Dienstregelingontwikkeling voor Railned (Timetable construction for Railned). *Technical Report, C.W.I. Center for Mathematics and Computer Science, Amsterdam*. 1994.

[12] P. Serafini, W. Ukovich. A Mathematical Model for Periodic Scheduling Problems. In *SIAM Journal on Discrete Mathematics*. 4(2), 1989.

[13] P.J. Zwaneveld, S. Dauzère-Pérès, S.P.M van Hoesel, L.G. Kroon, H.E. Romeijn, M. Salomon, and H.A. Ambergen. Routing Trains through Railway Stations: Model Formulation and Algorithms. In *Transportation Science*. 30(3):181–194, 1996.

[14] P.J. Zwaneveld, L.G. Kroon, S.P.M van Hoesel. Routing Trains through a railway station based on a node packing model. In *European Journal of Operational Research*. 128:14–33, 2001.