The Periodic Service Intention as a Conceptual Frame for Generating Timetables with Partial Periodicity [*]

Gabrio Caimi, Marco Laumanns, Kaspar Schüpbach,
Stefan Wörner, Martin Fuchsberger
Institute for Operations Research, ETH Zurich, Switzerland
e-mail: {caimig,laumanns,kaspasch,swoerner,fumartin}@ifor.math.ethz.ch

Abstract
Many railway companies in Europe operate periodic timetables. Yet, most timetables are not entirely periodic but have a mixture of different periodicity and many exceptions to cope with changing demand. Current approaches for automatic timetable generation are not able to deal with such partially periodic structure, but consider only fully periodic inputs. We therefore introduce the periodic Service Intention (pSI) as a framework where the customer-relevant information about train services can be described, including their periodicity information. We then address the problem of finding a feasible timetable that fulfills the requirements specified in a pSI without the need of manual postprocessing. We solve this problem by projecting intended train runs over equivalence classes and thereby reducing the pSI to an augmented instance for periodic timetabling. Thus it is possible to use existing models for periodic scheduling, such as the PESP, to generate periodic timetables with partial periodicity, which are finally rolled out to obtain the desired daily schedule according to the commercial requirements of the pSI. Results for a test case from the timetable of central Switzerland in 2008 show that this approach needs only slighly longer computation time than for a fully periodic instance, but the additional time is compensated by the fact that postprocessing becomes unnecessary and by the better quality of the solution. The approach is particularly well suited for offers with a strong periodicity but some irregularities, which could not be treated properly by existing methods.

Keywords
Timetable, Periodicity, Periodic Service Intention, Periodic Event Scheduling Problem, Projection, Equivalence Class

1 Introduction

Many railway companies in Europe operate periodic timetables. In Switzerland, for instance, the periodicity of the train services is considered a main driver of the strong increase in passengers since its introduction in 1982. The periodicity makes it easier to remember the departure times such that the passengers can be much more spontaneous in taking a train, even without having a timetable at hand. Nowadays, periodicity is considered a substantial part of the service offer and most of the trains are scheduled with a half-hour, one hour or two hours periodicity.

---

[*] We thank the Swiss Federal Railways, Infrastructure Division for funding and providing data and in particular Dr. Felix Laube, Muriel Perron, Samuel Roos, and Dr. Raimond Wüst for insightful discussions.

The demand, however, is not distributed uniformly over the day, and also different days of the week can have different demands. Morning peak hours involve a large demand on trips from suburbs to the city centers, and in late afternoon in opposite direction. Moreover, in the evenings there is less demand than during the day and it is focused on direct connections rather than on short trip times. Exceptions in the periodicity of the timetable are therefore important and useful to cope with these irregularities of the demand. For instance, the IC train from Zurich to Berne is scheduled half-hourly with departure at the minutes 00 and 32. During the morning peak, the frequency is increased with an additional train at 7:18. Another example of irregularity is the hourly IC from Zurich to Lucerne, which is usually planned to have only two intermediate stops, but in the late evening it stops at five additional stations and the trip takes 16 minutes longer. Finding an appropriate balance between the interest in a periodic timetable and the non-periodic demand is crucial and should be supported by suitable planning tools.

The common approach for generating a timetable is to first develop a rough description of the commercial offer for a whole day, based on the demand, which is then formalised to be used as an input for automatic timetabling and scheduling methods. To our knowledge, a formal framework for the specification of partial periodic train services is still missing in the literature. In [23, 6] the concept of the *Global Service Intention* (GSI) was proposed as a description of the commercial offer that can be found is an existing timetable. It contains the passenger-relevant information for each service during a day and serves as reference for online rescheduling, where changes in the timetable should focus on reducing consequences for customers.

Similarly, the issue of partial periodicity has also not been addressed satisfactorily in the literature on timetable generation methods. For both cases of periodic [22, 10] and non-periodic [3, 20] timetabling, there exist many models and algorithms in the literature with impressive real-world applications [5, 9]. Periodic timetables with a few non-periodic exceptions, however, have usually been treated as entirely periodic with some manual post-processing. This procedure has the obvious drawback that only one part of the day, usually the average hour with basic periodicity or the hour of highest frequency, is optimised. The timetable for the rest of the day is then obtained by simply removing trains from the generated timetable, or by manually adapting or inserting extra trains. This sequential approach could lead to suboptimal solutions for the parts of the day which are not directly addressed by the optimisation algorithm, which could be avoided formulating a simultaneous problem that constructs the entire timetable in one step.

In this paper we address the problem of generating timetables from a description of the commercial offer including arbitrary partial periodic structure. We present a new approach how the partially periodic structure of modern timetables can be described and effectively exploited for timetable generation. In particular, we want to generate timetables that balance objective values in different parts of the day. We approach this problem by first creating a framework where a daily commercial offer with this partial periodic structure can be formally described. We then devise a *projection* algorithm that reduces the problem to an augmented periodic problem. Well-known methods for periodic timetabling can be then applied, and the resulting timetable is finally *rolled out* in order to obtain the desired daily schedule fulfilling the commercial requirements.

2 Commercial Offer and Periodicity: the periodic Service Intention

As an interface between commercial offer and technical process planning, we introduce the concept of a *periodic service intention* (pSI). It consists of all services that a railway company would like to offer to the customers during a day. Each train service is specified, including the line, stopping stations, interconnection possibilities, periodicity, and time frame. The pSI is not the representation of a technical timetable, but it describes the commercial offer and contains therefore only the customer-relevant information. The precise departure and arrival times as well as the detailed train routing will be decided subsequently during the timetable generation. Hence, the pSI is an input to the timetabling problem.

The purpose of the pSI description is to have a framework in which potential commercial offers can be developed and evaluated. It serves as a starting point for process planning and analysis, e.g., timetable generation or rolling stock planning, as well as a level for discussions and negotiations between the railway infrastructure operator and the train operator companies.

We now formally describe the pSI. Its basic element is the train run with the passenger relevant description, i.e., stop stations, trip times, dwell times, time frame, and periodicity.

Definition 1 (Train run) *A train run z is defined as the run over $K+1$ nodes in the topology, repeated R times with periodicity ρ minutes:*

$$z = ((v_k, t_{stop}^{k-}, t_{stop}^{k+}, t_{trip}^{k-}, t_{trip}^{k+}, \omega_k^-, \omega_k^+)_{k=0}^K, \rho, R), \tag{1}$$

where $v_k \in V$ is the node in the topology visited in the k-th step of the train run, $t_{stop}^{k-/+}$ the minimal and maximal stopping time of the train in the station node (a value of zero means that the train passes the node without stopping), $[t_{trip}^{k-}, t_{trip}^{k+}]$ defines the allowed interval of the trip time between v_{k-1} and v_k, and $[\omega_k^-, \omega_k^+]$ is the (optional) time slot for the departure event of the first train recurrence. Let Z be the set of all train runs.

Each train run should have at least one associated time slot in order to approximately locate it temporally in the day. The time slots for all relevant events of the train run will then be derived from the time slots given in the pSI, as described in Section 4.1. They should not be too restrictive to enable flexibility in the timetable generation process but also not too large to avoid ambiguities, which could hamper the reduction to a periodic problem.

Connections between trains also belong to a commercial description, as they permit the passenger to change train in an efficient way to continue their trip. Note that it is necessary to specify between which repetition of both train runs the connection takes place. This could be seen as a restriction for the scheduling algorithm, but it is a necessary information for the projection procedure and in reality it is usually decided in the planning phase at which repetition the connection will occur.

Definition 2 (Connection) *A connection c is defined as the possibility for the passenger to change from train run z_1 to train run z_2 in station $v \in V$,*

$$c = (z_1, z_2, v, r_1, r_2, \theta^+), \tag{2}$$

where both train runs z_1 and z_2 travel through station node v, and the connection takes place for the first time during the r_1-th repetition of train run z_1 and the r_2-th repetition *of train run z_2. As a consequence, the connection has a periodicity of $\tilde{\rho} := lcm(\rho_1, \rho_2)$ minutes and will be repeated*

$$\tilde{R} := \min(\lceil \frac{\rho_1(R_1 - r_1 + 1)}{\tilde{\rho}} \rceil, \lceil \frac{\rho_2(R_2 - r_2 + 1)}{\tilde{\rho}} \rceil))$$

times. The connection takes place in at most θ^+ minutes. A lower bound θ^- is given by the properties of the considered station, computed as the minimal time that passengers need to go from the arrival platform to the departure platform. Let C be the set of all connections.

Time dependencies between train events are also an important commercial requirement, as they separate two different train runs in time that are (partially) covering the same demand.

Definition 3 (Dependency) *A dependency d is defined as a time constraints between two events of the periodic service intention, where*

$$d = (z_1, z_2, k_1, k_2, r_1, r_2, \theta^-, \theta^+). \tag{3}$$

The departure event of the k_1-th node of train run z_1 should occur between θ^- and θ^+ minutes before the k_2-th node of train run z_2. The dependency takes place for the first time during the r_1-th repetition of train run z_1 and the r_2-th repetition of train run z_2. As a consequence, the connection has a periodicity of $\tilde{\rho}$ minutes and will be repeated $\tilde{R}$ times, computed in the same way as in the connection case. Let D be the set of all dependencies.

For instance, the half-hourly connection between Zurich and Berne is assured by the temporal separation of the hourly trains St.Gallen–Geneva and Romanshorn–Brig. In this case a dependency entry can assure that the two trains together offer a half-hourly service on their common section Zurich–Berne. The dependency can be restrictive, forcing an exact time difference, or can just avoid trains to be too close in time.

We can now formally define the periodic Service Intention, collecting all these information and additionally giving the value T representing the period time, which we want to reduce the pSI to, in order to apply algorithms for periodic timetabling.

Definition 4 (Periodic Service Intention) *A periodic service intention (short pSI) $\mathcal{G}$ for a given railway topology is defined as*

$$\mathcal{G} = (T, Z, C, D), \tag{4}$$

where $T \in \mathbb{R}^+$ is the considered time period, Z the set of all (partial periodic) train runs, C the set of all connections, and D the set of all dependencies.

Finally, we can formally describe the timetable generation problem that we address in this paper:

Problem 5 (Timetable generation) *Given a pSI $\mathcal{G}$, the timetable generation problem has the goal to find a timetable which fulfils the commercial requirements of the pSI and is feasible under macroscopic safety requirements.*

3 PESP for the Periodic Case

This section shortly reviews the Periodic Event Scheduling Problem (PESP), a well-known model for generating fully periodic schedules introduced by Serafini and Ukovich [22], which was first applied to train scheduling by Schrijver and Steenbeck [21]. For a comprehensive review over the PESP model we refer to [8]. In our approach we use the PESP as a sub-procedure for generating periodic schedules from the projected problem, see Section 4 for details.

A schedule on the macro level consists of a list of departure and arrival times at the nodes (stations) in the aggregated network for all trains running within a time period T. Each departure or arrival of a train at a node is called an *event* i which takes place at a certain time π_i and is repeated every T minutes.

Interdependencies between events are modeled as constraints in the PESP. These constraints always concern two events i and j and define the minimum and maximum periodic time difference l_{ij} and u_{ij} between the two. The constraint bounds l_{ij} and u_{ij} are given as data of the model, and scheduling is then about finding event times π_i for each event i that fulfill all constraints of the form

$$l_{ij} \leq \pi_j - \pi_i + T p_{ij} \leq u_{ij}. \tag{5}$$

The integer variables p_{ij} allow the constraints to be fulfilled in the periodic sense. The events and constraints constitute the elements of the PESP. This can be represented as a directed (multi-)graph $G = (V, A)$ whose node set V is the set of events and whose arcs A denote the constraints. With each arc $a \in A$ connecting nodes $i, j \in V$, we associate corresponding lower and upper bounds l_a and u_a on the periodic time differences of events $i, j \in V$.

Various rules and restrictions that exist in the railway business can be modeled via PESP constraints of the form (5). The most important are the trip time on a track, the dwell time in a station, connection constraint between two trains in a station, and headway constraints, separating two trains running on the same track by at least the headway time h. The headway constraints do not prevent overtaking of trains during the run on the same track in case in a trip time arc it holds $u_{ij} - l_{ij} \geq 2h$. This is, of course, impossible without a collision. The problem can be solved, for instance, by reducing the trip time difference for each trip arc. Liebchen and Möhring [10] propose to subdivide an initial trip arc into new smaller ones such that $u_a - l_a < 2h$ for every new trip arc. There are many other constraints that should be considered in the timetable generation and can be modeled as PESP constraints [19, 8].

An objective function gives a measure of the quality of a schedule and guarantee that the output is a solution of maximum quality. This guaranteed optimality is an advantage of the computer-generated railway timetables compared to the human-made ones. A description of possible optimisation goals can be found in [19, page 57-64]. Typical goals are minimisation of the total passenger travel time, minimisation of the required number of train units or maximisation of some measure of robustness.

The PESP can be solved by the corresponding integer linear program (ILP) formulation [10, 19, 15]. Algorithms especially designed for the PESP problem have also been developed, e. g., constraint propagation [21], genetic algorithms [16], branch-and-cut [13], constraint generation [17] or adapted backtracking algorithm [22]. These are specialised algorithms for finding feasible solutions quickly. However, for optimised solutions mostly ILP solvers are used.

The *Cycle Periodicity Formulation* (CPF) is an alternative MIP formulation which turned out to be much more efficient in practice [8, 18, 19]. Instead of solving for the event time variables π_i, it solves for periodic tensions x_a. The tensions are the time differences between the two related events $x_a = \pi_j - \pi_i + Tp_a$ and must obey the bounds $l_a \leq x_a \leq u_a$ for each constraint $a \in A$. For periodic tensions to yield a periodic potential π_i at each node, the sum of all tensions along a cycle must be equal to an integer multiple of T, hence

$$\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = Tq_C, \tag{6}$$

where q_C is the integer number of period jumps along the cycle C, C^+ the set of the arcs on the cycle direction and C^- the set of arcs in the opposite direction.

The number of cycles in a graph can be exponential in the number of nodes, but it can be shown that it is sufficient to require (6) to hold for an *integral cycle basis* B of G [11, 12]. Such a basis has the property that each cycle C in G is a linear combination with integer coefficients of the cycles in B.

Finally, in [1] the PESP is extended to provide arrival and departure time slots for each event instead of exact times, resulting in the Flexible-PESP model. While remaining in the PESP framework, this model allows for more scheduling flexibility on the detailed level.

4 The Partial Periodic Case

The generation of partial periodic timetables from a given pSI is explained in the following. The procedure is divided into four steps:

1. Slot propagation
2. Projection to the periodic case
3. Solving of the periodic problem
4. Rolling-out

An overview of the procedure is given in Figure 1. The steps are explained in the upcoming sections with reference to the example of the Figure. Section 4.5 shows the conditions for the equivalence of the projected periodic and the original partial periodic problem.

4.1 Slot Propagation

The slot propagation is a preprocessing step that generates time slots (event time bounds) for each event. Some events have given time slots, defined in the train runs of the pSI (see Section 2) . For the other events, the slots can be derived using constraint propagation.

Figure 2 shows how the time slots are propagated within a train run. In Figure 2a a train run from A to B, with given time slots (green) for departure in A and arrival in B is considered. For the events in-between, time slots can be computed using constraint propagation. Feasible event times at point P are bounded below by the earliest possible departure time in A plus the fastest allowed trip time from A to P. Similarly, the event times have upper bounds. The event times reachable from the given departure time slot are

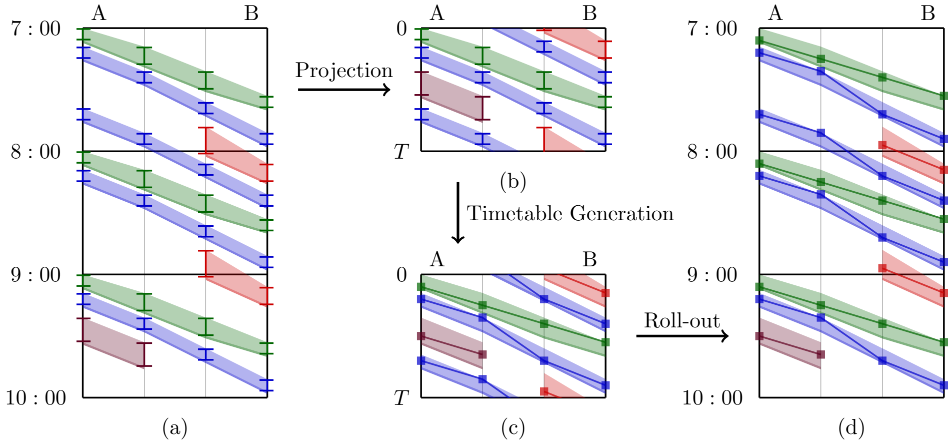

Figure 1: Illustration of the partial periodic timetable generation process

visualised in transparent blue. Similar bounds can be computed by propagating the trip time constraints from the B slot backwards. This gives two upper and lower bounds for each P and the feasible time slot in-between (dark blue area and blue intervals).

Figure 2b shows a case with three time slots. As each constraint is propagated forward and backward, three upper and three lower bounds can be computed for each P. The feasible event times lie between the highest lower bound and the lowest upper bound. It may happen that the defined slots (green bars) get restricted (blue intervals) through the propagation procedure.

Similarly, slots are propagated between the trains, using connection and dependency constraints, in both directions. Note that the resulting time slots always consist of one contiguous interval. If intervals are empty, it implies infeasibility of the pSI.

We make the following assumption for time slot lengths after propagation.

Assumption 1 (Maximum Time Slot Length) $\Delta\omega_i + \Delta\theta_{ij} < T \qquad \forall i, (i, j)$

where $\Delta\omega_i = \omega_k^+ - \omega_k^-$ is the length of the time slot at event i and $\Delta\theta_{ij} = \theta_k^+ - \theta_k^-$ is the span of the constraint between i and j, in both directions. Note that the assumption together with $\Delta\theta_{ij} \geq 0$ implies $\Delta\omega_i < T$. The assumption shall be satisfied for each event and each incoming and outgoing constraint.

The necessity of Assumption (1) is explained in Section 4.5. If the assumption is violated, the procedure is stopped. To continue, the $\Delta\omega_i$ may be reduced by defining additional or stronger time constraints in the pSI, or by choosing a lager T.

4.2 Projection

We explain here, how a pSI with partial periodicity can be transformed into a periodic timetabling problem. Figure 1a shows the morning hours of a pSI. Four train types (different colors) run on a track from A to B, visualised by the time slots obtained through constraint propagation. The Projection step leads to a periodic projection of the problem (Figure 1 b), having in this example the length $T = 60'$. It contains the trains from the original

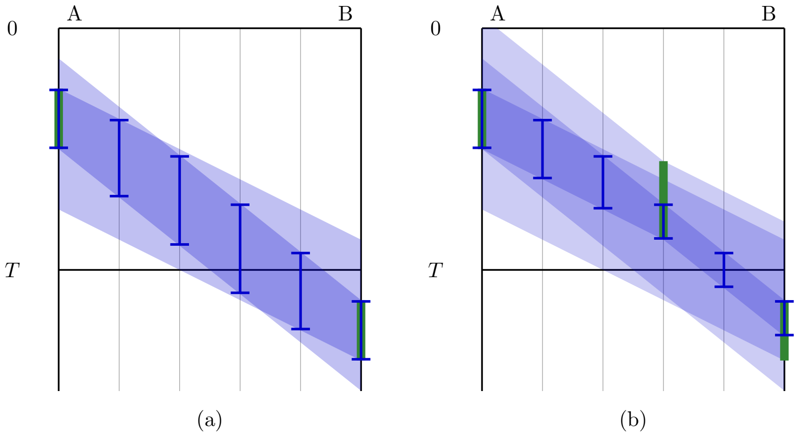

Figure 2: Slot propagation

problem with modulo T projected event slots. The projections of trains with periodicity T are identical, so one representative is sufficient. This reduces the number of trains in the periodic problem considerably.

More formally, we introduce equivalence classes for train runs. Runs belong to the same equivalence class if their schedules are identical except for a temporal shift of an integer multiple of T. This is the case if a train has a periodicity $\rho = Tk$ with integer $k \in \mathbb{N}$.

If the periodicity of a pSI is such that the run happens λ times within Tk then λ equivalence classes are needed (if the train is repeated at least λ times). If the expression

$$\frac{\rho}{T} = \frac{k}{\lambda} \qquad k, \lambda \in \mathbb{N}.$$

is rational and k, λ are chosen such that k/λ is irreducible, then λ is the number of equivalence classes required. For instance, if $\rho = 40'$ and $T = 60'$, then $\lambda = 3$ and $k = 2$.

The projected periodic problem has length T and contains one train run of each equivalence class, see Figure 1 for an example. The green train runs exactly once per hour and is therefore represented once in the projection. The blue train runs once per hour but with additional half-hour periodicity in the morning rush hour, leading to two representatives in the projected periodic problem.

The projected Problem is of PESP form. The PESP nodes are the projected events of the equivalence classes. The PESP constraints are the projected trip time, connection and dependency constraints. Constraints between two equivalence classes are introduced if they occur at least once between the corresponding equivalence classes in the original problem. E.g., if a connection between two hourly trains is only defined in the pSI to happen every second hour, it is equivalent to require it for every hour as the other will take place in any case.

If the periodicity of a train is smaller than T, multiple equivalence classes for this train become necessary. The required fixed time shift between them is enforced by using periodicity constraints. E.g., $T = 60'$ and $\rho = 30'$ results in $\lambda = 2$ required equivalence classes. Introducing periodicity constraints asserts the temporal difference of exactly $30'$ between the two.

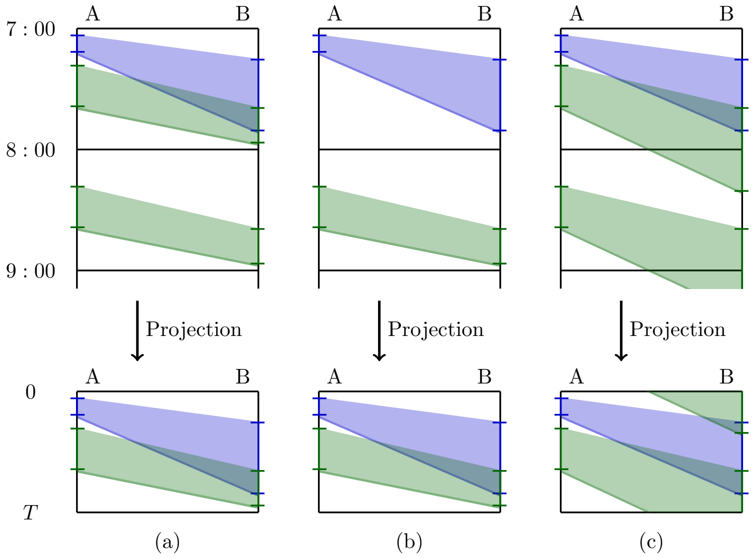

Figure 3: Necessity of headway constraints

Somewhat more complicated is the case for headway constraints.They guarantee that two trains do not collide (on single-track with opposing directions), overtake or get too close. Headway constraints between equivalence classes are necessary if their time slots overlap in the original problem and are always introduced for two trains on a common track section. Multiple trains are treated by considering each pair separately. It is therefore sufficient to discuss the possible cases for a pair of trains on one track section between event nodes.

For each such train pair, one has to check whether the event time slots of the two trains get closer than the required headway distance. For this purpose, a safety zone with length of the required headway is added to each slot. If these slots with safety zones overlap, the trains may possibly get too close, and headway constraints need to be introduced.

The procedure to determine the necessary headway constraints is described in Algorithm 1. The possible cases for a pair of trains on a common track section are described here and are illustrated in Figure 3.

In the trivial case where the slots of two projected equivalence classes do not overlap, there is no need for headway constraints. For any combination of trajectories inside the slots, the trains run at a safe distance. This is the case in the example of Figure 1.

If the slots belonging to two equivalence classes overlap in both projected and original problem, headway constraints are necessary to avoid potential collisions (Figure 3a).

If the slots do overlap in the projected case but the corresponding slots in the original problem are not overlapping, headway constraints should not be introduced (Figure 3b). As the slots are sufficiently separated in the original problem, no headway constraints are necessary here. It is even better to omit headway constraints, as their introduction leads to an unnecessary reduction of solution possibilities. E.g., the green and blue trains may

Algorithm 1 Checking necessity of headway constraints

```
for each pair of projected trains (equivalence classes) do
  determine common track sections
  for each common track section do
    find overlapping time slots in the projected period. Slots are considered to be overlapping if they violate the minimum required headway distance
    {maximum $4$ overlappings possible, 2 per track section end}
    $headway\_necessary = false$
    $headway\_ok = true$
    for each overlapping do
      check in the original problem if the overlapping also occurs there
      if overlapping also in original problem then
        $headway\_necessary = true$
      else
        $headway\_ok = false$
      end if
    end for
    if $headway\_necessary = true$ and $headway\_ok = true$ then
      introduce headway constraints
    else if $headway\_necessary = false$ and $headway\_ok = false$ then
      no headway constraints
    else if $headway\_necessary = true$ and $headway\_ok = false$ then
      introduce headway constraints {warning: equivalence assumption violated}
    else
      no headway constraints {headway constraints have no consequences here as trains cannot conflict}
    end if
  end for
end for
```

without conflicts have the same periodic arrival time in B, which would be prohibited by a headway constraint.

The projected case can have four overlappings, two at each end of the track section. In some cases it may happen that one overlapping requires headway constraints while another one implies that they should not be introduced (Figure 3c). The safety aspect requires to introduce headway constraints despite of the possible loss of solution space. In this case, the equivalence between original problem and projected problem would be lost (see Section 4.5 for details).

Assumption 2 (Headway Special Case) *The case (headway_necessary = true) and (headway_ok = false) of Algorithm 1 does not occur (see Figure 3c).*

This assumption is necessary for the equivalence of the periodic problem. If it is violated, a larger T or reduced time slots in the pSI can be used to overcome the problem.

4.3 Periodic Timetable Generation

The projected periodic problem with trip time, connection, dependency, periodicity and headway constraints can then be solved by using the PESP formulation, described in Section 3, resulting in event times for the equivalence classes (see Figure 1c).

Additional to the constraints discussed above, the time windows computed in the slot propagation step are included into the PESP formulation. The start and end times of the slots are projected to the periodic problem (modulo T) and introduced as lower and upper bounds of constraints pointing from an artificial zero-node to the events.

The objective function used for solving the PESP can basically be the same as in the original problem, depending linearly on the length of trip times or connections etc. In order to be equivalent, it needs to include weights taking account of the number of repetitions of a trip time or connection in the original case.

4.4 Rolling - Out

The periodic timetable generated as a solution to the projected periodic problem can be rolled-out to a partial periodic timetable for the entire day (Figure 1d).

The PESP solution consists of event times for each train in the projected problem. These trains are representatives of an equivalence class of trains in the original problem, being identical except for a temporal shift of kT. It is therefore sufficient to re-transform the event times of an equivalence class to the the corresponding time slots of the original problem. Note that Assumption 1 implies for the time slot size $\Delta\omega < T$, which makes the correspondence between original and projected problem unique.

The result is a timetable consisting of event times for each train run described in the pSI and fulfilling all required constraints.

4.5 Equivalence

We have presented a procedure to generate partial periodic timetables form a pSI by solving a periodic problem. We show in the following that the original partial periodic and the projected periodic problems are equivalent, if Assumptions 1 and 2 hold.

The rolled-out solution to the periodic problem fulfils the trip time, connection and dependency constraints defined in the pSI. We show here that the PESP constraints (5) are equivalent to the pSI constraints

$$\theta^- \leq \omega_j - \omega_i \leq \theta^+$$

under Assumption 1. We apply Eq. (5) to the pSI data $l_{ij} = \theta^-$, $u_{ij} = \theta^+$, $\pi_i = \omega_i - k_i T$, $\pi_j = \omega_j - k_j T$, with $k_i, k_j \in \mathbb{Z}$. It results in

$$\theta^- \leq \omega_j - k_j T - \omega_i - k_i T + p_{ij} T \leq \theta^+$$

$$\theta^- \leq \omega_j - \omega_i + (p_{ij} - k_i - k_j) T \leq \theta^+$$

which is equivalent to the pSI constraint for the integer valued expression $p_{ij} - k_i - k_j = 0$. We show that this is the case if the assumption holds. From slot propagation we have the following event time bounds:

$$\omega_i^- \leq \omega_i \leq \omega_i^+$$

$$\omega_i^- + \theta^- \leq \omega_j^- \leq \omega_j \leq \omega_j^+ \leq \omega_i^+ + \theta^+.$$

This leads to the bounds

$$(p_{ij} - k_i - k_j)T \leq \theta^+ - \omega_j + \omega_i \leq \theta^+ - (\omega_i^- + \theta^-) + \omega_i^+ = \Delta\omega_i + \Delta\theta_{ij},$$

and

$$(p_{ij} - k_i - k_j)T \geq \theta^- - \omega_j + \omega_i \geq \theta^- - (\omega_i^+ + \theta^+) + \omega_i^- = -(\Delta\omega_i + \Delta\theta_{ij}).$$

Combining both, we get

$$-\frac{\Delta\omega_i + \Delta\theta_{ij}}{T} \leq (p_{ij} - k_i - k_j) \leq \frac{\Delta\omega_i + \Delta\theta_{ij}}{T}.$$

It follows, together with the integrality of the expression, that $p_{ij} - k_i - k_j = 0$. Therefore, under the Assumption 1, $\Delta\omega_i + \Delta\theta_{ij} < T$, the equivalence of the PESP to the pSI constraint is shown.

The equivalence of the headway constraints was motivated in Section 4.2. Assumption 2 excludes the critical case where the equivalence cannot hold and solutions may be lost. If no assumptions are violated and a solution to the projected problem is found, then it fulfils all the constraints from the original problem. Conversely, if no solution to the PESP is found and the assumptions hold, it means that the pSI defines an infeasible problem. Both assumptions can be satisfied by increasing the value of T, which is discussed next.

4.6 Choice of T

As the projection is possible for an arbitrary value of projection period T, the question arises of which is the optimal choice. It has to satisfy the required assumptions and should reduce the problem size optimally.

Assumption 1 depends directly on T and can easily be satisfied by increasing its value. Assumption 2 also depends on T, but in a more complicated way. Doubling the value from T to $2T$ certainly preserves the satisfaction. This is not generally true for values in-between.

Increasing T generally leads to satisfaction of the assumptions. In particular, it can always be found a T_{max} large enough to contain all time slot of the original problem and fulfilling both assumptions.

The choice of a small T is of course more efficient. If the pSI contains some periodic structure, the number of equivalence classes and therefore the size of the projected periodic problem shrinks if T divides the periodicities ρ_i of many trains. Using $T = T_{\text{max}}$ leads to a projected problem with one equivalence class per train. This is equivalent to a non-periodic task-scheduling problem without any reduction of problem size.

Usually, the offer of a railway company has a basic periodicity of half an hour, one hour or two hours. It is the natural choice to use these values for T. Computational results for various choices of T are given in the next section.

5 Computational Results

A test case was set up to validate the concepts and algorithms of this work. The scenario includes the cities of Lucerne, Zug and Arth-Goldau as the major nodes in the network, as illustrated in Figure 4. The periodic service intention is reverse-engineered from the 2008 SBB timetable for a standard Wednesday. It contains Intercity trains running from Baar (in direction Zurich) and Sursee (Basel) to Erstfeld and the Gotthard tunnel through the Alps. Additionally, there are Interregional trains running from Lucerne to Baar and to Biberbrugg (St. Gallen). Regional trains run in the triangle Lucerne – Zug – Arth-Goldau with several stops in between, as well as on all other lines in the considered network.

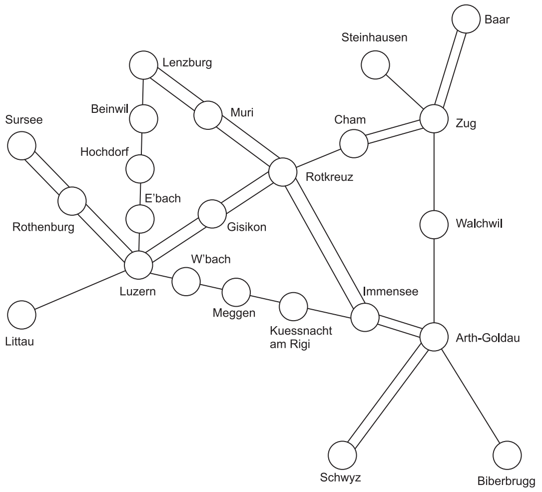

Figure 4: The test case region connecting the towns Zug – Lucerne – Arth-Goldau in central Switzerland. The track topology is partly double track and partly single track and is used by regional and intercity trains as well as freight trains. All the events in the pSI and in the PESP model correspond to departure or arrival times at stations in this picture.

The code is written in Java, and the resulting ILP is then solved with the commercial solver ILOG CPLEX 11. The tests were conducted on a x86-64 machine (Dual-Core AMD Opteron(tm), 4 GB RAM). The projection algorithm works almost instantaneously for all tested scenarios. For all computations we have used an objective function which minimises the sum over all passenger-relevant travel times: trip-, dwell-, and connection times.

Table 1 presents the results for the pSI of the test case. Furthermore, in order to better compare the additional complexity of the partial periodic case, a version of the test case with only the regular trains is also tested, omitting all the exception that occur. This version results in a fully periodic pSI and would have been the input of classical approaches for periodic timetabling. The resulting optimal timetable would then need some (manual or automatic) postprocessing to adapt it for each period in the day to fulfil the requirements of the pSI. This adaptation can also have negative effects on the timetable quality, as the exceptions are not considered during the optimisation phase. This postprocessing phase is not considered here. We have tested the algorithm for two different values of the time period T, as well as without the projection ($T = T_{max}$), considering the whole day as a single period.

One can first observe that, depending on the chosen time period T for the projection, the generated (equivalent) PESP instances have different sizes. As most trains in the test scenario have a periodicity of one hour, the projection with $T = 60'$ results in a PESP

Scenario	T	# variables	# int. variables	# constraint	CPU time $[s]$
Partial periodic pSI	60	2206	963	3449	24*
	120	2936	1278	4594	23
	no	12 122	5168	19 076	130
Fully periodic	60	802	341	1263	6
	120	1344	568	2120	2
	no	10 732	4544	16 920	70

Table 1: Statistics of the generated PESP instance and solution time of the ILP optimisation for two different scenarios and some values of T. The asterisk * means that the equivalence assumption was violated, resulting in an infeasible instance.

instance of smaller size. On the other hand, the smaller the value T the smaller the time slots for the events should be as well. For instance, the initial time slots specified in the pSI are relatively large, and choosing $T = 60'$ leads to a violation of the equivalence Assumption 2, eventually resulting in an infeasible PESP instance. In the case of $T = 120'$, the projection is possible by keeping the equivalence of the problem, and an optimal solution is computed in less than half a minute. For comparison, solving the same problem without making use of the projection and the equivalence classes (i.e. choosing $T = T_{max}$) one will have an ILP of (much) larger size and it will take much longer to find an optimal solution. An optimal solution to this scenario is then rolled out for a complete day and results directly in a schedule fulfilling the pSI with its partial periodicity.

The fully periodic scenario can be projected to a PESP instance of smaller size, as all special trains are not considered here. The smaller T is, the larger the difference, because regular trains need less equivalence classes than special train runs occurring only once. In the non-projected version the difference between the partial periodic and the fully periodic PESP instances is small, as periodic trains are repeated often, whereas special trains occur only few times. Fully periodic instances can be solved 2-10 times quicker, but need a postprocessing than can also be time consuming, and the solution quality might deteriorate considerably.

The generally good computation times compared to the purely periodic case can be partially explained with the structure of the problem. Composing a pSI requires planning that already determines some structure of the timetables, thus limiting the search space of the solver. However, the main complexity of timetabling remains unchanged, consisting, among other aspects, of finding feasible train orderings on the tracks.

Summarising, the best value of T for solving Problem 5 depends on the pSI but can be roughly determined as the smallest possible periodicity of the train runs where the equivalence assumption is not violated. Directly considering the pSI with its partial periodicity for the optimisation problem leads to longer computation times, yet avoiding its explosion, and are largely compensated by the fact that no more postprocessing is necessary and all requirements in the pSI are taken into consideration for the optimisation, which results in timetables of better quality.

6 Conclusion and Outlook

We have presented a new approach how the commercial offer of train services can be formally described and used for timetable generation. In particular, periodicity is nowadays an important part of the service and must be considered in the preparation of the commercial offer. We have therefore introduced the periodic Service Intention (pSI), a framework in which the customer-relevant information about train services can be specified, including their periodicity information. Such a pSI seems a more realistic description of the necessities of a railway timetable over a whole day than previous approaches based on pure periodicity, as the transport demand is not constant over the day. It permits the development and analysis of commercial railway offers that are not completely periodic but contain some periodic structure.

We solve the timetable generation problem by projecting intended train runs over equivalence classes and thereby reducing the pSI to a periodic timetabling instance while keeping the problem equivalent. This partial periodic structure can be exploited effectively, as each equivalence class is represented only once in the projected problem. It is then possible to use existing well-known models for periodic scheduling, and also take advantage of future improvements in this field. The stronger the periodicity of the offer, the larger the reduction of the problem size and shorter the computation times. The proposed model and approach is therefore particularly well suited for offers with a strong periodicity but some irregularities, which could not be treated properly by the existing periodic timetable generators. The pSI is a general framework which generalises previous service descriptions, and it is therefore able to handle any periodic structure of the intended service, including the extremes of no periodicity and full periodicity.

Adding the scheduling on the micro level is the next logical step of this method, which we are currently working on. Existing methods for micro scheduling cannot be directly applied, because the equivalence classes are not the same as a single train run, and two trains could use the same resource simultaneously in the projected problem, yet remaining feasible because they are temporally separated in the original problem. Adaptations to cope with these new features are therefore necessary.

In future work, the presented approach should be compared to classic approaches with manual postprocessing in order to analyse computation times and quality of the generated timetables. Intensive discussions with planners are also necessary to make the pSI an effective instrument for describing service intentions in practice.

References

[1] G. Caimi, M. Fuchsberger, M. Laumanns, and K. Schüpbach. Periodic railway timetabling with event flexibility. In C. Liebchen, R.K. Ahuja, and J.A. Mesa, editors, *ATMOS 2007*. IBFI, Schloss Dagstuhl, Germany, 2007.

[2] G. Caimi, T. Herrmann, D. Burkolter, F. Chudak, and M. Laumanns. Design of a railway scheduling model for dense services. *Networks and Spatial Economy*, March 2009. to appear.

[3] A. Caprara, M. Fischetti, and P. Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50(5):851–861, 2002.

[4] I.A. Hansen and J. Pachl, editors. *Railway, Timetable & Traffic*. Eurailpress, 2008.

[5] L. Kroon. Fahrplanoptimierung mit mathematischen Modellen bei den Niederländischen Eisenbahnen. *Eisenbahntechnische Rundschau*, 6:359–362, 2008. In German.

[6] F. Laube and V. Mahadevan. 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, 2008.

[7] F. Laube, S. Roos, R. Wüst, M. Lüthi, and U. Weidmann. PULS 90 - ein systemumfassender Ansatz zur Leistungssteigerung von Eisenbahnnetzen. *Eisenbahntechnische Rundschau*, 3:104–107, 2007. In German.

[8] C. Liebchen. *Periodic Timetable Optimization in Public Transport*. PhD thesis, Technische Universität Berlin, 2006.

[9] C. Liebchen. The first optimized railway timetable in practice. *Transportation Science*, 42(6), 2008.

[10] C. Liebchen and R. Möhring. The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables - and Beyond. In F. Geraets et al., editors, *Algorithmic Methods for Railway Optimization*, LNCS 4359, pages 3–40. Springer, 2007.

[11] C. Liebchen and L. Peeters. On cyclic timetabling and cycles in graphs. Technical Report 761-2002, TU Berlin, Department of Mathematics, Combinatorial Optimization and Graph Algorithms Group, 2002.

[12] C. Liebchen and R. Rizzi. Classes of cycle bases. *Discrete Applied Mathematics*, 155(3):337–355, 2007.

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

[14] M. Lüthi, A. Nash, U. Weidmann, F. Laube, and R. Wüst. Increasing railway capacity and reliability through integrated real-time rescheduling. In *Proceedings of the 11th World Conference on Transport Research, Berkeley*, 2007.

[15] K. Nachtigall. Periodic Network Optimization and Fixed Interval Timetables. Habilitation Thesis, University Hildesheim, 1998.

[16] K. Nachtigall and S. Voget. A genetic algorithm approach to periodic railway synchronization. *Computers & OR*, 23(5):453–463, 1996.

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

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

[19] L.W.P. Peeters. *Cyclic Railway Timetable Optimization*. PhD thesis, Erasmus University Rotterdam, 2003.

[20] G. Sahin, R. K. Ahuja, and C. B. Cunha. New approaches for the train dispatching problem. *submitted to Transportation Research Part B*, 2005.

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

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

[23] R. Wüst, F. Laube, S. Roos, and G. Caimi. Sustainable Global Service Intention as objective for controlling Railway Network Operations in Real Time. In *Proceedings of the 8th World Congress of Railway Research (WCRR)*, Seoul, Korea, 2008.