Resolving instability in railway timetabling problems

Bešinović, Nikola; Quaglietta, Egidio; Goverde, Rob M.P.

DOI
10.1007/s13676-019-00148-3

Publication date
2019

Document Version
Final published version

Published in
EURO Journal on Transportation and Logistics

Citation (APA)
Bešinović, N., Quaglietta, E., & Goverde, R. M. P. (2019). Resolving instability in railway timetabling problems. *EURO Journal on Transportation and Logistics*, *8*(5), 833-861. https://doi.org/10.1007/s13676-019-00148-3

EURO J Transp Logist
https://doi.org/10.1007/s13676-019-00148-3

RESEARCH PAPER

Resolving instability in railway timetabling problems

Nikola Bešinović[1] · Egidio Quaglietta[1,2] · Rob M. P. Goverde[1]

Received: 15 July 2017 / Accepted: 22 October 2019
© The Author(s) 2019

Abstract
A significant growth of the railway transportation demand is forecasted in the next decades which needs an increase of network capacity. Where possible, infrastructure upgrading can provide extra capacity; although in some cases, this is not enough to satisfy the entire transportation demand even if optimised timetabling is performed. We propose a heuristic model to develop a stable timetable which maximises the satisfaction of transportation demand in situations where network capacity is limited. In case the demand cannot be fully satisfied, the model relaxes the given line plan and timetable design parameters. The aim is to keep as many train services as possible and reduce the level of service minimally. We develop a mixed integer linear programming (MILP) model for minimising the cycle time to find an optimised stable timetable for the given line plan. The heuristic iteratively solves the MILP model and applies relaxation measures. We tested the model on the Dutch network. The results showed that the model can generate stable timetables by removing train services from the critical circuit, and also, higher transportation demand can be satisfied by additionally relaxing timetable design parameters.

Keywords Timetabling · Periodic event scheduling problem (PESP) · Instability · Minimum cycle time

1 Introduction

A significant increase in the demand of passenger and freight transportation is expected to load railway networks in the near future. In this context, infrastructure managers strive to find operational and/or infrastructural solutions to allow higher traffic volumes running on the network to satisfy the forecasted transportation demand. The set of train services designed to meet such expected transportation demand is called the target line plan. To meet the demand, railway planners have the objective to

✉ Nikola Bešinović
n.besinovic@tudelft.nl

1 Department of Transport and Planning, Delft University of Technology, Delft, The Netherlands

2 Control, Command and Signalling, Network Rail, London, UK

design a timetable that possibly runs all trains of the target line plan within a scheduled cycle time T, usually coinciding with one hour. In railway planning, it is usual to have train services repeating every cycle T in a so-called periodic timetable. Network capacity is, however, not always sufficient and infrastructure upgrades are often necessary to accommodate a denser train service plan. The possibility of installing additional tracks, platforms and/or flyovers in bottleneck areas such as stations and junctions, is mostly restricted by budget constraints and the lack of physical space, especially for stations and junctions located in densely built urban areas. A more convenient and sustainable alternative would instead be upgrading the conventional signalling system with technologies suitable for running high-capacity high-speed (HCHS) railway traffic, such as the European Train Control System (ETCS) (Stanley 2011). For instance, the UK railway infrastructure manager Network Rail is currently opting for these types of improvements with the delivery of the Digital Railway programme Digital Railway Programme (2016). The Digital Railway aims at deploying advanced technologies in the area of control, command and signalling (e.g. optimal timetabling tools, driver advisory systems, traffic management systems, automatic train operation, ETCS signalling) to meet a 40% increase in transportation demand forecasted to load the UK network within the next 30 years.

Installation of advanced control and signalling systems, however, is not going to be sufficient if train services are not planned effectively to maximise the utilisation of the additional capacity enabled by these new technologies. In this context, enhanced timetabling models are necessary to be applied to lay out the target line plan smoothly on the network under the control of advanced HCHS control and signalling technologies. In case the target line plan is incompatible with the residual capacity of the network (i.e. not all planned train services can actually run on the network within cycle time T), such timetabling models shall be able to fit in as many train lines as possible while providing efficient train operations and a mitigation of delay propagation. In other words, timetabling shall focus on maximising utilisation of infrastructure capacity while providing robustness to stochastic disturbances. We define the minimum cycle time λ as the minimum amount of time over which all train events (i.e. arrivals, departures, passings) in the target line plan can be scheduled without conflicts. According to Heidergott et al. (2005) and Goverde (2007), network-level capacity occupation (stability) of a periodic timetable can be expressed by the minimum cycle time λ of the timetable. A timetable that satisfies the condition $\lambda < T$ is called structurally stable Goverde (2007). Planners always aim to identify a stable timetable which can accommodate the entire target line plan with the scheduled cycle time so as to satisfy the forecasted demand. Instead, the question is how to tackle scheduling problems when a stable timetable cannot be found because the increased demand is higher than the additional capacity gained with the deployed enhanced control and signalling systems. In such a situation, additional timetabling solutions and/or measures shall be considered so as to minimise penalties for partially satisfying the transportation demand, i.e. minimise deviations from the target line plan. Therefore, is it necessary to cancel some of the train services in the target plan? If needed, how to cancel services while minimising the impact on level of service?

In this paper, we propose a mathematical model for resolving timetable instability $\lambda \geq T$ caused by an overly ambitious target line plan. In particular, the model finds an optimised stable timetable (structure) that satisfies the transportation demand as much as possible. To do so, a heuristic approach has been developed that integrates an optimisation model and relaxation rules to minimise necessary corrections to the target line plan. A timetable structure is defined as an ordered sequence of train event times on a railway network during a basic period that provides feasible operations with respect to the minimum process times (e.g. running, dwell, turnaround times). In essence, it can be represented by a compressed timetable with $T = \lambda$. We design a mixed integer linear programming (MILP) model for solving a timetabling problem that minimises the cycle time λ for a given line plan. This MILP model finds an optimal timetable structure, which uses the network capacity minimally. If the optimal timetable structure is unstable, $\lambda > T$, then the line plan is relaxed. Three relaxation measures are proposed to adjust the target line plan and timetable design parameters. The latter include relaxing regularity constraints and relaxing train-related constraints, and the former reducing line frequencies. The iterations between the minimal cycle time optimisation and relaxations are repeated until the optimised stable timetable structure has been found. When a stable timetable (structure) is obtained, time allowances can be optimally allocated to maximise robustness versus stochastic disturbances. An application of the proposed approach is performed for a part of the Dutch railway network and for a forecasted increased transportation demand. Results show that the presented model produces a stable timetable that minimise impacts on the transportation demand to satisfy.

The main contribution of the paper is that it presents the first timetabling model that tackles timetable instability of periodic timetables in (over)saturated networks. We do not assume that all train lines from the given line plan can be scheduled. This makes the model more general, provides more flexibility to find a stable solution and allows a wider application in railway timetable planning. In particular, the model could be used in dense networks where the capacity use (i.e. minimum cycle time) is already becoming critical, $\lambda \approx T$; also, the model could evaluate infrastructure improvements projects. Second, the proposed procedure for computing an initial solution and stronger upper bound for λ in MILP significantly reduced its computation time. Third, natural measures to relax the given line plan were successfully implemented and contributed to creating stable timetable structures. Fourth, the algorithm for resolving instability shows the importance of relaxing train lines that are part of the critical circuit (consisting of the critical events and processes defining the minimum cycle time λ) opposed to relaxing random train lines. These extensions to timetabling model make it a very useful support tool for timetable planning in areas with high demand and/or scarce infrastructure capacity and can help in finding a maximal set of train lines and corresponding frequencies that satisfies (most of) the transportation demand.

In Sect. 2, we give a literature review on timetabling models and stability-related research. Section 3 introduces the periodic event scheduling problem (PESP) and the PESP-based model for minimising cycle time, PESP-λ. Section 4 defines first the assumptions and priority rules and then, the heuristic algorithm for resolving instability and improvements for PESP-λ. Section 5 gives the computation results of the proposed heuristic for two test scenarios on the Dutch network. Finally, Sect. 6 gives conclusions and main observations.

2 Literature review

A target line plan defines a set of requested train lines with its origin and destination, stopping stations and its frequency in number of train services per scheduled cycle time. A timetable consists of event times such as arrival and departure times in stations and processes between events like running, dwelling and transfer times. Process times also include infrastructure constraints (i.e. headways) between events that guarantee safe operations. In periodic timetabling, a cycle time T is given and all events are selected in the interval between 0 and T.

Timetable feasibility is the ability that a timetable exists for a given line plan without violating any train- and passenger- and safety constraints. Timetable stability is the ability of a timetable to absorb initial and primary delays, so that delayed trains return to their scheduled train paths without rescheduling. Timetable efficiency is the ability to run trains as fast as possible and thus allocate only limited running and dwell time supplements.

Aiming at timetabling efficiency may create a significant speed difference between train lines of different types. For example, if two train lines, a fast and slow (e.g. intercity and local), on a corridor are scheduled with minimal time supplements, then the speed difference between the two will be significant and the two trains together would need more infrastructure capacity. On the other hand, if a faster train is allowed to run slower and thus, allow a more homogeneous service, then less capacity will be used (Hansen and Pachl 2014).

Figure 1 shows the influence of heterogeneity on the minimum cycle time. It gives the minimum cycle times of two train lines with a frequency of two in a period of length T running in the same direction over a single track. The dashed line is the first train repeated in the next period. Figure 1a, b represents heterogeneous (more homogeneous) services with minimum cycle time λ_{ht} (λ_{ho}). The running time difference between train services in Fig. 1a is evident, while the running times of the fast services in Fig. 1b are extended and more similar to the one of the slower services. Clearly, λ_{ho} is smaller than λ_{ht} due to smaller necessary headways between train services.

Periodic railway timetabling problems are often presented as a periodic event scheduling problem (PESP) introduced by Serafini and Ukovich (1989). Afterwards, a significant amount of research has been assigned to solving timetabling based on PESP. For solving PESP, Schrijver and Steenbeek (1994) applied constraint programming to find a feasible timetable, while Kümmling et al. (2015) used SAT solvers to the same problem. By adding an objective function to the PESP formulation, the timetabling problem can be solved using mixed integer programming (MIP) techniques as elaborated in Peeters (2003). Other papers that further developed PESP-based models for timetabling are Caimi et al. (2011), Kroon and Peeters (2003), Kroon et al. (2013), Liebchen (2009), Nachtigall (1993) and Nachtigall and Opitz (2008). In addition, Cacchiani and Toth (2012) give an overview of railway timetabling models.

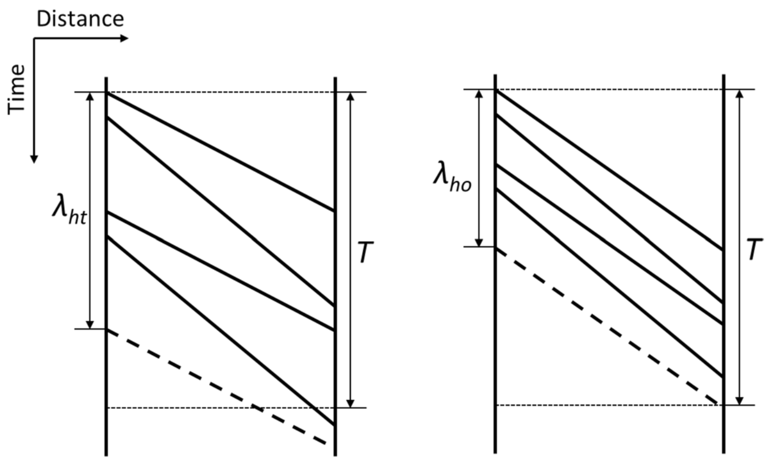

Fig. 1 Effect of speed on the minimum cycle time

Most of the models for solving timetabling problems (PESP) assume that it is possible to schedule all services from the line plan. Only recently, Kümmling et al. (2015), Polinder (2015) and Bešinović et al. (2016) presented approaches for solving problems of local infeasibility in periodic timetabling problems. Polinder (2015) resolves conflicts reported in the planning model DONS Schrijver and Steenbeek (1994) by relaxing neighbouring processes. Kümmling et al. (2015) proposed a similar approach to support the planning model TAKT (Kümmling et al. 2015). Differently, Bešinović et al. (2016) proposed an iterative micro–macro approach for designing (microscopically) conflict-free, stable and robust timetables. The macro-model computes a timetable, which is evaluated on microscopic conflict-freeness and (local) stability. Here, stability is defined at the local (station or corridor) level as a minimum necessary amount of buffer times. Still, Kümmling et al. (2015), Polinder (2015) and Bešinović et al. (2016) assume that all train services will be possible to schedule after applying small adjustments to the process times. Thus, more general timetabling models for dealing with potential instability should be considered.

The idea of minimising the cycle time for measuring stability was introduced by Bergmann (1975) for a single-track line and homogeneous fleet. Heydar et al. (2013) extended this model to a single-track unidirectional line that adheres to a cyclic timetable and considered two types of trains. The objective was to minimise the capacity occupation and minimise the total dwelling time of local trains at all stations. Petering et al. (2015) extended the model of Heydar et al. (2013) to allow selection of stop platforms in a station and schedule train overtakings. Sparing and Goverde (2017) developed an extension to the PESP model that minimises the cycle time and train running times which is applicable to both lines and networks. Zhang and Nie (2016) further expanded Sparing and Goverde (2017) by adding flexible overtaking constraints and heuristics to speed up the computations. In addition, authors analysed the effect of some timetable design parameters on the minimum cycle time. Output of these models is an (near) optimal timetable structure, i.e. a compressed timetable and not a final one.

To translate a timetable structure to an actual timetable, Bešinović and Goverde (2016) proposed a two-stage model for computing a stable and robust timetable. In particular, the first stage solved the minimal cycle time problem, while the second stage distributed time allowances to improve the timetable robustness. In addition, several objective functions were proposed and tested for the second stage.

The concept of timetable stability is often unrecognised in the literature on railway timetabling which is partly due to the common assumption that a given line plan naturally provides a stable timetable structure. Therefore, it is crucial to make a distinction between timetable stability and timetable feasibility. In general, a model for minimising the cycle time λ finds an optimal timetable structure to a given input. Such timetable structure is unstable if $\lambda > T$. If $\lambda = T$, the timetable is called critical (Goverde 2007). Any T-feasible timetable (i.e. feasible with cycle time T) satisfies $\lambda \leq T$, so it may be stable or critical. λ is the smallest T for which a timetable (structure) is T-feasible. Each periodic timetabling problem becomes T-feasible for a sufficiently large T. To make a timetable that can be operated, λ has to be smaller than desired T which could be done only by relaxing certain input constraints. In this paper, we extend the approach of Sparing and Goverde (2017) and define a more general approach to resolve timetable instability and design optimised stable timetables. Finally, we limit ourselves to finding an optimised stable timetable structure, while the final timetable can be generated by applying the approach from Bešinović and Goverde (2016).

3 Model formulation

3.1 Periodic event scheduling problem (PESP)

The timetabling approach is based on a periodic event-activity network (PEAN) represented by a weighted directed graph $N = (E, A, T, l, u)$, which is associated with a target line plan Q. A train line $q \in Q$ defines a requested periodic train service characterised by its origin and destination, stopping pattern and frequency f_q within a given scheduled cycle time T. The set E of events consists of periodic arrival, departure and pass-through events for each train line in Q in each station along its route. This means that if an event i is scheduled at time π_i then it will also occur at times $\pi_i + k \cdot T$ for $k = 1, 2, \ldots$ For each event, we determine the event time in the basic period $\pi_i \in [0, T)$.

Set A represents processes $(i,j) \in A$, where i and j are two consecutive events and can interpret various rules and restrictions. Running times are the times needed for a train to run between two timetabling points. A lower bound l_{ij} for the running time represents the nominal running time, which is the minimum running time increased by a certain percentage to satisfy stochastic train behaviour. The upper bound u_{ij} is the maximum running time extension with respect to the passenger quality of service. The set of running processes is denoted as A_{run}. Dwell times are the durations of a train stop in a station. The minimum represent a time needed to board and alight the train, while the upper bound u_{ij} limits the waiting time for passengers. The set of dwell processes is denoted as A_{dwell}. A passenger connection is a transfer of passengers from a feeder to a connecting train in a station. The minimum transfer time l_{ij} defines the necessary time to alight from the first train, walk to the departure platform, and board the second train. A set of connection processes is denoted as A_{conn}. Safety constraints between two trains based on the given signalling system are defined as A_{infra}. Formally, these constraints can be written as:

$$\pi_j - \pi_i + z_{ij}T \in [l_{ij}, u_{ij}], \quad \forall (i,j) \in A = A_{\text{run}} \cup A_{\text{dwell}} \cup A_{\text{connection}} \cup A_{\text{infra}}.$$

The binary variable z_{ij} represents a modulo parameter that determines the order of events i and j within a period T for given bounds $[l_{ij}, u_{ij}]$ and equals 1 if $\pi_j < \pi_i$ or 0, otherwise. This binary property of z_{ij} holds assuming $l_{ij} \leq u_{ij}$, $0 \leq l_{ij} < T$ and $0 \leq u_{ij} - l_{ij} < T$. This constraint can also be written as $\pi_j - \pi_i \in [l_{ij}, u_{ij}]_T$.

We also define subsets of events and processes for each train line E_q and A_q, respectively. Each train line $q \in Q$ with $f_q = 1$ consists of a sequence of process times $a = (i,j)$, where i and j are two consecutive events. For $f_q > 1$, the train line q consists of f_q train services. In that case, events and processes of a train line q are replicated f_q times and q_k depicts the kth repetition of the train service in a basic period where $k = \{1, \dots, f_q\}$. Subsets of events i assigned to the train service q_k are defined as E_{q_k} and subsets of processes (i, j) as A_{q_k}. To secure the regular train services of the same line, meaning that services q_k are equally separated in time, we introduce regularity constraints A_{reg}. These constraints are defined between services of one line, where i and j are events of two following services. The time separation between two consecutive services is equal to T/f_q and can be written as:

$$\pi_j - \pi_i + z_{ij}T = T/f_q, \quad \forall q \in Q : f_q > 1, (i,j) \in A_{\text{reg}}.$$

Here, i and j are events of two consecutive services of the same train line. Figure 2 gives a small example of a periodic event-activity network with two trains stopping at a station.

PESP is originally a feasibility problem, and we adopt the common mathematical formulation as:

$$\text{(PESP)} \quad l_{ij} \leq \pi_j - \pi_i + z_{ij}T \leq u_{ij}, \quad \forall (i,j) \in A \tag{1}$$

$$0 \leq \pi_i \leq T - 1, \quad \forall i \in E \tag{2}$$

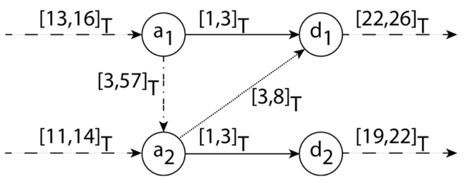

Fig. 2 An extract of a periodic event-activity network for two trains arriving (a) at and departing (d) from a station with running (dashed line), dwell (solid), transfer (dotted) and headway (dash-dotted) constraints

$$z_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \tag{3}$$

Constraint (1) defines bounds on the process times. Constraint (2) gives the periodicity of the events and schedules them in the interval [0, T). Constraint (3) defines the modulo constraint as a binary decision variable.

3.2 PESP for minimising cycle time

We aim at finding the optimal timetable structure over N by minimising the cycle time λ. To find the optimal timetable structure over N, we introduce the timetabling model that minimises the cycle time λ.

The difference between minimum and scheduled cycle time defines the available time allowances (time supplements and buffers). Therefore, the computed optimal timetable structure gives the train orders that use the infrastructure in the most optimal way and leave the most time allowance. Note that time allowances may be negative if $\lambda > T$. This means that although the model finds the optimal structure for the given line plan, it cannot be scheduled within the scheduled cycle time T, and thus, the timetable structure is unstable. The allocation of time allowances plays an important role in designing robust timetables and may depend on typical types of delays occurring in the network, e.g. primary delays caused by extra running/dwelling times and secondary delays propagated from other delayed trains. For distributing time allowances in a defined timetable structure, we refer to Sparing and Goverde (2017) and Bešinović and Goverde (2016).

The problem of finding the optimal timetable structure is formulated based on PESP and consists of solving the problem of minimising the cycle time. In addition, minimisation of journey times is used as a secondary objective term to prevent an excessive extension of journey times. The new MILP formulation of minimising the cycle time is then the following:

$$(\text{PESP} - \lambda) \quad \text{Minimise} \quad \lambda + \sum_{(i,j)\in A_{\text{run}} \cup A_{\text{dwell}}} \tau_{ij}(\pi_j - \pi_i + y_{ij}) \tag{4}$$

subject to

$$l_{ij} \leq \pi_j - \pi_i + y_{ij} \leq u_{ij}, \quad \forall (i,j) \in A \tag{5}$$

$$\pi_j - \pi_i + y_{ij} = \lambda / f_q, \quad \forall (i,j) \in A_{\text{reg}}, \forall q \in Q \tag{6}$$

$$0 \leq \pi_i \leq \lambda - 1, \quad \forall i \in E \tag{7}$$

$$0 \leq \lambda \leq \lambda_{\max}, \tag{8}$$

$$0 \leq y_{ij} \leq \lambda, \quad \forall (i,j) \in A \tag{9}$$

$$y_{ij} \geq \lambda - \lambda_{\max}(1 - z_{ij}), \quad \forall (i,j) \in A \tag{10}$$

$$y_{ij} \leq z_{ij}\lambda_{\max}, \quad \forall (i,j) \in A \tag{11}$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A \tag{12}$$

The objective function (4) represents minimising the cycle time and total journey times. Here, τ_{ij} defines a process-dependent weight that may differ for running and dwell processes. The weights are defined in such a way that the sum of weighted journey times is significantly smaller than λ to maintain the first term of the objective function dominant. In addition, the weights can reflect the practitioner's needs to prioritise running versus dwelling processes. Constraint (5) defines bounds on the process times. Constraint (6) synchronises train services of train lines. Constraint (7) sets the events in a periodic interval $[0, \lambda)$. Constraint (8) defines λ to be strictly positive and smaller than a given upper bound $\lambda_{\max}$. Since the scheduled cycle time T from (1) to (2) is substituted with a decision variable λ, the constraint (1) would become nonlinear because of the new nonlinear term $z_{ij}\lambda$. Hence, this is linearised by introducing new variables $y_{ij} = z_{ij}\lambda$ and constraints (9)–(11) according to Sparing and Goverde (2017). Here, $\lambda_{\max}$ is a suitable upper bound for the objective value λ. In the remainder of the paper, we refer to the model for minimising cycle time as PESP-λ. The output of PESP-λ is the minimum cycle time λ and the optimal timetable structure (π, z) where π_i are event times for all $i \in E$ and z_{ij} modulo parameters for each arc $(i,j) \in A$.

3.3 Critical circuit

As mentioned before, λ is a network stability measure. To identify processes that restrict having a smaller λ, we introduce additional terms. A circuit is a closed sequence of events in N. We focus only at elementary circuits, i.e. circuits in which each vertex (i.e. events) has exactly one incoming and outgoing arc. Note that we consider directed circuits. We refer to a circuit that builds the minimum cycle time as to a critical circuit C_λ. Determining the critical circuit is performed by the function *getCriticalCircuit*, which consists of three steps: (1) find all strongly connected components using a depth-first search algorithm, (2) compute all circuit times and corresponding cycle means (i.e. the sum of all processes in a circuit divided by the number of periods needed to perform all processes) over all components, and (3) assign the circuit with the biggest cycle mean as the critical circuit. Such computed biggest cycle mean corresponds to the minimum cycle time λ. Alternatively, the critical circuit can be computed using the policy iteration algorithm (Goverde 2007, 2010).

To determine the most constraining events in the network, we compute and analyse the critical circuit C_λ. If an event i is in C_λ, we refer to it as a critical event and include it in the set E_λ. Events on the critical circuit identify the critical processes A_λ in the network. In addition, we make a set of critical services Q_λ where q_k is in Q_λ if it has at least one event in E_λ. The corresponding processes in the critical circuit

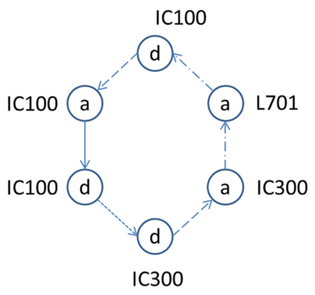

Fig. 3 Example of a critical circuit including three arrival (*a*) and three departure (*d*) events of three different train lines. (Line types are the same as in Fig. 2)

can be running, dwell, connection, regularity or headway processes. Figure 3 gives an example of a critical circuit that comprises six arrival and departure events over four train lines.

To obtain a stable timetable, the timetable must be operated with a scheduled cycle time $T > \lambda$. To tackle instability of the timetable, i.e. reduce λ, we need to make changes (at least) on the critical circuit C_λ and in particular, relax critical lines. Otherwise, relaxing a random train line, that is not being part of the critical circuit may not affect the stability of the whole system, and the value of λ would stay unchanged. For example, removing a train service in a low-dense network area, the events of which are not part of the critical circuit may result in unchanged (in)stability of the timetable.

4 Resolving timetable instability

4.1 Assumptions and priority rules

When relaxing the target line plan Q, we want to choose a most suitable train line (one or more) $q \in Q_\lambda$ to relax, which affects transportation demand the least. Thus, after discussions with planners, we introduce a set of priority rules and assumptions that allows to incorporate the unsatisfied demand implicitly. Such rules suggest which train line should be adjusted first and are based on the train line characteristics.

Before generating the rules, we make the underlying assumptions. First, all train lines of the same service type (i.e. stopping pattern) have the same passenger capacity between two stops and are treated as equally important. Second, as a consequence of the previous, overall transport capacity of a train line differs when changing the number of stations, i.e. a longer train line of the same service type transports more passengers; hence, it is more important to maintain train lines that stop at more stations. Third, intercity long-distance trains have higher capacity than regional trains. Fourth, each train line should be maintained in the timetable, meaning that the frequency for any given line $q \in Q$ may be relaxed at most to $f_q = 1$.

We select four train line characteristics for determining their priorities: the covered distance from origin to destination, line type, number of stops and line frequency. In general, the goal is to relax the train line that would affect less passengers in the network. Hence, we define four lexicographic rules that should be followed in the given order: first, choose the line type with less passenger demand; second, choose a line that covers the shortest distance; third, choose the line that has the least number of stops, and fourth, choose a line with the highest frequency. Finally, if two (or more) lines have all criteria equal, then the one is selected randomly.

These assumptions and priority rules could be easily extended or substituted with realistic passenger loads, representing either current or expected transportation demand. In addition, these assumptions should be adjusted to the concerned case study. For example, some shorter services may be crowded and almost impossible to be substituted by alternative transport systems. So, cancelling certain train lines may be forbidden given the passenger demand. However, such data were not available and thus out of the scope for this research. Note that priority of a line q is translated to corresponding train services q_k.

4.2 Measures for resolving timetable instability

Relaxation measures that are considered in this paper are relaxing the given target line plan and timetable design parameters. A target line plan represents a transport demand; however, it may happen that not all services can run due to limited available infrastructure. Also, a desired level of service (LoS) is determined by a given set of timetable design parameters (i.e. a given amount of transfer times, a maximum rate of running and dwell time supplements). Hence, apart from relaxing line frequencies, relaxing timetable design parameters can provide a smaller λ for the same size of the line plan. Even more, relaxing timetable design parameters may enable more demand to be satisfied. In practice, planners strive to accept all train service requests from railway undertakings since each additionally scheduled train brings additional profit to the infrastructure manager. Thus, the infrastructure manager may tend to sacrifice the LoS to some extent to schedule as many trains as possible. In particular, we propose the following measures: relax train line frequency M1, relax regularity constraints M2, and relax train-related constraints M3.

Measure M1 Measure M1 reduces the frequency of a train line while at least one service of each train line is maintained. Using M1 essentially lowers the total number of trains in the network and provides more possibility to fit remaining trains in the timetable. It is important to tackle train lines that exist on the critical circuit and not a random one. Otherwise, if a random train service has been selected, then it may happen that the critical circuit remains the same and does not affect the cycle time λ. Since the target line plan represents the transportation demand, we want to ensure that the least number of train services is removed and thus have a limited unsatisfied demand. In practice, the critical circuit C_λ typically includes (one or more) events from multiple train lines. We choose to remove one train service at a time.

In each iteration, the frequency of the train line with the lowest priority is decreased by one train service per scheduled cycle time T and the periodic event-activity network N is rebuilt. In particular, we first choose a critical service from Q_λ with the lowest priority based on the defined rules and refer to it as q_{crit}. Then, all events and processes of q_{crit}, $i \in E_{q_{\text{crit}}}$ and $(i,j) \in A_{q_{\text{crit}}}$, are removed from E and A, respectively. Additional constraints like headways, connections and regularities related to q_{crit} are also removed from A. Finally, the train line frequency is updated (i.e. reduced) in the priority list.

Measure M2 The first timetable design parameter, regularity, initially restricts the separation of train services of a line exactly to T/f_q. However, by allowing some degree of freedom to these constraints, we may achieve a better timetable stability. Thus, measure M2 introduces a relaxation parameter S, which is defined as a certain tolerance time that relaxes the regularity constraints. Constraints (6) are extended to:

$$T/f_q - S \leq \pi_j - \pi_i + y_{ij} \leq T/f_q + S, \quad \forall (i,j) \in A_{\text{reg}}, \forall q \in Q.$$

Measure M3 The second timetable design parameter that can influence timetable stability is the maximum allowed running time supplement rate. For a train running between events i and j, a running time bound u_{ij} represents the sum of technical minimal running time and the maximum time supplements, where the latter is often defined as a certain rate of the former. In essence, u_{ij} prevents that excessive time is scheduled which could lead to inefficient service. Both l_{ij} and u_{ij} are computed by given timetable design parameters and their values are decided by timetable planners which may have a significant impact on λ. Based on Fig. 1, more homogenised transport services may lead to smaller necessary headways between train services. Thus, allowing more time supplements to fast trains may result in a smaller minimum cycle time for the whole network. To provide more flexibility and use more time supplements when needed, we increase available running time supplements (measure M3). Thus, we introduce the relaxation parameter for running time supplements W and apply it to all upper bounds on A_{run}. The parameter $W \geq 1$ presents the multiplication factor for maximum allowed running times. Constraints (5) become:

$$l_{ij} \leq \pi_j - \pi_i + y_{ij} \leq u_{ij} \cdot W, \quad \forall (i,j) \in A_{\text{run}}.$$

In summary, measure M1 implies that the total number of train services will be reduced and transport demand may not be completely satisfied. Measures M2 and M3 suggest slight reduction in the expected LoS by relaxing planning rules. The latter two measures are always more acceptable and easier to implement than reducing the line frequencies. Based on the experts experience, we determined a quantitative value of three proposed measures and applied them when developing the algorithm for resolving timetable instability. In particular, it is preferable to relax first regularity constraints, then running time constraints and if no other options, then eventually train line frequencies.

4.3 Algorithm for resolving $\lambda \geq T$

Minimising cycle time is an NP-hard problem (Sparing and Goverde 2017) and solving PESP-λ once for the given line plan may result in high and even unacceptable running times. Hence, Sparing and Goverde (2017) proposed an algorithm to dynamically adjust bounds on λ during the optimisation run to speed up the computation times. Even with these improvements, the computation times for bigger instances were several hours. In our paper, the considered problem has an additional degree of freedom being that it is unknown whether the given line plan even provides a stable solution. And if not, the model would need to be extended with additional decision variables and constraints to allow defined relaxation measures. These extensions could make a PESP-λ significantly more complex and possibly unsolvable even for small instances. Another difficulty may arise regarding dynamical reassigning priorities of train lines. Therefore, we propose a heuristic approach that iteratively solves PESP-λ and applies the most appropriate relaxation measure. Algorithm 1 gives the workflow of the proposed heuristic for resolving instability and finding a stable timetable structure. Note that in each iteration an (intermediate) solution of PESP-λ given by (4)–(12) is optimal for a given instance. In general, the output of PESP-λ can be referred to as an optimal timetable structure, and only if $\lambda < T$ as an optimal stable timetable structure. In our case, due to a heuristic nature of applying relaxation measures, we refer to the solution of Algorithm 1 as an optimised stable timetable structure.

The algorithm takes as an input the target line plan Q, the train events and process times and corresponding headways represented as N, scheduled cycle time T, and parameters for S and W such as minimum and maximum values and relaxation steps. It also initialises λ to an infinitely large value, and S and W to the minimum values. The output of the algorithm is the optimised stable timetable structure (π, z, λ) and statistics on applied measures such as individual use of each measure and their final values. In general, let us consider one line plan Q, determined by the total number of services as a search neighbourhood, then the Algorithm 1 first uses M1 as long as the solution is far from a stable solution and seeks good (and relaxed) neighbourhoods. Once it reaches a potentially promising neighbourhood, it delves into this area and searches nearby solutions (i.e. the same line plan) by relaxing on M2 and M3. If a stable solution is found, then the algorithm terminates; otherwise, it continues the search in the new neighbourhood with a further relaxed line plan. It is also more important to preserve running time constraints over the regularity constraints. Thus, we always relax on M2 first, and M3 second. Applying relaxation measures in a relatively strict manner closely replicates their priorities determined by planning experts.

In each iteration, PESP-λ is solved first and the solution (π, z, λ) is obtained. Then, the choice of applied measure has been made based on the size of λ. If λ is not significantly bigger than T, but still hold $\lambda \geq T$, then regularity or train running is relaxed in a strictly defined order. The algorithm first relaxes regularity constraints M2 and applies it in subsequent iterations until S reaches $S_{\max}$. Once $S = S_{\max}$, then the algorithm relaxes running constraints M3 and it may be also repeated in several consecutive iterations until W reaches $W_{\max}$. If $\lambda \gg T$, then neither relaxations on regularity nor train running times can provide a stable solution, and the algorithm opts for the measure M1. Section 5.2 gives empirical experiments to quantify $\lambda \gg T$ for the given network and to determine a cycle time threshold that enables to obtain stable solutions when applying M2 and M3. The critical circuit C_λ is computed by *getCriticalCircuit* for (π, z, λ) and, respectively, sets of critical events and critical lines are determined, E_{crit} and Q_{crit}. Then, we choose a train service with the lowest priority q_{crit} and remove the corresponding events and processes from N. If S and/or W reached their maximum values, then we reset them to the minimum ones. This allows the algorithm to use again one of these two measures in following iterations. Algorithm 1 terminates when $\lambda < T$ is found.

Algorithm 1 incorporates a hot-start procedure for PESP-λ in each iteration to improve its computational efficiency. In the first iteration, the upper bound for the minimum cycle time $\lambda_{\max}$ is set to infinity, while in every other iteration, $\lambda_{\max}$ takes the value of λ from the previous iteration. Since a current iteration includes some relaxation of the input (compared to the previous iteration), it holds that a solution from previous iteration is feasible in the current. Exceptionally, when the algorithm returns from exploring M2 or M3, i.e. reached their maximum values, to using M1 again, an initial solution (π, z) and λ for PESP-λ is assigned from the last iteration in which measure M1 has been previously applied. This is to prevent obtaining an infeasible solution after resetting S and W to stricter (non-relaxed) values. Thus, using the computed solution in the previous iteration as the initial/starting one in the current iteration makes the PESP-λ model more computationally efficient. For example, in the ith iteration, M1 is used and the solution (π^i, z^i) and λ^i is generated. Then, in subsequent iterations, measures M2 and M3 are applied until jth iteration in which regularity and running times constraints are being relaxed maximally, $S = S_{\max}$ and $W = W_{\max}$, and the corresponding solution becomes (π^j, z^j) and λ^j. Note that solutions of ith and jth iteration have the same number of trains (i.e. the same number of events and processes), and $\lambda^i \geq \lambda^j$ while S and W for jth are relaxed. Then, in the $j+1$th iteration, the algorithm uses (π^i, z^i) and λ^i as a starting solution, while S and W are reset to $S_{\min}$ and $W_{\min}$, respectively.

Algorithm 1 Computing a stable timetable

```
Input: Line requests $Q$, $N = (E, A, T, l, u)$, minimum and maximum regularity relaxation
$S_{\min}$ and $S_{\max}$, regularity step $S_{\text{step}}$, minimum and maximum running relaxation $W_{\min}$
and $W_{\max}$, running relaxation step $W_{\text{step}}$
Output: stable timetable structure $(\pi, z)$, minimum cycle time $\lambda$, removed trains $R$,
regularity parameter $S$, running time parameter $W$
Initialise: $\lambda \leftarrow +\infty$, $(\pi, z) \leftarrow$ infeasible, $S \leftarrow S_{\min}$, $W \leftarrow W_{\min}$
while $\lambda \geq T$ do
    $(\pi, z, \lambda) \leftarrow$ solve PESP-$\lambda$
    if $\lambda \gg T$ OR $(S = S_{\max}$ AND $W = W_{\max})$ then
        if $S = S_{\max}$ AND $W = W_{\max}$ then
            Reset relaxation parameters: $S \leftarrow S_{\min}$, $W \leftarrow W_{\min}$
            $(\pi, z, \lambda) \leftarrow (\pi, z, \lambda)$ from the last known iteration where M1 was applied
        end if
        $C_\lambda \leftarrow$ getCriticalCircuit $((\pi, z), \lambda)$
        M1: Relax train line frequency of a critical service $C_\lambda$
        Choose $q_{\text{crit}} \in Q_\lambda$
        Update events $E$: $E \leftarrow E \setminus E_{q_{\text{crit}}}$
        Update processes $A$: $A \leftarrow A \setminus A_{q_{\text{crit}}}$
        Update $R$: $R \leftarrow R + 1$
    else if $\lambda \geq T$ then
        if $S < S_{\max}$ then
            M2: Relax regularity times $S \leftarrow S + S_{\text{step}}$
        else
            M3: Relax running times $W \leftarrow W + W_{\text{step}}$
        end if
    end if
end while
return $(\pi, z)$, $\lambda$, $R$, $S$, $W$.
```

4.4 Computing the initial solution for Algorithm 1

The defined PESP-λ model within Algorithm 1 performed well and reported small computation times in initial tests on instances of a few train lines. However, after applying PESP-λ on real-life instances tested in Sect. 5 (e.g. over 20 train services with frequency 2 and $T = 1800$ s), the model had difficulties to find a feasible solution within a reasonable time. To speed up computation times of PESP-λ, we develop a procedure to find an initial solution for PESP-λ and a good upper bound $\lambda_{\max}$ by solving the original PESP model with a fixed scheduled cycle time T. PESP is solved multiple times and each time with an increased value of T until a solution is found. Algorithm 2 describes the procedure for computing the initial solution for Algorithm 1. The input for Algorithm 2 is the line plan Q, periodic event activity network N, an initial value for the scheduled cycle time T, and the incremental step for the scheduled cycle time δ_T. The output is a feasible solution for a certain T_f. If a feasible solution was not found in the first iteration of PESP, T_f is increased for δ_T and the model is rerun. Algorithm 2 stops when a feasible solution is found. The obtained T_f from the last iteration is used to strengthen the upper bound on λ in Constraint (7), i.e. $\lambda_{\max} \leftarrow T_f$. This new $\lambda_{\max}$ together with the initial solution $(\pi_{\text{feas}}, z_{\text{feas}})$ notably reduce computation times of PESP-λ.

Algorithm 2 Computing an initial solution for PESP-λ

```
**Input:** Line requests $Q$, $N = (E, A, T, l, u)$, $\delta_T$
**Initialise**: $T_f \leftarrow T$, $(\pi,z) \leftarrow$ infeasible
**while** $(\pi,z)$ infeasible **do**
    solve PESP for given $T_f$
    **if** no solution found **then**
        $T_f \leftarrow T_f + \delta_T$
    **end if**
**end while**
out: $(\pi_{feas}, z_{feas})$, $\lambda\text{max} \leftarrow T_f$
```

5 Experimental results

5.1 Scenarios

We evaluate the capabilities of the model for resolving timetable instability on a highly utilised railway network in the central Netherlands. The considered network is bounded by the four main stations Utrecht (Ut), Eindhoven (Ehv), Tilburg (Tb) and Nijmegen (Nm), with a fifth main station 's-Hertogenbosch (Ht) in the middle and 20 additional smaller stations and stops. Four corridors connect Ht to the other main stations. Figure 4 depicts the passenger line plan of this network with 20 train lines.

For the experiment scenarios, we consider variants of target line plans with different numbers of train lines ranging from 14 to 30 and with frequencies between 1 and 2 train services per scheduled cycle time T. The scheduled cycle time equals

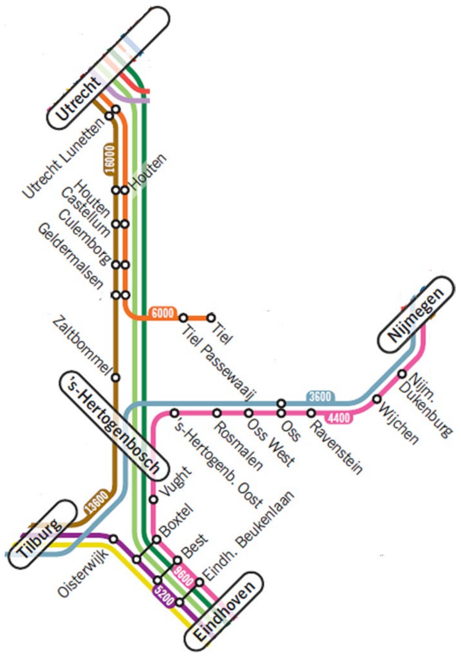

Fig. 4 Passenger line plan showing 20 train lines

$T = 1800$ s for all cases, with an exception of a test case for describing the effect of relaxation measures (Sect. 5.2), then $T = 1200$ s. The real-life scenario serves as a representative scenario that reflects the current demand on the network with 20 train lines all with frequency 1 and $T = 1800$ s. Weights τ_{ij} for all processes equal 10^{-5}. We do not consider station capacity. Following empirical experiments with $\lambda \gg T$, we determined the cycle time threshold that allows reaching stable solutions applying measures M2 and M3, see Sect. 5.3. We adopt cycle time threshold equal to $1.12 \cdot T$. Thus, while $\lambda > 1.12 \cdot T$, then measure M1 is always used. In a different network, this cycle time threshold may take a different value. The maximum number of iterations for Algorithm 1 is set to 40 iterations. The maximum CPU time for solving PESP-λ, i.e. one iteration of Algorithm 1, is set to 500 s. To compute the upper bound $\lambda_{\max}$ and a feasible solution (π, z) quickly, the time limit for Algorithm 2 is set to 50 s. This may lead to a slightly more relaxed upper bound $\lambda_{\max}$, but it satisfies the more important criterion that it provides a feasible solution to PESP-λ in Algorithm 1. Table 1 summarises characteristics of defined scenarios.

Table 2 defines parameters for relaxing measures M2 and M3 of Algorithm 1 such as minimum, step and maximum values. In addition, for M1, we chose to remove one critical train service at the time. Values for relaxing regularity and running time constraints are defined so as to allow up to two consecutive iterations of each relaxation measure.

In the following, we perform three types of analyses. First, the effect of single measures on the behaviour of Algorithm 1 is explained (Sect. 5.2). Second, the cycle time threshold has been determined (Sect. 5.3). Third, the performance of

Table 1 Characteristics for tested scenarios

Parameter	Notation (unit)	sc0	Experiments
Number of lines	$\|Q\|$	20	[14, 30]
Average frequency	f_q	1	[1, 2]
Total number of train services	$\sum_{q=1}^{\|Q\|} f_q$	20	[20, 60]
Scheduled cycle time	T (s)	1800	1800
Time step for Algorithm 2	δ_T (s)	–	100

Table 2 Input parameters for Algorithms 1 and 2

Parameter	Notation (unit)	Value
M2 minimum	$S_{\min}$ (s)	0
M2 step	S_{step} (s)	60
M2 maximum	$S_{\max}$ (s)	120
M3 minimum	$W_{\min}$	1
M3 step	W_{step}	0.1
M3 maximum	$W_{\max}$	1.2

our approach is demonstrated by applying different combinations of relaxation measures (Sect. 5.4).

5.2 Explaining the effect of relaxation measures

We interpret the effect of single measures on the computed timetable structure within Algorithm 1. For this purpose, we used a test case with $T = 1200$ s, $Q = 20$ and average frequency 2. Also, all three relaxation measures M1, M2 and M3 are considered. An assumed cycle time threshold is $1.3 \cdot T$ that quantifies $\lambda \gg T$. The solution efficiency in each iteration is measured as the sum of running and dwell time supplements and referred to as the total amount of time supplement. More time supplement means a lower efficiency. The similar behaviour was also observed in other test cases and we use the following one to show impacts of relaxation measures to minimal cycle time and solution efficiency. Figure 5 shows the convergence of the minimal cycle time and changes in time supplements, i.e. solution efficiency, through iterations until $\lambda < T$ was satisfied. An applied measure is denoted in each iteration by a corresponding marker.

We observed that in the first three iterations, M1 ($\times$ marker) is used consecutively since $\lambda \gg T$. Measure M1 usually provides a reduction of the minimal cycle time λ. Once λ is not significantly bigger than T (*iter* ≥ 4), other measures are considered. The relaxing line frequencies was applied again at iteration 8, after the other two measures were relaxed maximally ($S = S_{\max}$ and $W = W_{\max}$) and no further improvement was possible using M2 and M3. When doing so, S and W were reset to their initial values $S_{\min}$ and $W_{\min}$.

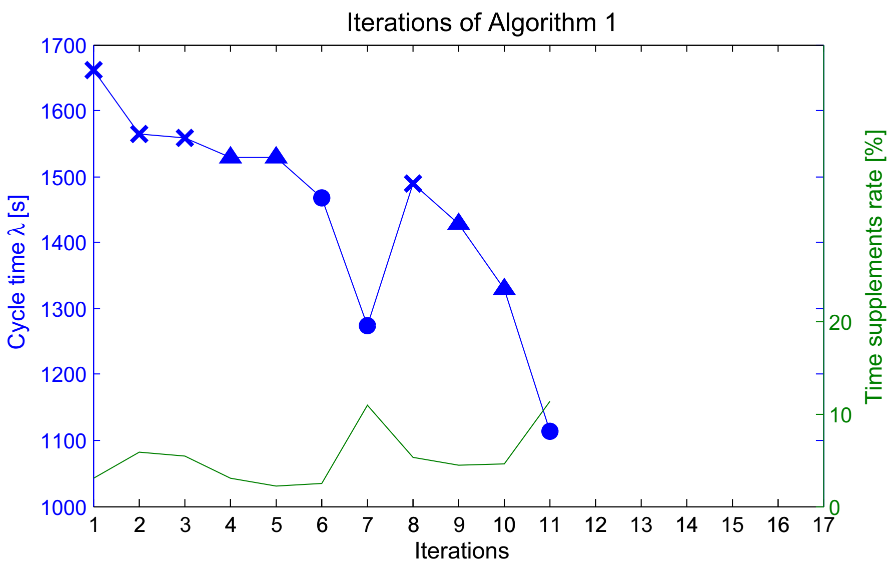

Fig. 5 Results for the test scenario of Algorithm 1 with measures M1 (×), M2 (filled triangle) and M3 (filled circle)

Measure M2 (▲ marker) was applied four times, in iterations 4, 5, 9 and 10. The values of S_{step} and $S_{\max}$ allowed maximally two consecutive iterations with M2. The first time applied, in iteration 4 (also holds for 9), M2 generated a solution with smaller λ and better efficiency. Then, in the following iteration 5, no improvement on λ is reported compared to the previous one, but M2 reduced time supplements and thus improved the efficiency. In iteration 10, a significantly lower λ is achieved with only slight deterioration of efficiency. On one hand, measure M2 does not always provide better λ because the critical circuit C_λ may not always include regularity arcs. For example, C_λ may consist of multiple headways and/or running times only. On the other hand, relaxing regularity times may positively impact the solutions efficiency.

Measure M3 (• marker) was applied after maximally relaxing regularity constraints, i.e. in iterations 6 and 11. Parameters W_{step} and $W_{\max}$ allowed maximally two consecutive iterations with M3. The first time M3 was used (in iteration 6), a better λ is found at the expense of relaxed running times. In iteration 7, λ was further improved, with even greater reduction of efficiency.

For some cases when M3 was applied, we observed that λ may remain unchanged, but create better efficiency. This results from the fact that a larger solution space admits better solutions. For instance, allowing more time supplements relaxes certain arcs (i.e. get more time supplements), which may allow other arcs to use less time supplements resulting in an overall better efficiency. Similarly to M2, measure M3 does not always provide better λ because the critical circuit C_λ may not always include running time constraints. For example, C_λ may consist of multiple headways and/or dwell times only. Finally, M1 may not always provide an improvement to λ, which may be due to the fact that multiple circuits exist in N with the same cycle time.

Overall, measure M1 relaxes the solution more radically and while $\lambda \gg T$, it is the only relaxation worth using. Only when λ is closer to T then Algorithm 1 considers also measures M2 and M3. After relaxing M2 and M3 to their maxima and still a stable solution was not found, the algorithm relaxes line frequencies again while resetting S and W to the smallest values. Measure M3 tends to provide bigger solution improvements than M2. Also, M3 seems to improve a solution more often, which may be due to the fact that there are more running arcs compared to regularity arcs in N; hence, more possibility to use a relaxation measure. A solution obtained by relaxing M2 and M3 may both increase and decrease solution's efficiency in subsequent iterations. The efficiency is often decreased when a smaller λ is found, while it is increased when λ stayed unchanged. The latter happens due to more flexibility and bigger search space for PESP-λ, while the former is due to smaller search space imposed by smaller λ.

5.3 Empirical analysis of $\lambda \gg T$

We performed empirical experiments of $\lambda \gg T$ using various sizes of target line plans, to determine the cycle time threshold that allows reaching stable solutions using measures M2 and M3. To do so, we (1) run Algorithm 2 to obtain an upper bound $\lambda_{\max}$ for a given line plan and then (2) run Algorithm 1 with measures M2 and M3 only and evaluate whether a stable solution could be obtained. Figure 6 reports the outcome including computed $\lambda_{\max}$ values and the corresponding cycle time λ over multiple scenarios. We can see that as long as the starting upper bound $\lambda_{\max}$ is under 2100 s, then it is possible to reach a stable timetable by relaxing M2 and M3. Based on these experiments, we adopt $\lambda > 1.12 \cdot T$ as the cycle time threshold to describe $\lambda \gg T$.

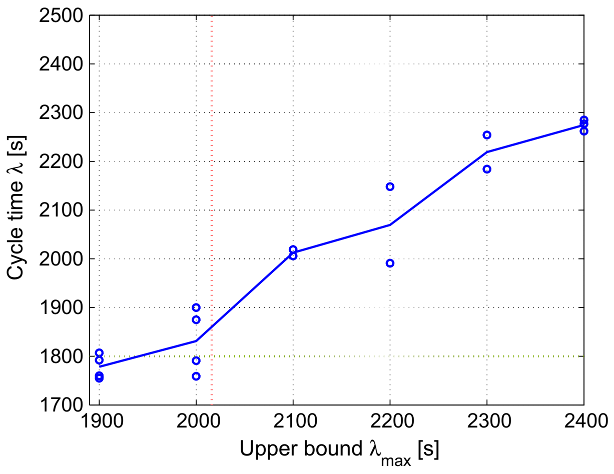

Fig. 6 An effect of an upper bound $\lambda_{\max}$ on the computed cycle time λ

5.4 Experimental results

We present the performance results of the proposed approach for resolving instability in periodic timetables. Two algorithm variants are considered: first, using only line plan relaxation (denoted M1) and second, using all three measures combined (denoted M123). We report computational performance of the heuristic for resolving instability (Algorithm 1) and a procedure for generating an initial solution for Algorithm 1 (Algorithm 2), and evaluate the quality of the computed solutions. For the former, we report an analysis of Algorithm 1 and computation times of both algorithms. For the later, we compare the minimum cycle times and two metrics of level of service: an average running time supplement rate and a regularity interval. A regularity interval of a solution is computed as an average time over all regularity activities A_{reg}.

Figure 7 shows relation between the size of the target line plan and an average number of iterations of Algorithm 1 needed to reach an optimised stable timetable solution. As long as a target line plan has 42 train services or less, exactly one iteration is performed, meaning that all services have been scheduled for an initial target line plan and timetable design parameters. At most 31 iteration was sufficient to find an optimised stable timetable, i.e. for M1 and 60 train services. On average, variant M1 needs typically lesser number of iterations compared to the variant M123. This can be explained by the fact that M123 tries to relax timetable design parameters (applies M2 and M3) once $T \leq \lambda \leq 1.12 \cdot T$. In case a stable solution is not found, then a new train service is removed (applies M1). Therefore, it may be expected that every time relaxation measures M2 and M3 are applied, additional iterations are performed. An exception to this behaviour is the target line plan with 44 train services. Here, M1 relaxes train frequencies, while M123 schedules the complete target line plans due to a bigger search space. For target line plans between 52 and 58 train

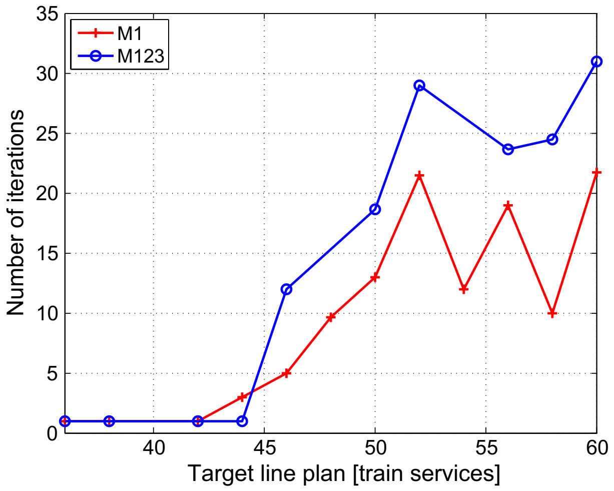

Fig. 7 Number of train services in target line plan vs number of iterations

services, we may also see certain oscillations in the number of iterations needed when increasing the target plan, which may occur due to random scenarios drawn in our experiments. In particular, some instances may represent more "difficult" target line plans, i.e. more services concentrated in congested parts of the network, while others have more evenly distributed train lines over network. In a real-life application, this observed behaviour is expected to be less significant, since planners would start with a current line plan and only add extra train services to it. In this case, it is expected that the number of iterations increases (or stagnates, but never decreases) with larger target line plans. The computation times of Algorithm 1 follow a similar pattern as of the number of iterations and it typically varied from 1000 s to 16000 s over all scenarios.

Table 3 gives performance statistics of Algorithms 1 and 2. It reports average CPU times and the number of iterations for both algorithms, an average time to solve PESP-λ and an average optimality gap if an optimal solution for PESP-λ is not found. For target line plans smaller that 44 train services, number of iterations for Algorithm 2 is 1, meaning that a feasible timetable solution is found for $T = 1800$ s. Therefore, Algorithm 1 needs exactly 1 iteration to find an optimised stable timetable. Algorithm 2 typically needs at most 6 iterations and finds an initial solution within 260 s. For Algorithm 1, number of iterations and CPU time increase with the size of the target line plan. It is observed that for sizes bigger than 44 train services, time limit is typically reached. Thus, an optimal solution was not found in each iteration, while the reported optimality gap was always under 20%. Figure 8 reports computation times for Algorithm 2 over all scenarios based on number of train services. The computation times for Algorithm 1 follow a similar pattern as of Algorithm 2. Finally, scenarios with less then 36

Table 3 Performance statistics for Algorithms 1 and 2

Target line plan	Algorithm 2		Algorithm 1		PESP-λ	Gap (%)
	Iterations	CPU (s)	Iterations	CPU (s)		
36	1	0	1	264.10	264.13	0
38	1	0.95	1	320.22	320.22	11.38
40	1	2	1	411.00	411.00	13.43
42	1	5.2	1	500.10	500.07	19.50
44	1.33	26.63	1.67	597.37	264.06	6.30
46	1.50	31.68	8.50	4250.30	500.08	17.61
48	2.67	95.73	7.33	3667.20	500.07	16.88
50	4.00	165.04	12.22	6112.06	500.08	16.93
52	4.80	211.17	16.40	8201.40	500.08	14.90
54	4.00	172.30	17.10	8300.40	500.04	19.56
56	3.42	131.85	15.50	7709.31	463.81	16.12
58	4.67	153.75	16.97	8233.93	500.12	18.64
60	5.50	257.48	19.88	9876.54	500.07	14.16

train services are not included in Table 3 since the optimal stable timetables were found without any relaxations and within 200 s.

Figure 9 shows relation between the size of a target line plan and an average size of optimised stable timetable solutions. It can be seen again that for target line plans under 42 train services, both M1 and M123 were able to schedule all services. Looking closer, M123 was able to fit all services also for target line plans with 44 train services, while M1 needed to cancel some train services. Overall, Algorithm 1 with

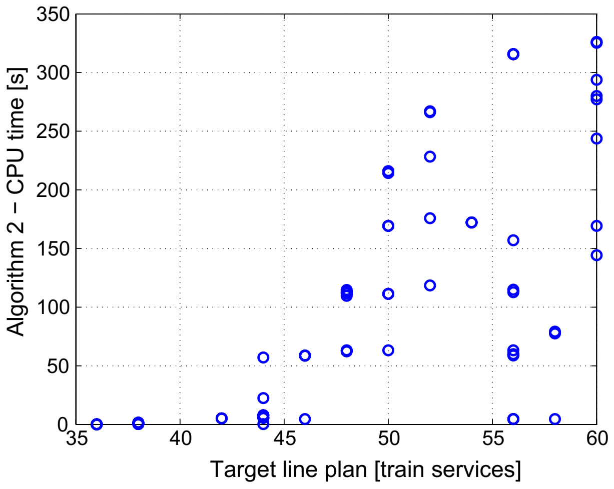

Fig. 8 Computation times for Algorithm 2

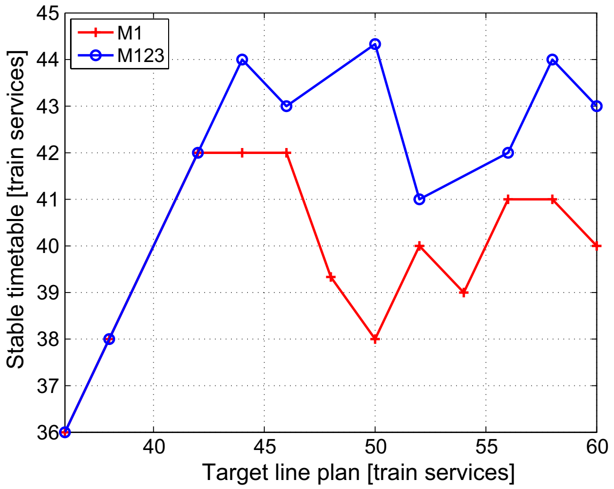

Fig. 9 Number of train services in target line plan vs number of train services in optimized stable timetable

M1 on average creates stable timetable solutions with less train services. The biggest difference in number of train services was at 50, equalling 38 for M1 compared to 45 for M123. It can be concluded that the considered network tends to become saturated with around 40 train services for M1 and around 42 for M123.

Figure 10 compares the target line plan and an observed regularity intervals. It can be seen that M1 tends to have a higher regularity interval as opposed to M123 for all scenarios. This is expected to some extent since M1 imposes a strict regularity constraint being equal to $\frac{\lambda}{2}$. (For train lines with frequency 1, regularity interval is not computed.) Also, values of the regularity interval smaller than $T = 1800$ s are due to $\lambda < T$. For some cases, regularity intervals for M1 and M123 become similar (e.g. at 52 and 58 services). Looking at cases with more than 44 train services, the regularity intervals for M1 range between 870 and almost 900s and seem to be rather stable. For M123, due to given relaxation of regularity constraints, a more deviating behaviour is observed and it ranges from 810 s to almost 890 s. Figure 11 compares the target line plan and an average running time supplements rate. The resulting time supplement rate is typically bigger for M123 over all scenarios. This is due to the fact that time supplements are relaxed in this variant. In particular, for M1, average time supplement rate is typically between 3.5% and 5.5%, and for M123 is bigger than 5.5% and reaches 7.5%.

Figure 12 provides an in-depth relation between average running time supplement rate and regularity intervals. It shows that lower time supplement rates usually lead to a higher regularity interval. In particular, for time supplement rate smaller than 5.5%, regularity intervals are higher than 870 s. However, when biger rates are observed, regularity interval may become as low as 800 s. In addition, higher time supplement rates (lower regularity intervals) correspond more to M123 (blue dots). Therefore, we could say that using M123, we may expect somewhat lower level of service in terms of both regularity and average running time supplements.

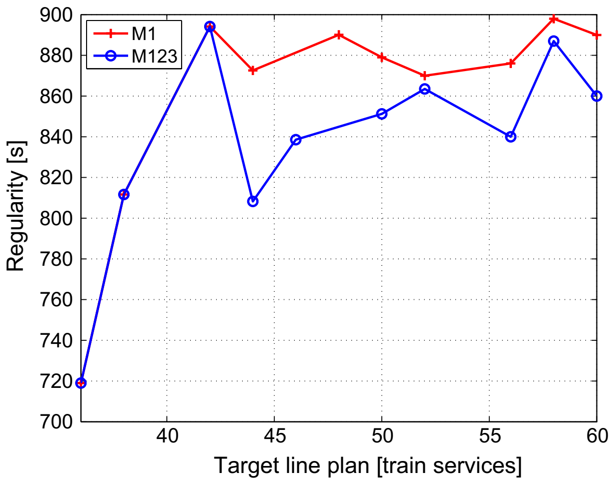

Fig. 10 Number of train services in target line plan vs regularity

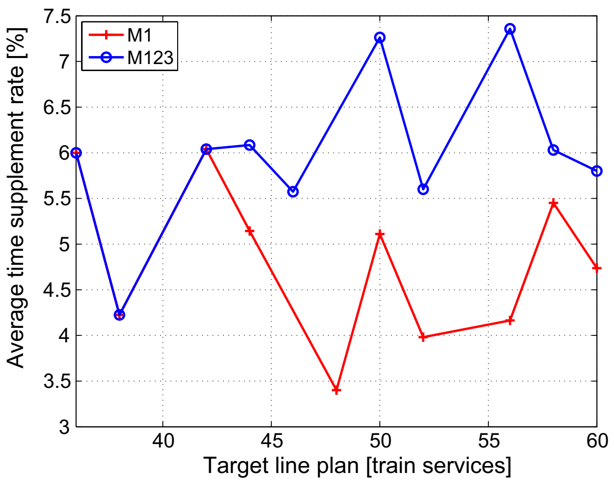

Fig. 11 Number of train services in target line plan vs average running time supplement rate

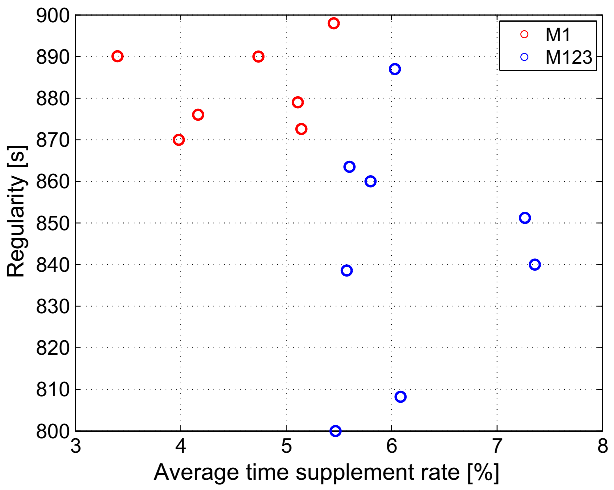

Fig. 12 Average running time supplement rate vs regularity (color figure online)

Figures 13, 14 and 15 evaluate quality of generated solutions, they compare the obtained stable timetables with minimum cycle times, regularity intervals and average running time supplement rates, respectively. Figure 13 clarifies that solutions with less train services may also obtain smaller minimum cycle times. For solutions with more than 48 services, λ is always bigger than 1740 s, thus providing at most 60 s of additional time allowances. So, increasing number of services in the solution leads to $\lambda \simeq T$. It means that the more services exist in the solution, the closer is the railway network to its saturation. Comparing performance of M1 and M123, for the

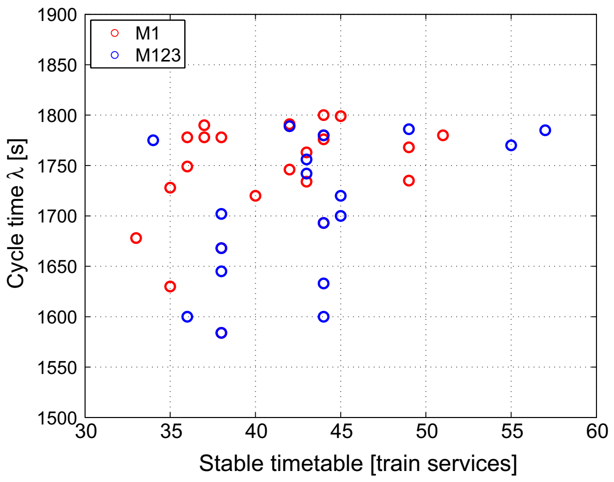

Fig. 13 Computed minimum cycle time λ

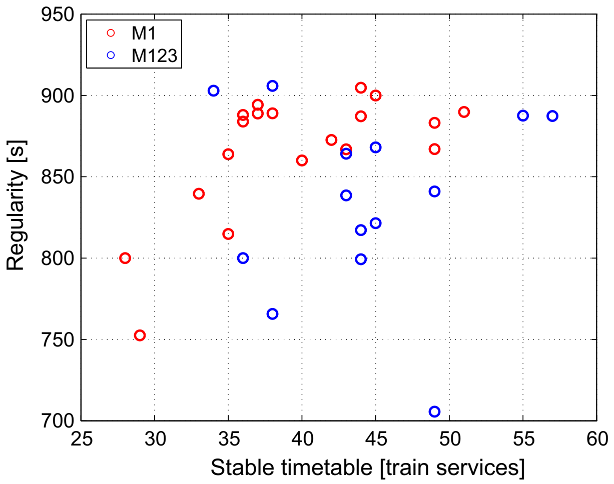

Fig. 14 Number of train services in stable timetable vs regularity

same size of solutions, it appears that M123 reaches smaller λ. Such smaller λ can be beneficial when allocating time allowances to design robust timetables, e.g. as in Bešinović and Goverde (2016).

In Fig. 14, for smaller solutions' sizes, regularity has larger variations, particularly when a stable timetable is smaller than 45 train services. When increasing the number of train services, the regularity interval comes closer to 900 s, i.e. $\frac{\lambda}{2}$, independent of the algorithm variant. It seems that M123 leads to more scheduled train services.

In Fig. 15, we can see that an increasing number of train services in the solution tend to lead to higher time supplement rates. In particular, for solutions with more than 45 train services, always more than 5% is needed, independent of the algorithm variant. Also the stricter variant M1 typically allocates less time supplements (red dots).

Table 4 shows performance statistics over all scenarios for variants M1 and M123: minimum cycle time λ, number of iterations (Iter), number of applications of each relaxation measure (M1, M2, M3), number of train services in stable timetables (Stable TT), average running time supplement rate (AvgRTsupp), total running and dwell time supplements (TotalSupp) and regularity interval. Variant M1 reports a higher computed cycle time λ, 1730.42 compared to 1689.71 for M123. Variant M123 usually needs more iterations, and thus longer computation time. However, on average, it relaxes less line plan compared to M123. Variant M123 applies M1 approximately 10 times while M1 as much as 15 times. In M123, M2 is used around 5 times while M3 less than that, a little over 4 times. This suggests that when relaxing design parameters, not always maximal relaxation (i.e. $S = S_{\max}$, $W = W_{\max}$) was necessary to find a stable solution. Providing a larger search space to M123 resulted in having almost 5 train services more scheduled on average, 44.29 as opposed to

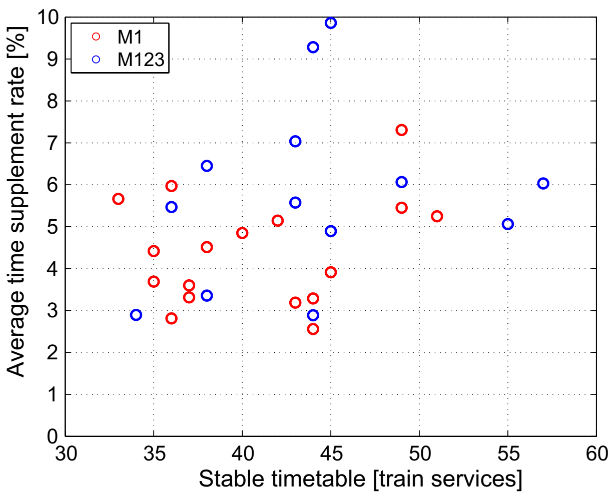

Fig. 15 Number of train services in stable timetable plan vs average running time supplement rate (color figure online)

Table 4 Overall computation statistics for variants M1 and M123

Variant	λ (s)	Iter	M1	M2	M3	Stable TT	AvgRTsupp (%)	TotalSupp (s)	Regularity (s)
M1	1730.42	16.11	15.11	0	0	40.53	4.41	8548.58	865.58
M123	1689.71	19.79	10.68	5.07	4.14	44.29	6.13	11117.71	836.04

s

40.53. More train services, however, lead to higher average running time supplement rates and lower regularity intervals. Overall, we could say that M1 variant provides a higher level of service over multiple scenarios; however, this may often be a consequence of lesser train services scheduled. Therefore, if more services is an imperative in the planning stage, then we need to relax timetable design parameters and accept a certain cost of lost level of service.

6 Conclusions

This paper proposed a new approach for resolving instability in railway timetabling problems. We developed a heuristic that relaxes the target line plan and timetable design parameters. The heuristic integrates a mixed integer linear programming model for minimising the cycle time PESP-λ and applies different combinations of relaxation measures. The algorithm runs until a stable solution is found and an optimised stable timetable structure is retrieved.

We observed that if transport demand is significantly bigger than the infrastructure capacity can handle, then it is necessary to relax some line frequencies. Even more, it is important to determine and relax train lines that occur in the critical circuit. Relaxing regularity constraints does not always generate a better solution, and relaxing running times seems more effective. By relaxing regularity and running time constraints, we may need more iterations to find a stable solution, but more importantly, such a solution tends to have less train services removed due to more flexibility given to the PESP-λ model. Therefore, it would be preferable to use all the available measures to generate a stable timetable structure and minimise the unsatisfied demand. In case that more regularity, as well as an overall higher level of service , is preferred, it can be achieved by restricting timetable design parameters and relaxing only frequencies of train lines. This, however, may come with a cost that less train services are scheduled in such a stable timetable. Our approach can be used to determine a saturation level of a given railway network.

Future research could be addressed in several directions. First, an actual forecasted transport demand can be considered in the timetabling problem instead of the current assumptions. Second, identifying additional combinations of relaxation measures could improve further satisfaction of transport demand. Third, developing more robust heuristics with a more advanced searching technique that integrate multiple measures within each iteration could reduce computation times. Fourth, we may try additional relaxation measures, such as shortening or splitting train lines, which could result in less relaxed line frequencies and better demand satisfaction. The proposed model for resolving instability can be a very useful support tool in timetable planning for areas with high demand and/or scarce infrastructure capacity and can help in finding a stable timetable that maximises satisfied transport demand.

Acknowledgements The research has been supported by the Erasmus+ mobility grant and performed during the first author's visit to the UK infrastructure manager Network Rail.

References

Bergmann DR (1975) Integer programming formulation for deriving minimum dispatch intervals on a guideway accommodating through and local public transportation services. Transp Plann Technol 3(1):27–30

Bešinović N, Quaglietta E, Goverde RMP, Roberti R (2016) An integrated micro-macro approach to robust railway timetabling. Transp Res Part B Methodol 87:14–32

Bešinović N, Goverde RMP (2016) Two-stage stability-for-robustness approach for periodic railway timetabling. In: The 9th Triennial symposium on transportation analysis (TRISTAN IX), 13–17 June 2016, Oranjestad, Aruba

Cacchiani V, Toth P (2012) Nominal and robust train timetabling problems. Eur J Oper Res 219(3):727–737

Caimi G, Fuchsberger M, Laumanns M, Schüpbach K (2011) Periodic railway timetabling with event flexibility. Networks 57(1):3–18

Goverde RMP (2007) Railway timetable stability analysis using max-plus system theory. Transp Res Part B Methodol 41(2):179–201

Goverde RMP (2010) A delay propagation algorithm for large-scale railway traffic networks. Transp Res Part C Emerg Technol 18(3):269–287

Hansen IA, Pachl J (eds) (2014) Railway timetabling and operations, 2nd edn. Eurail, Hamburg

Heidergott B, Olsder GJ, Van Der Woude J (2014) Max-Plus at work: modeling and analysis of synchronized systems: a course on Max-Plus algebra and its applications. Princeton University Press, Princeton

Heydar M, Petering MEH, Bergmann DR (2013) Mixed integer programming for minimizing the period of a cyclic railway timetable for a single track with two train types. Comput Ind Eng 66(1):171–185

Kroon LG, Peeters LW (2003) A variable trip time model for cyclic railway timetabling. Transp Sci 37(2):198–212

Kroon LG, Peeters LW, Wagenaar JC, Zuidwijk RA (2013) Flexible connections in PESP models for cyclic passenger railway timetabling. Transp Sci 48(1):136–154

Kümmling M, Großmann P, Nachtigall K, Opitz J, Weiß R (2015) A state-of-the-art realization of cyclic railway timetable computation. Public Transport 7(3):281–293

Kümmling M, Großmann P, Opitz J, Weiß R, Nachtigall K (2015) Extraction of Significant Conflicts in Periodic Timetabling. In: The 13th conference on advanced systems in public transport (CASPT), 19–23 July 2015, Rotterdam, The Netherlands

Liebchen C (2008) The first optimized railway timetable in practice. Transp Sci 42(4):420–435

Nachtigall K (1993) Exact solution methods for periodic programs. Technical Report 14/93, Hildesheimer Informatik-Berichte

Nachtigall K, Opitz J (2008) A modulo network simplex method for solving periodic timetable optimisation problems. In: Kalcsics J, Nickel S (eds) Operations research proceedings 2007. Operations Research Proceedings, vol 2007. Springer, Berlin, Heidelberg

Peeters LWP (2003) Cyclic railway timetable optimization, Ph.D. thesis, Erasmus Universiteit Rotterdam, Rotterdam, The Netherlands

Petering MEH, Heydar M, Bergmann DR (2015) Mixed-integer programming for railway capacity analysis and cyclic, combined train timetabling and platforming. Transp Sci 50(3):892–909

Polinder GJ (2015) Resolving infeasibilities in the PESP model of the Dutch railway timetabling problem. MSc thesis, Delft University of Technology, Delft

Schrijver A, Steenbeek A (1994) Dienstregelingontwikkeling voor Railned (Timetable construction for Railned). Technical report, Center for Mathematics and Computer Science, Amsterdam

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

Sparing D, Goverde RMP (2017) A cycle time optimization model for generating stable periodic railway timetables. Transp Res Part B Methodol 98:198–223

Stanley P (2011) ETCS for engineers. Eurail Press, Hamburg

The Digital Railway Programme (2016) www.digitalrailway.co.uk. Accessed 28 Sept 2016

Zhang X, Nie L (2016) Integrating capacity analysis with high-speed railway timetabling: a minimum cycle time calculation model with flexible overtaking constraints and intelligent enumeration. Transp Res Part C Emerg Technol 68:509–531