Resolving instability in railway timetabling problems

Bešinović, Nikola; Quaglietta, Egidio; Goverde, Rob

Publication date
2017

Document Version
Accepted author manuscript

Published in
The 7th International conference on Railway Operations Modelling and Analysis (RailLille2017)

Resolving instability in railway timetabling problems

Nikola Bešinović[1,*], Egidio Quaglietta[2], Rob M.P. Goverde[1]
[1]Department of Transport and Planning, Delft University of Technology, The Netherlands
[2]Control, Command & Signalling, Network Rail, United Kingdom
[*]E-mail: n.besinovic@tudelft.nl, Phone: +31 (0) 15 2784914

Abstract
A growth of the railway transportation demand is forecasted in the next decades which needs an increase of network capacity. Where possible, infrastructure upgrading can release extra capacity, although in some cases this is not enough to satisfy the entire transportation demand unless optimised timetabling is performed. We propose a heuristic approach to develop a stable and timetable which maximise 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. In addition, the aim is to maximize the satisfied demand by keeping as many train services as possible. We develop the mixed integer programming (MIP) model for minimizing cycle time to find an optimal stable timetable for the given line plan. The heuristic iteratively solves the MIP 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, PESP, instability, minimal 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 the expected transportation demand is called *target line plan*. In order to meet the demand, railway planners have the objective to design a timetable that possibly runs all trains in the target line plan within a scheduling period T, called as *nominal cycle time*, usually coinciding with one hour. In railway planning, it is usual to have train services repeating every period 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 those stations/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) [28].

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* [7]. 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) in order 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 laid 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 period T), such timetabling models shall be able to fit in as many train lines as possible while providing smooth running and a mitigation of delay propagation. In other words, timetabling shall focus on maximising utilisation of infrastructure capacity while providing robustness versus stochastic disturbances. We define the minimum cycle time λ as the minimum amount of time during which all train events in the target line plan can be scheduled without conflicts. According to Heidergott et al. [11] and Goverde [8], 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 *stable* [8]. Planners always aim to identify a stable timetable which can accommodate the entire target line plan within the nominal cycle time so to satisfy the forecasted demand. Instead, the questions is how to tackle scheduling problems where 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 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 minimizing 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 *optimal stable timetable (structure)* that satisfies the transportation demand as much as possible. To do so, a heuristic approach has been developed that integrates an optimization model and relaxation rules to minimize necessary corrections to the target line plan. We design a mixed integer programming (MIP) model for solving a timetabling problem that minimizes the cycle time λ for a given line plan. This MIP 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 synchronization constraints and relaxing train-related constraints, and former reducing the line frequencies. The iterations between the minimal cycle time optimization and relaxations are repeated until the optimal stable timetable structure has been found. When a stable timetable (structure) is obtained, time allowances can be optimally allocated to maximize robustness versus stochastic disturbances. An application of the proposed approach is performed for a part of the Dutch railway network and for an assumed increased transportation demand. Results show that 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. For our model, we do not assume that all train lines from the given line plan can be scheduled. This provides the model more flexibility to find a stable solution and allows a wider application in railway timetable planning. In particular, such model could be used in the dense networks where the capacity use (i.e., minimum cycle time) is already becoming critical, $\lambda = T$, and also, to evaluate infrastructure improvements projects. Second, the proposed procedure for computing an initial solution and stronger upper bound for λ in MIP 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 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 an maximal set of train lines to satisfy (most of) the transportation demand.

In Section 2, we give the literature review. Section 3 introduces periodic event scheduling problem (PESP) and the model for minimizing cycle time based on PESP, PESP-λ. Section 4 defines first the assumptions and priority rules and then, the heuristic algorithm for resolving instability and improvements for PESP-λ are presented. Section 7 gives the computation results of proposed heuristic for two tested scenarios on the Dutch network. Finally, Section 6 gives the conclusions and main observations.

2 Literature review

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. 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. A timetable structure of a considered timetable gives a train order and feasible event times. Process times tend to equal the minimal times, and extended only to allow all events to become feasible. The timetable structure corresponds to the computed minimum cycle time λ, all events are between 0 and λ and represents the compressed timetable.

Timetable feasibility is the ability that a timetable exists for a given line plan and all train- and passenger- and safety-related 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.

Periodic railway timetabling problem are often presented as a periodic event scheduling problem (PESP) introduced by Serafini and Ukovich [26]. Afterwards, a significant amount of research have been assigned to solving timetabling based on PESP. For solving PESP, Schrijver and Steenbeek [25] applied constraint programming to find a feasible timetable, while Kümmling et al. [14] used SAT solvers to the same problem. By adding an objective function to the PESP formulation, then the timetabling problem can be solved using mixed integer programming (MIP) techniques [22]. Other papers that further developed PESP-based models for timetabling are [15], [18], [16], [20].

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. [13], Polinder [24] and Bešinović et al. [3] presented approaches for solving problems of local infeasibility in periodic timetabling problems. Polinder [24] resolves conflicts reported by the planning model DONS [25] and tries to resolve them by relaxing neighbouring processes. Kümmling et al. [13] proposed a similar approach to support the planning model TAKT [14]. Differently, Bešinović et al. [3] proposed an iterative micro-macro approach for designing (microscopically) conflict-free, stable and robust timetables. The macro model computes a timetable, which has been evaluated on microscopic conflict-freeness and (local) stability. Here, stability is defined at the local (station or corridor) level as the time share that has to be free, i.e., not used by running trains. If a timetable is recognized as locally unstable, then running times have been relaxed and the macroscopic model rerun. Even these three models 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 minimizing the cycle time for measuring stability was introduced by Bergmann [1] for a single-track line and homogeneous fleet. Heydar et al. [12] 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 minimize the capacity occupation and minimize the total dwelling time of local trains at all stations. Sparing and Goverde [27] developed an extension to the PESP model that minimizes the cycle time and train running times which is applicable to both lines and networks. Petering et al. [23] extended the model of Heydar et al. [12] to allow selection of stop platform in a station and schedule train overtakings. Output of these models is an (near) optimal timetable structure, i.e., a compressed timetable and not a final one.

In order to translate a timetable structure to an actual timetable, Bešinović and Goverde [4] proposed a two-stage model for computing a stable and robust timetable. In particular, the first stage solved 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 structure. Therefore, it is crucial to make a distinction between timetable stability and timetable feasibility when minimizing cycle time. In general, a model for minimizing cycle time may find a feasible (or optimal) timetable structure for a given input. However, such feasible (or optimal) timetable structure may not be acceptable if $\lambda > T$, meaning that all trains cannot be scheduled within the desired nominal cycle time. Thus, even though a feasible solution exists, a timetable structure may be unstable. The only way to make a timetable that can be operated is to have λ smaller T which could be done only by relaxing certain input constraints. In this paper, we extend the approach of Sparing and Goverde [27] and define a more general approach to resolve timetable instability and design optimal stable timetables. Finally, we limit ourselves to finding an optimal stable timetable structure, while the final timetable can be generated by applying [4].

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 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 characterized by its origin and destination, stopping pattern and frequency f_q within a given nominal cycle time T. In addition, if $f_q > 1$, then line q consists of f_q train services. 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 process times $(i, j) \in A$, 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 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 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 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 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:

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

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.

We also define subsets of events and processes for each train line E_q and A_q, respectively. When $f_q = 1$, each train line $q \in Q$ consists of a sequence of process times $a = (i, j)$, where i and j are two consecutive events. When $f_q > 1$, events and processes of a train line q are replicated f_q times and depict a given train service q_i where $i = \{1, \ldots, f_q\}$. In order to secure the synchronised scheduling of train services of the same line, meaning that services q_i are equally separated in time, we introduce *synchronization constraints* A_{sync}. These constraints are defined between services of one line, where i and j are events of two following services. Time separation between such two services equal to T/f_q and can be written as:

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

Finally, subsets of events i assigned to the train service q_i are defined as E_{q_i} and subsets of processes (i, j) as A_{q_i}. Figure 1 gives an small example of a periodic event-activity network with two trains stopping at a station.

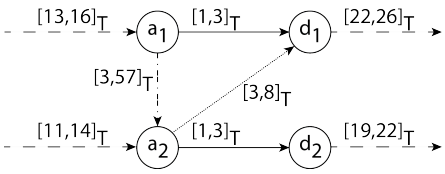

Figure 1: An extract of a periodic event-activity network for two trains arriving (a) and departing (d) a station with running (dashed line), dwell (full), transfer (dotted) and headway (dash-dotted) constraints

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

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

$$0 \leq \pi_i < T, \qquad \forall i \in E \tag{2}$$

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

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 a modulo constraint as a binary decision variable.

3.2 PESP for minimizing cycle time

The model aims at finding the optimal timetable structure over N by minimizing the cycle time λ. The minimum cycle time is the shortest time duration in which all the events scheduled in the timetable (within the nominal cycle time) are feasible for all precedence constraints such as minimum running times, dwell and connection times and minimum headway times required by the infrastructure. A *circuit* is a sequence of events in N. We focus only at elementary circuits, i.e., circuits in which no vertex (i.e., events) but the first and last appears twice. We refer to a circuit that builds the minimum cycle time as to the *critical circuit* C_λ[1].

The difference between minimum and nominal cycle time defines the available time allowances (time supplements and buffers). Therefore, the computed optimal timetable structure gives the train order that uses the infrastructure in the most optimal way and therefore, leaves the most time allowances. 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 nominal cycle time T, and thus, the timetable structure is unstable.

The problem of finding the optimal timetable structure is formulated by considering the cycle time T as variable and substituting it with λ, and then solving the problem of minimizing the cycle time. In addition, we use minimization of journey times as a secondary objective term with a small weight α to prevent an excessive extension of journey times. The new MIP formulation of minimizing cycle time is then the following:

$$(PESP - \lambda) \qquad \text{Minimize} \quad \lambda + \alpha \sum_{(i,j) \in A} \tau_{ij}(\pi_j - \pi_i + y_{ij}) \tag{4}$$

[1] In theory, multiple circuits with the cycle time equal to λ may occur; however, in practice, the critical circuit is most often unique due to the great dependency between events in N.

subject to

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

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

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

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

$$0 \leq y_{ij} \leq \lambda \tag{9}$$

$$y_{ij} \geq \lambda - M(1 - z_{ij}) \tag{10}$$

$$y_{ij} \leq M\lambda \tag{11}$$

The objective function (4) is minimizing the cycle time and total journey times. Here τ_{ij} defines a process type dependent weights, and may, for instance, differ for running and dwell times. Constraint (5) defines bounds on the process times. Constraint (6) synchronizes 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 that the given upper bound $\lambda_{\max}$. Since the nominal cycle time T from (1)-(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 linearized by introducing new variables $y_{ij} = z_{ij}\lambda$ and constraints (9)-(11) according to [27]. Here, M is a suitable upper bound for the objective value λ. In particular, M is equal to the the upper bound of λ, $M = \lambda_{\max}$. In the remainder of the paper, we refer to the model for minimizing cycle time as PESP-λ. The output of PESP-λ is the minimum cycle time λ and the timetable structure (π, z) where π_i are event times for all $i \in E$ and z_{ij} modulo parameters for each arc $(i,j) \in A$.

4 Resolving timetable instability

4.1 Assumptions

To determine a 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 in the network. In addition, we make a set of critical services Q_λ and q_i is in Q_λ if it has at least one event in E_λ.

To obtain a stable timetable, the timetable must be operated with a nominal cycle time $\lambda < T$. In order 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, undertaking any relaxation measure on a random train line, i.e., not being part of the critical circuit, may not impact the the stability of the whole system, i.e., value of λ would stay unchanged. For example, removing a train service in a low-dense network area which events are not part of the critical circuit may result in unchanged (in)stability of the timetable.

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, such that affects the least transportation demand. Thus, after discussions with planners, we introduce a set of priority rules and assumptions which would allow to incorporate the unsatisfied demand implicitly. Such rules suggest which train line should be better to adjust first and are based on the train line characteristics.

Before generating the rules, we made the underlying assumptions. First, all train lines of the same service type have the same passenger load (passenger capacity) between two

stops and are treated as equally important. Second, as a consequence of previous, overall transport capacity of a train line differ 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 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 selected four train line characteristic for determining their priorities that are 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 with less significant type (i.e., local before intercity); 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. The first rule defines that we choose first the local line over the intercity line given the assumption that the former has significantly less passengers and thus, imposes lower demand dissatisfaction. For the second rule, removing lines covering shorter distances is preferred as it is more probable that passengers can find alternative travel options such as bus, taxi or car. The third rule suggests that by opting for the line with lesser stops, we will affect lesser number of passengers. The fourth rule suggests that by reducing a frequency of the most frequent trains, we may still provide an overall high-frequency service to all passengers. 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. 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_i.

4.2 Measures for resolving timetable instability

Relaxation measures that are considered in this paper are relaxing the given line plan and timetable design parameters. In particular, we propose the following measures: relax train line frequency (M1), relax synchronisation constraints (M2), and relax train-related constraints (M3).

Measure M1 reduces the frequency of a train line while guarantees that 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 trains in the timetable. It is important to tackle train lines/services 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. The critical circuit C_λ may contain only a few events of two trains lines, but often includes (one or more) events of multiple train lines. Corresponding processes in the critical circuit can be running, dwell, connection or headway processes. Figure 2 gives an example of a critical circuit that comprises six arrival and departure events over four train lines.

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 the E and A, respectively. Additional constraints like headways,

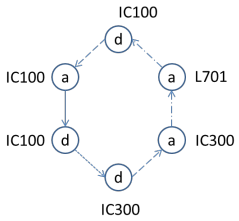

Figure 2: Example of a critical circuit including six arrival (a) and departure (d) events of four different train lines. (Line types are the same as in Figure 1)

connections and synchronizations related to q_{crit} are also removed from A. Finally, the train line frequency is updated (i.e., reduced) in the priority list.

Target level of service of the timetable can be achieved by setting a given set of timetable design parameters (i.e. a given amount of transfer times, as well as 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, relaxed timetable design parameters may allow that less train services need to be removed. This is highly important since having as much as possible train services is always the first priority.

Synchronization constraints allow regular intervals between trains of the same train line. For example, a train line with $f = 3$ and for $T = 60$ minutes has three synchronized services every 20 minutes. Initially, synchronization constraints restrict 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 stable timetable with less reduction in line frequencies. Instead of having exact 20-minute services, we may allow a small deviation of this synchronization value. Thus, measure M2 introduces a relaxation parameter S, which is defined as a certain flexibility time that relaxes the synchronization constraints. Constraints (6) are extended to:

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

Another timetable design parameter that can influence timetable stability are maximum allowed running time supplements. For a train running between events i and j, a running time bound u_{ij} represent the sum of technical minimal running time and the maximum time supplements, where the later is often defined as a certain rate of the former. In essence, u_{ij} prevents that too excessive time is scheduled that would lead to inefficient service and thus, lower quality of service. Both l_{ij} and u_{ij} are timetable design parameters, their values are decided by timetable planners and may have a significant impact on λ.

The aim in timetabling is to have efficient train services, meaning that trains could run as fast as possible and scheduled with only small running time supplements. This may create a significant speed difference between trains lines of different types. For example, if two train lines, a faster and slower (eg., intercity and local), on a corridor are scheduled with minimal time supplements, then the speed difference between the two will be significant and two trains together would need more infrastructure capacity to run over. On the other hand, if a faster train is allowed to run slower and thus, allow more homogeneous service, then less capacity will be used [10].

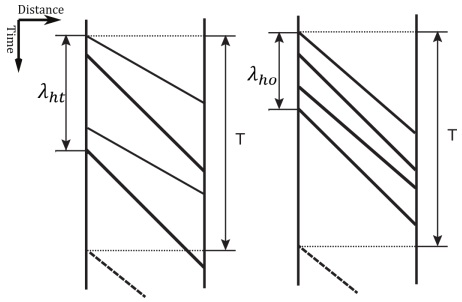

Figure 3: Effect of speed on the minimum cycle time

Figure 3 shows the influence of heterogeneity on the minimum cycle time. It gives minimum cycle times of two train lines with the frequency of two in period T running over a single track. The dashed line is the first train repeated in the next period. Figure 3a (3b) represents a heterogeneous (homogeneous) services with minimum cycle time λ_{ht} (λ_{ho}). The speed difference between train services in Figure 3a is evident, while running times of fast service in Figure 3b are extended. Clearly, λ_{ho} is smaller than λ_{ht} due to smaller necessary headways between train services. Thus, allowing more time supplements to fast trains, may result in 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.

A higher maximum running time that may indeed result in reduced timetable efficiency, but could also provide a more demand to be satisfied. Thus, we introduce the relaxation parameter for running times supplements W and apply to all upper bounds in A_{run}. The parameter W presents the multiplication factor for maximum allowed running times. Constraints (5) become:

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

In general, 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 service quality by relaxing planning rules. However, these 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 measures M1-M3 and applied them when developing the algorithm for resolving timetable instability. In particular, it is always preferable to relax first synchronization constraints, then running time constraints and if no other options, then eventually train line frequencies.

4.3 Algorithm for resolving $\lambda > T$

Minimizing cycle time is an NP-hard problem and solving PESP-λ once for the given line plan may result in high and even unacceptable running times. Due to this, Sparing (2016) proposed an algorithm to dynamically adjust bounds on λ during the optimization run to speed up the computation times. Even with these improvements, the computations times for some bigger instances were several hours. However, in our paper, the considered problem has additional degree of freedom being that 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 measures such as reducing line frequencies and relaxing constraints. This could make a PESP-λ significantly more complex and possibly unsolvable even for small instances. Another difficulty may be to dynamically reassigned priorities of train lines. Therefore, we propose a heuristic approach that iteratively solves PESP-λ and applies the most appropriate relaxation measure. In general, we refer to the output of PESP-λ as an optimal timetable structure, and only if $\lambda < T$, then the output is referred as an *optimal stable timetable structure*. Algorithm 1 gives the workflow of the proposed heuristic for resolving instability in timetabling problem and finding an optimal stable timetable structure.

The algorithm takes as an input the target line plan Q, the train process times and corresponding headways are represented as N, nominal cycle time T_o, synchronisation and running relaxation parameters like initial values for S and W, relaxation steps, and maximum values for each measure. It also initializes λ to an infinitely large value, and S and W to the minimum values. The output of the algorithm is the optimal stable timetable structure (π, z), λ and statistics on applied measures like individual use of each measure and their final values. In each iteration, PESP-λ is solved first and the solution (π, z, λ) is obtained. Then, the choice of an applied measure has been made based on the size of λ. In general, if $\lambda \gg T$, then no relaxations on synchronization nor train running times can provide a stable solution. So, the algorithm opts for the measure M1 in this case. The critical circuit C_λ is computed (*getCriticalCycle*) for (π, z, λ) and sets of critical events and critical lines are determined, E_{crit} and Q_{crit}, respectively. 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.

In the first iteration, the upper bound for the minimum cycle time $\lambda_{\max}$ is set to infinity, while in every other iteration, the $\lambda_{\max}$ takes the value of λ from the previous iteration. Since a current iteration includes some relaxation of the input (from the previous one), then it holds that a solution for (π, z) and λ also exists. In this way, by giving a solution computed in the previous iteration, we make PESP-λ more computationally efficient. Exceptionally, when the algorithm returns from exploring M2 (or M3) (i.e., reached 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. For example, after i-th iteration using M1 the solution (π_i, z_i) and λ_i is generated. Then, the algorithm continues to apply M2 and M3 in consequent iterations while synchronisation and running times constraints are being relaxed eventually to $S_{\max}$ and $W_{\max}$ and in j-th iterations creates (π_j, z_j) and λ_j. Here, π_i and π_j have the same number of events, $\lambda_i \geq \lambda_j$ and S and W for j are relaxed. In the iteration $j + 1$, the algorithm uses the (π_i, z_i) and λ_i for initial solution, while S and W are reset.

On the other hand, if λ is not significantly bigger T, but still $\lambda \geq T$, then we apply one of the two remaining measures in a strictly defined order. The algorithm first chooses to relax synchronization 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 $W_{\max}$. Algorithm 1 terminates when $\lambda < T$ is found.

In general, if we consider one line plan Q, determined by total number of services, as a

search neighbourhood, we may say that the Algorithm 1 first uses M1 as long as the solution is far from a stable solution and searches for good (and relaxed) neighbourhoods. Once it reaches a potentially promising solution neighbourhood, it delves into this area and searches nearby (i.e., the same line plan) solutions by relaxing on M2 and M3. If a solution is found, then the algorithm terminates and otherwise, continues the search in the new neighbourhood that has a relaxed line plan. Designing the preferences in applying relaxation measures in a relatively strict manner closely replicates their priorities determined by planning experts. In particular, it is more important to preserve running time constraints over the synchronization constraints. Thus, we always relax on M2 first, and apply train-related relaxation only when $S = S_{\max}$. In addition, after initial tests, we accepted $\lambda > 1.3 \cdot T$ as a significant difference between λ and T.

Algorithm 1 Computing a stable timetable

Input: Line requests Q, $N = (E, A, T_o, l, u)$, nominal cycle time T, minimum and maximum synchronization relaxation $S_{\min}$ and $S_{\max}$, synchronization step S_{step}, minimum and maximum running relaxation $W_{\min}$ and $W_{\max}$, train relaxation step W_{step}
Output: stable timetable structure (π, z), minimum cycle time λ, removed trains R, synchronisation parameter S, running time parameter W
Initialize: $\lambda \leftarrow +\infty$, ($\pi$,z) $\leftarrow$ infeasible, $S \leftarrow S_{\min}$, $W \leftarrow W_{\min}$
while $\lambda \geq T$ **do**
  $(\pi, z, \lambda) \leftarrow$ solve PESP-λ
  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
    $C_\lambda \leftarrow$ getCriticalCycle $((\pi, z), \lambda)$
    M1: Relax train line frequency of a critical service C_λ
    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 synchronization 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 (π, z), λ, R, S, W.

4.4 Computing initial solution for Algorithm 1

The defined PESP-λ model within Algorithm 1 performed well and reported small computation times in initial tests on smaller instances. However, after applying PESP-λ on the

bigger ones, the model had difficulties to find a feasible solution within a reasonable time. This would lead to a high computation times even for a single run of PESP-λ, which was not acceptable for our purposes. We then observed that initializing PESP-λ with a feasible solution (π, z) significantly speeds up computation times. Thus, we proposed a procedure to find an initial solution for PESP-λ by solving a feasibility problem PESP for a fixed nominal cycle time, i.e., the original PESP model. PESP is solved multiple times and each time with an increased value of nominal cycle time 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 nominal cycle time T_o, and the incremental step for the nominal cycle time δ_T. The output is a feasible solution for a certain T. If a feasible solution was not found in the first iteration of PESP, the stable solution for the given N and T_o does not exist. Thus, the instability should be resolved by Algorithm 1. In addition, the obtained T from the last iteration is used to strengthen the upper bound on λ in Constraint (7), $\lambda_{\max} \leftarrow T$. This new $\lambda_{\max}$ together with the initial solution (π_{feas}, z_{feas}) radically reduced computation times of PESP-λ.

Algorithm 2 Computing an initial solution for PESP-λ

Input: Line requests Q, $N = (E, A, T_o, l, u)$, δ_T
Initialize: $T \leftarrow T_o$, (π,z) $\leftarrow$ infeasible
while (π,z) infeasible **do**
  solve PESP for given T
  if no solution found **then**
    $T \leftarrow T + \delta_T$
  end if
end while
out: (π_{feas}, z_{feas}), $\lambda_{\max} \leftarrow T$

5 Experimental results

5.1 Scenarios

We evaluate the capabilities of the model for resolving timetable instability on a highly utilized railway network in the central Netherlands. In particular, 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.

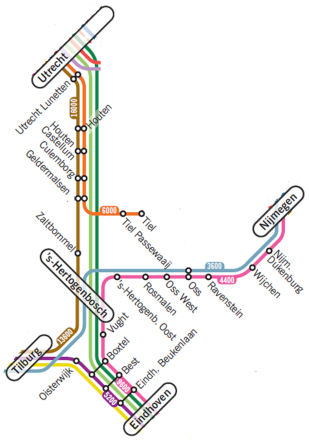

Figure 4: Line plan

Scenario	Notation	sc1	sc2
Number of lines	$\|Q\|$	20	20
Average frequency	f_q	3	2
Total number of train services	$\sum_{q=1}^{\|Q\|} f_q$	60	40
Number of events	$\|E\|$	456	304
Number of processes	$\|A\|$	1770	940
Nominal cycle time	T_o	1800	1200
Time step for Algorithm 2	δ_T	300	100
Upper bound for PESP-λ	$\lambda_{\max}$	3000	1800

Table 1: Input data for tested scenarios

For testing purposes, we used two target line plans that varied in frequencies and planned nominal cycle times. We applied Algorithm 1 on the realistic lines operating in the Netherlands and assumed the increase in their frequencies in order to mimic a possible higher future demand. Two scenarios, $sc1$ and $sc2$, are generated. Table 5.1 defines the characteristics of $sc1$ and $sc2$ giving the number of train lines, average frequency and total number of trains, number of events and number of processes and the upper bound for PESP-λ. Both scenarios consist of 20 train lines, while lines in sc1 (sc2) have frequency of three (two). The nominal cycle time for $sc1$ and $sc2$ equal 1800s and 1200s, respectively. For $sc2$, $T_o = 1200s$ would suggest an increase in provided level of service[2]. The upper bound $\lambda_{\max}$ was computed using Algorithm 2.

Table 1 defines initial parameters for PESP-λ model and the Algorithm 1 like nominal cycle time, as well as minimum, step and maximum values for relaxation measures. In addition, for M1, we chose to remove one critical train service at the time. Values for relaxing synchronisation (and running time) constraints are defined so to allow up to two consecutive iterations of each relaxation measure.

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	$W_{\min}$	0
M3 step	W_{step}	0.1
M3 maximum	$W_{\max}$	0.2

In the following sections, we first explain the effect of single measures on the behaviour of Algorithm 1. Second, Algorithm 1 is tested with applying only certain relaxation measures, or all three. Third, we compare effect of applying M1 on the critical line and on a random one.

[2]The current Dutch timetable operates with $T_o = 30$.

5.2 Explaining the effect of relaxation measures

We analyse the effect of single measures on computed timetable structure within Algorithm 1. For this purpose, we used scenario *sc2* with all three relaxation measures active M1, M2 and M3. We measured the solution efficiency as the total amount of time supplements in the iteration. The reported behaviours were observed in other test cases and we use the following one to show different impacts of relaxation measures to minimal cycle time and solution efficiency. Figure 5 shows the convergence of a minimal cycle time and changes in time supplements, i.e., solution efficiency, through iterations until $\lambda < T$ was satisfied. Applied measure is denoted in each iteration by a different marker.

We observed that, in the first three iterations, M1 ($\times$ marker) is used consecutively since $\lambda \gg T$. Measure M1 usually provide a reduction of the minimal cycle time λ. Once, λ is not significantly bigger than T ($iter = 4$), other measures are considered. Afterwards, relaxing frequencies was used again at iteration eight, when other two measures were relaxed maximally ($S = S_{\max}$ and $W = W_{\max}$) and no further improvements was possible. While doing so, S and W were reset to their initial values $S_{\min}$ and $W_{\min}$.

Measure M2 ($\bullet$ marker) started two times in total, that is in iterations four and nine. The values of S_{step} and $S_{\max}$ allowed maximally two consecutive iterations with M2. The first time applied each time, in iteration four (also holds for nine), M2 generated a solution with smaller λ and better efficiency. Then, in the following iteration five, no improvement on λ is reported compared to the previous one, but M2 reduced time supplements and thus improved the efficiency. On the other hand, in iteration 10, only slight improvement of λ is achieved. In this case, it negatively affected the solution's efficiency by generating more time supplements. Measure M2 does not always provide better λ because the critical cycle C_λ may not always include synchronization arcs. For example, C_λ may consist of multiple headways and/or running times only. However, relaxing synchronization times may positively impact the solutions efficiency.

Measure M3 ($\blacktriangle$ marker) was applied after maximally relaxing synchronization constraints, i.e., in iterations six and 11. The W_{step} and $W_{\max}$ allowed maximally two consecutive iterations with M3. The first time M3 was used, i.e., in six, (compared to five) a better λ is found at the expense of relaxed running times. In iteration six, cycle time was 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 bettter efficiency. This may be counter-intuitive that by allowing more time supplements we generate more efficient solution. Nevertheless, this can easily be due the the fact that by allowing more time supplements, only certain and limited number of arcs was relaxed (i.e., used more time supplements given), which allowed that other arcs use even less time supplements and resulted in 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 existed in N with the same cycle time. (Note that this was not the case in Figure 5).

Overall, measure M1 relaxes the solution more radically and while $\lambda \gg T$, then 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 maximums and still stable solution was found, the algorithm tries to relax frequencies again while resetting S and

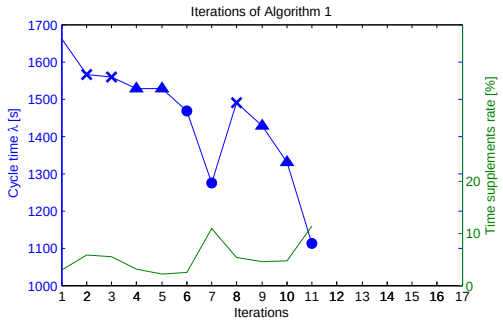

Figure 5: Results for a single run of Algorithm 1 with M1, M2 and M3 – scenario $sc2$

W to the strictest values. Measure M3 tend to provide bigger solution improvements than M2. Also, M3 seems to improve a solution more often, since may be due to the fact that there are more running arcs compared to sync arcs in N, hence, more possibility to use the exploit the certain relaxation measure. A solution obtain by relaxing M2 and M3 may both increase and decrease solution's efficiency in subsequent iterations. The efficiency is often increased when the smaller λ is found, while is decreased when λ stayed unchanged. The latter happens due to more flexibility and bigger search space for PESP-λ, while former is due to smaller search space imposed by smaller λ.

5.3 Results for considered scenarios

We tested four variants of Algorithm 1 for finding a stable and feasible timetable structure applying different sets of relaxing measures for both scenarios. In particular, the first set uses only M1, the second set considers M1 and M2, and the third set uses all three relaxations M1, M2 and M3. As a fourth one, we used an alternative version of M1 with updating a random train line instead of a critical line and refer to this relaxation as M1random, while M2 and M3 were not applied.

Initially, we also considered a variant that included only measures M2 and M3. For the two considered scenarios $sc1$ and $sc2$, this variant did not generate stable timetable structures. Instead, the Algorithm 1 terminated when both relaxation parameters S and W reached their maximum values, i.e., after four iterations. Obtained λ equalled 2723 s and 1405 s for $sc1$ and $sc2$, respectively. Thus, the results for his variant were not included in the further discussion.

Table 3 reports results of the developed algorithm for resolving timetable instability for the given test network and input data from Table 2. For the considered four combinations of Algorithm 1, the table gives, for each scenario (Sc) the number of iterations needed for each $\lambda < T$ solution (iter), obtained λ in seconds, number of times each measure was applied (MeasCount), total time supplements (TotalSup) in seconds, average running time supplements (AvgRTsupp) in %, relaxed sync (S) and running times (W) and computation times (CPU) in seconds. Three values in column MeasCount depict the number of times measures M1, M2 and M3 are applied, respectively.

The initial cycle time computed by Algorithm 2 equalled 3000 and 1800 s for $sc1$ and $sc2$, respectively. M1random in both scenarios needed the most iterations, 22 and 18, respectively. It was clearly outperformed by all other variants that consider applying M1 on the critical circuit. The stable timetable structure for the first three variants was obtained

Table 3: Results for combination of used measures for resolving the timetable instability

Sc	Measures	T_o	iter	λ	MeasCount	TotalSup	AvgRTsup	S	W	CPU
sc1	M1	3000	16	1705	16 / 0 / 0	7537	2,44	0	0	4900
sc1	M1+M2	3000	20	1618	11 / 8 / 0	15604	8,27	120	1,3	6100
sc1	M1+M2+M3	3000	18	1450	10 / 4 / 3	19900	15,18	120	1,4	5500
sc1	M1random	3000	22	1705	21 / 0 / 0	11545	6,46	0	1,3	6700
sc2	M1	1800	8	1165	7 / 0 / 0	7480	5,50	0	1,3	2500
sc2	M1+M2	1800	10	1147	3 / 6 / 0	10192	6,04	120	1,3	3100
sc2	M1+M2+M3	1800	11	1113	3 / 4 / 3	14517	11,38	120	1,4	3400
sc2	M1random	1800	18	1139	17 / 0 / 0	3837	2,44	0	1,3	5500

in at most 20 and 10 iterations. Even though M1 needed less iterations (and CPU time), it relaxed more train lines than the other two variants, M1+M2 and M1+M2+M3. This is due to the fact that the first one relaxes lines in each iteration, while the other may relax also synchronisation and running time constraints. Total time supplement, i.e., solutions efficiency is the smallest for the the first variant, when no other relaxation is possible. On the other hand, when timetable design parameters were relaxed, average running time supplements for M1+M2 reach 8.27% and 6.04% and for M1+M2+M3 they extend to 15.18% and 11.38%. These results are easy to understand, since the stable solutions are obtained with $S = S_{\max}$, and for variants M1+M2+M3, additionally with $W = 1, 4$. Finally, computation times were proportional to the number of iterations and ranged from 2500s to 6700s over all cases.

Figure 6 shows the improvements of λ through iterations for four variants of Algorithm 1. For $sc1$, M1random clearly performed the worst and during iterations, it reported unchanged λ over multiple iterations. This is caused by the random choice of train line to relax. On the other hand, variant with M1 improves λ in (almost) every new iteration. This proves the benefits of considering critical circuit and lines on it when selecting a candidate line.

For scenario $sc1$, we observe a similar behaviour of all three variants in the first 12 iterations while $\lambda \gg T$. Also, some iterations without improvement of λ could be due to multiple critical circuits of the same length, and by fixing only one train line in each iteration, the other one, still remains unchanged. However, small variations in iterations 1 to 12 may occur due to random parameter for choosing a critical train line (the critical line is chosen randomly if more than two lines have the same observed priorities). Once λ becomes smaller than $1.3 \cdot T$ then other relaxation measures are used in M1+M2 and M1+M2+M3 variants, and solution improvements gets different patterns. It seems that $W = W_{\max}$ generates a significant improvement ('drop') of the solution which can be clearly seen in (at least) the last iterations of M1+M2+M3. For scenario $sc2$, M1random again underperformed and was dominated by all other variants. Here, we see that initial solution (equal for all variants) is much closer to the nominal cycle time T, and so, measure M2 (and M3) was applied for the first time in the third iteration. Again, M1+M2+M3 reports a significant drop of λ in seventh iteration for relaxed S and W to $S_{\max}$ and $W_{\max}$. Since a stable solution was not found, S and W were reset to $S_{\min}$ and $W_{\min}$ and another critical train line was relaxed in the next iteration.

Measure M1 shows only strictly decreasing behavior as by relaxing a frequency on a critical line, the resulting λ in the following iterations may only be smaller (or equal due to multiple critical circuits). Since only measure M1 was applied, each iteration has exactly

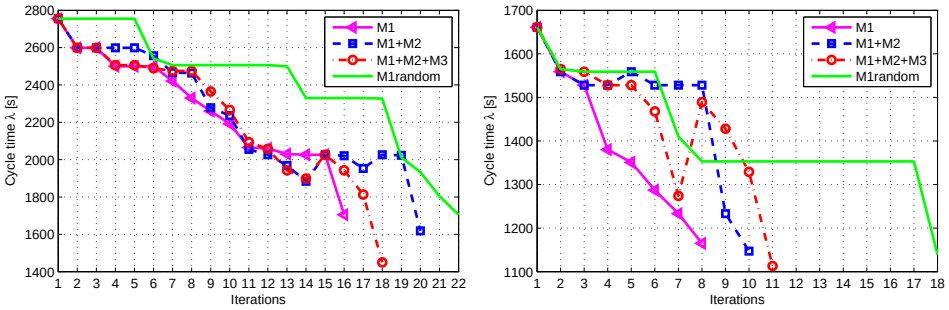

Figure 6: Results for combination of used measures for resolving the timetable instability for $sc1$ (left) and $sc2$ (right)

one train service less than the previous one, and the number of iterations equals the number of removed train services q_{crit}. On the other hand, combinations M1+M2 and M1+M2+M3 have 'drops' in values of λ, i.e., local improvements. This is due to local searches of M2 (and M3) for the same line plan by only relaxing synchronization (and running) constraints. Also, the latter combination that includes M3 has bigger drops as it observes more significant improvements due to additional use of M3. By applying M2 (and M3), we may find a final solution in less iterations. Finally, M1random has the similar behavior as M1 but a slower improvement, which was expected as we applied M1 on a random train line.

6 Conclusions

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

We observed that if a transport demand is significantly bigger than what 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 synchronization constraints does not always generate a better solution, while relaxing running times seems more effective than relaxing synchronization constraints. By relaxing synchronisation and running time constraints, we may need more iterations to find a stable solution. More importantly, such a solution may have less train services removed due to more flexibility of the PESP-λ. Therefore, it would be preferable to use all available measures in order to generate a stable timetable structure and minimize the unsatisfied demand.

Future research could be addressed in several directions. First, current or forecasted transport demand can be considered in the timetabling problem. Second, identifying additional combinations of constraint 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. Second, 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. Third, a more robust heuristic that integrates multiple measures in each iteration and

more advanced search techniques could be developed. These improvements would make the model 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 satisfaction of 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

[1] Bergmann DR. (1975) Integer programming formulation for deriving minimum dispatch intervals on a guideway accommodating through and local public transportation services, Transportation Planning and Technology, 3(1), 27-30.

[2] Bešinović N, Quaglietta E, Goverde RMP. (2016) Microscopic models and network transformations for automated railway traffic planning. Computer-Aided Civil and Infrastructure Engineering, 32(2), 89-106.

[3] Bešinović N, Quaglietta E, Goverde RMP, Roberti R. (2016). An integrated micromacro approach to robust railway timetabling. Transportation Research Part B: Methodological, 87, 14-32.

[4] Bešinović N, Goverde RMP. (2016) Two stage stability-for-robustness approach for periodic railway timetabling. Working paper.

[5] Cacchiani V, Toth P. (2012) Nominal and Robust Train Timetabling Problems, European Journal of Operational Research, 219 (3), 727737.

[6] Caimi G, Fuchsberger M, Laumanns M, Schpbach K. (2011) Periodic railway timetabling with event flexibility, Networks, 57(1), 3-18.

[7] The Digital Railway Programme, http://digitalrailway.co.uk, Visited on 28.09.2016.

[8] Goverde RMP. (2007) Railway timetable stability analysis using max-plus system theory, Transportation Research Part B: Methodological, 41(2), 179-201.

[9] Goverde RMP. (2010) A delay propagation algorithm for large-scale railway traffic networks, Transportation Research Part C: Emerging Technologies, 18(3), 269-287.

[10] Hansen IA, Pachl J. (2014) Railway timetabling and operations, Second edition, Eurail, Hamburg.

[11] 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.

[12] 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, Computers & Industrial Engineering, 66(1), 171-185.

[13] Kümmling M, Großmann P, Opitz J, WeißR, & Nachtigall, K. Extraction of Significant Conflicts in Periodic Timetabling, The 13th Conference on Advanced Systems in Public Transport (CASPT 2015) 19-23 July 2015, Rotterdam, The Netherlands.

[14] 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.

[15] Kroon LG, Peeters LW. (2003). A variable trip time model for cyclic railway timetabling. Transportation Science, 37(2), 198-212.

[16] Kroon LG, Peeters LW, Wagenaar JC, Zuidwijk RA. (2013). Flexible connections in PESP models for cyclic passenger railway timetabling. Transportation Science, 48(1), 136-154.

[17] Kroon LG, Maroti G, Retel Helmrich MJ, Vromans MJCM, Dekker R. (2008) Stochastic improvement of cyclic railway timetables. Transportation Research Part B, Methodological, 42 (6), 553-570.

[18] Liebchen, C. (2008). The first optimized railway timetable in practice. Transportation Science, 42(4), 420-435.

[19] Liebchen C, Peeters L. (2009) Integral cycle bases for cyclic timetabling, Discrete Optimization, 6(1), 98-109.

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

[21] Nachtigall K, Opitz J. (2008). Solving periodic timetable optimisation problems by modulo simplex calculations. In OASIcs-OpenAccess Series in Informatics (Vol. 9). Schloss Dagstuhl-Leibniz-Zentrum fr Informatik.

[22] Peeters LWP. (2003) Cyclic railway timetable optimization, PhD thesis, Erasmus Universiteit Rotterdam, Rotterdam, The Netherlands.

[23] Petering MEH, Heydar M, Bergmann DR (2015). Mixed-Integer Programming for Railway Capacity Analysis and Cyclic, Combined Train Timetabling and Platforming. Transportation Science.

[24] Polinder GJ (2015). Resolving infeasibilities in the PESP model of the Dutch railway timetabling problem. Report, Delft University of Technology.

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

[26] Serafini P, Ukovich W. (1989) A mathematical model for periodic event scheduling problems, SIAM Journal on Discrete Mathematics, 2, 550-581.

[27] Sparing D, Goverde RMP, (2017) A cycle time optimization model for generating stable periodic railway timetables, Transportation Research Part B: Methodological, 98, 198-223.

[28] Stanley P. (2011) ETCS for Engineers, Eurail Press, Hamburg.