



An iterative heuristic for passenger-centric train timetabling with integrated adaption times

Gert-Jaap Polinder^{a,b}, Valentina Cacchiani^c, Marie Schmidt^{a,b,*}, Dennis Huisman^{b,d,e}

^a Rotterdam School of Management, Erasmus University, P.O. Box 1738, 3000DR Rotterdam, The Netherlands

^b Erasmus Center for Optimization in Public Transport (ECOPT), Rotterdam, The Netherlands

^c DEL, University of Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy

^d Econometric Institute, Erasmus School of Economics, Erasmus University Rotterdam, P.O. Box 1738, 3000DR Rotterdam, The Netherlands

^e Process quality and Innovation, Netherlands Railways, P.O. Box 2025, 3500HA Utrecht, The Netherlands

ARTICLE INFO

Keywords:

Public transportation planning
Periodic timetabling
Heuristic
Integration of timetabling and passenger routing
Adaption time

ABSTRACT

In this paper we present a method to construct a periodic timetable from a tactical planning perspective. We aim at constructing a timetable that is feasible with respect to infrastructure constraints and minimizes total perceived passenger travel time. In addition to in-train and transfer times, our notion of perceived passenger time includes the adaption time (waiting time at the origin station). Adaption time minimization allows us to avoid strict frequency regularity constraints and, at the same time, to ensure regular connections between passengers' origins and destinations. The combination of adaption time minimization and infrastructure constraints satisfaction makes the problem very challenging.

The described periodic timetabling problem can be modelled as an extension of a Periodic Event Scheduling Problem (PESP) formulation, but requires huge computing times if it is directly solved by a general-purpose solver for instances of realistic size. In this paper, we propose a heuristic approach consisting of two phases that are executed iteratively. First, we solve a mixed-integer linear program to determine an *ideal* timetable that minimizes the total perceived passenger travel time but neglects infrastructure constraints. Then, a Lagrangian-based heuristic makes the timetable feasible with respect to infrastructure constraints by modifying train departure and arrival times as little as possible. The obtained feasible timetable is then evaluated to compute the resulting total perceived passenger travel time, and a feedback is sent to the Lagrangian-based heuristic so as to possibly improve the obtained timetable from the passenger perspective, while still respecting infrastructure constraints. We illustrate the proposed iterative heuristic approach on real-life instances of Netherlands Railways and compare it to a benchmark approach, showing that it finds a feasible timetable very close to the ideal one.

1. Introduction

For a passenger, a public transportation system is attractive if it offers a route from their origin to their destination with low travel time, no or few transfers, and at the 'right' time, i.e., with the desired departure (or arrival) time. As the desired departure time differs from passenger to passenger, a good public transport system offers several alternative routes per origin–destination (OD) pair at various departure times, so that no passenger has to deviate too much from their desired departure time.

In state-of-the art models on periodic timetabling, this quality requirement is covered by the inclusion of 'regularity' (or 'synchronization') restrictions among trains that belong to the same line, or

share considerable part of the physical route. However, in a dense line network, where many different lines share (part of) their physical route, and in addition many passenger routes require a transfer, it is far from obvious which trains should be synchronized to obtain an *ideal* timetable. Polinder et al. (2021) proposes a model that includes adaption time (the time from desired departure time until the scheduled departure) to be able to drop pre-imposed regularity constraints in the construction of an ideal timetable. Polinder et al. (2021) approach periodic timetabling from a strategic perspective and therefore do not consider any infrastructure constraints.

However, for railway transportation, the infrastructure imposes severe constraints on the scheduling, so that an *ideal* timetable, computed

* Corresponding author at: Rotterdam School of Management, Erasmus University, P.O. Box 1738, 3000DR Rotterdam, The Netherlands.

E-mail addresses: polinder@rsm.nl (G.J. Polinder), valentina.cacchiani@unibo.it (V. Cacchiani), schmidt2@rsm.nl (M. Schmidt), huisman@ese.eur.nl (D. Huisman).

<https://doi.org/10.1016/j.cor.2022.105740>

Received 19 April 2021; Received in revised form 23 November 2021; Accepted 31 January 2022

Available online 12 February 2022

0305-0548/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

without considering infrastructure constraint, is in many cases not feasible. In particular on dense networks like the Dutch one, trains of different lines compete for infrastructure utilization. An extension of the current infrastructure would be very costly and time consuming.

Therefore, in this paper we present an approach to solve the *Passenger-Oriented Timetabling (POT)* problem that aims at finding a timetable which is feasible with respect to infrastructure, and minimizes the passengers' total perceived travel time, which is defined as the travel time on the 'best' route for each passenger including adaption time. The problem is defined as follows:

Passenger-Oriented Timetabling (POT): Given an infrastructure network with stations and tracks connecting them, and a line plan, specifying line routes, stopping patterns and frequencies: find a periodic timetable including all or a subset of the trains that satisfies the headway constraints induced by the infrastructure network and that minimizes total perceived travel time, where we assume that passengers will travel on shortest routes according to perceived travel time.

We consider the infrastructure constraints on a macroscopic level. That is, we consider railway stations with a number of tracks connecting them. In macroscopic timetabling, headway constraints impose a minimum time difference between trains that share part of the tracks to avoid crossings and overtakings that are not possible due to the infrastructure and to enforce a minimum safety distance between trains running on the same track. Further details like block sections and signalling systems are not relevant at the tactical planning stage and can be included in a later planning stage (Radtke, 2014).

In the context of strategic timetabling, Polinder et al. (2021) combine a Mixed Integer Linear Program (MILP) formulation of a PESP model with an approach for modelling total perceived travel time. A similar approach would be possible for the POT problem, adding infrastructure constraints (which are not considered in Polinder et al., 2021) as PESP constraints, and using indicator variables to model the cancelling of trains (see Appendix A.4). However, the resulting MILP is very difficult to solve (as is shown in Section 5).

For this reason, in this paper we propose an iterative approach that combines (extended versions of) two existing approaches. First, we compute an *ideal* timetable, that is: a timetable that does not need to respect infrastructure constraints, using the method proposed in Polinder et al. (2021) for strategic timetabling. Secondly, we transform this ideal timetable into a feasible timetable, that is: let it satisfy the infrastructure constraints, using an extension of the Lagrangian heuristic (LH) proposed in Cacchiani et al. (2010) with the goal to find a feasible timetable that stays as close as possible to the ideal timetable. As an additional step, we compare the resulting feasible timetable to the ideal one and evaluate how the changes influence the quality of the timetable. Based on this, we provide feedback to the Lagrangian heuristic to improve the quality of the newly found timetable.

Our contribution in this paper is twofold: First, we propose an iterative heuristic approach to construct a periodic timetable that satisfies infrastructure constraints and minimizes the total perceived travel time (that includes the adaption time). The combination of perceived travel time minimization and infrastructure constraints satisfaction to make the timetable feasible makes the problem very challenging. Second, we illustrate our approach on three case studies on the Dutch railway network providing insights on critical points in the network and trade-offs in timetable construction. We show that our algorithmic approach performs better than the alternative of directly incorporating infrastructure constraints in the integer program for timetabling, and that it converges to a feasible timetable very close to the ideal one.

The remainder of this paper is organized as follows. In Section 2, we give an overview on research that is related to and relevant for this study. In Section 3 we introduce and define the POT problem in detail. Afterwards, we describe our iterative approach to solve this problem in Section 4. We test our approach on three case studies on the Dutch railway network in Section 5. Finally, the paper is concluded in Section 6.

2. Related work

Timetabling with PESP. The problem of finding a train timetable is extensively studied in the literature. Often, the timetabling problems are modelled as the problem of assigning times to nodes ('events') in a graph ('event-activity network') where arcs ('activities') represent the time constraints. This formulation makes periodic timetabling a special case of the Periodic Event Scheduling Problem (PESP, Serafini and Ukovich, 1989), which can be used for various applications with periodically recurring events. Details how PESP can be applied in railway timetabling problems can be found in Odijk (1996), Peeters (2003), Liebchen and Möhring (2007). Essentially, PESP is a feasibility problem, as its task is to find a feasible solution, satisfying a set of restrictions. Often PESP is extended by an objective function that is to be optimized (cf. Peeters, 2003; Liebchen, 2008; Caimi et al., 2017). These objective functions can cover a number of subjects, like optimizing the customer satisfaction, minimizing the costs of the operator, finding a timetable that is as close as possible to an infeasible input timetable, or computing a delay-resistant timetable (Cacchiani and Toth, 2012; Lusby et al., 2018). The optimization version of PESP is normally much more computationally challenging than solving the feasibility problem. Approaches to solve such problems cover various techniques, like integer-programming (Liebchen, 2008; Nachtigall, 1994; Liebchen and Peeters, 2009), a modulo-simplex heuristic (Nachtigall and Opitz, 2008) or combining machine-learning with a SAT formulation (Matos et al., 2018). In our paper, we employ a PESP-based model for the first phase of our approach, in which the ideal timetable is computed based on the total perceived travel time.

Timetabling based on time-space graphs. Time-space graphs constitute an alternative graph-based modelling approach to event-activity networks. In these approaches, time is discretized, and a time-expanded network is used: nodes correspond to train departures and arrivals at specific time instants, and a path in the graph corresponds to a timetable. In these models, variables represent the choice of arcs (or paths) of this graph. Approaches based on these kind of models have mainly been used for aperiodic timetabling, although recent works have shown their effectiveness for periodic timetabling as well (Martin-Iradi and Røpke, 2021; Zhang et al., 2019).

Time-space graph models easily embed the running and dwelling time constraints within the graph definition, and allow the option of not scheduling some trains by assigning them a dummy path: this is particularly useful when a feasible solution with all trains scheduled does not exist, for example in highly congested networks. The drawback of these models is the size of the graph that can be extremely large for practical instances. For this reason, most approaches in this category solve the timetabling problem heuristically by decomposing it through column generation (Cacchiani et al., 2008, Martin-Iradi and Røpke, 2021) or Lagrangian relaxation (Brännlund et al., 1998, Caprara et al., 2002, Cacchiani et al., 2010, Zhang et al., 2019, Ait-Ali et al., 2020): indeed, computing a timetable for a single train corresponds to solving a shortest path problem, and can be efficiently done by dynamic programming algorithms.

In our paper, we extend the heuristic from (Cacchiani et al., 2010) to deal with periodic timetabling, and use it in the second phase of our approach, to make a given *ideal* timetable feasible.

Passenger-centric objective functions. There are several approaches to measure the quality of a timetable from the viewpoint of the passengers, and, as shown in Hartleb et al. (2019), the choice of the evaluation approach will have an impact on which timetables are considered to be 'good' and 'optimal'.

In the OR literature, most papers minimize total passenger travel time. The most simple models for measuring and optimizing travel time within a PESP approach minimize a function over the weighted durations of the activities in the timetabling instance, where weights represent the number of passengers using that activity (cf. Peeters,

2003). This relies on the assumption that it is a priori known on which activities passengers travel.

Schmidt and Schöbel (2015) propose a mixed-integer linear programming model that integrates (aperiodic) timetabling and passenger routing. Borndörfer et al. (2017) provide a similar, PESP-based model for the periodic case and study the impact of different routing assumptions. As PESP is already a challenging problem in itself, the integration of this problem with passenger routing makes it even more difficult to find good solutions. Schiewe and Schöbel (2020) propose an ‘applicable’ approach that relies on (heuristic) preprocessing and bound generation. Other approaches (Lübbe, 2009; Siebert and Goerigk, 2013) solve the problem iteratively: first passengers are routed through the network. Based on these fixed routes a timetable is computed. Then passengers are rerouted based on the timetable. This is repeated until a stopping criterion is met.

Martin-Iradi and Røpke (2021) propose a time–space-graph-based approach to find periodic timetables minimizing passenger travel time, and include frequency constraints to guarantee that trains of the same line are spread along the period. In a column generation approach that is designed to minimize the travel times of the trains, each feasible solution found during the process is evaluated with respect to the passenger travel time, and the best solution is kept. In Farina (2018), the same problem as in Martin-Iradi and Røpke (2021) is considered and modelled on a time–space graph, and a Large Neighbourhood Search algorithm is proposed.

The above-mentioned approaches have in common that they evaluate timetables based on the assumption that every passenger will choose the shortest route (with respect to (perceived) travel time) towards his destination, just as we do.

Including adaption time into passenger-centric objectives. Besides the travel time between departure at the origin and arrival at the destination, also the number of travel options between origin and destination and their timing play a crucial role in evaluating timetables from a passenger perspective (de Dios Ortúzar and Willumsen, 2011). E.g., a timetable with four travel options between origin and destination, offered every fifteen minutes, would most likely be preferred to a timetable where there is just one such option (or four, all departing at the same time), even if in the latter case the travel time is slightly shorter.

Focusing in the evaluation of a timetable solely on passenger travel time, measured from departure at the origin, neglects the effect that the spread of travel options over time has on the quality of a timetable. This can be overcome by including *adaption time* in the objective function, while making an assumption on the distribution of ‘desired departure times’ of passengers over time.

There are several publications on timetabling on lines and corridors, where adaption time is explicitly included in the objective function. Often, ‘adaption time’ is called ‘waiting time’ in this context. We use the term ‘adaption time’ throughout this literature review, also when referring to literature where the authors use the term ‘waiting time’, to avoid confusion with the waiting time at transfers. For single rail rapid transit lines, Barrena et al. (2014a,b) propose, respectively, an exact and an adaptive large neighbourhood search minimizing adaption time. A single rail line is also considered in Zhu et al. (2017), where a bi-level model is proposed: the upper level model determines the train headway times to minimize the total passenger perceived costs (given by adaption time, in-vehicle time and penalty costs associated with arriving at the destination outside the desired interval), while the lower level determines passenger arrival times at their origin stations. A genetic algorithm is used to solve it. Yin et al. (2017) proposed an integrated approach to determine train schedules and speed profiles with the aim of minimizing energy consumption and passenger adaption time. A Lagrangian based algorithm is developed for solving it for a bidirectional urban metro line. A rail corridor is considered in Niu et al. (2015), where a quadratic model to determine train timetables based on given time-varying passenger demand data is proposed.

Wang et al. (2015) propose a very detailed event-driven model for timetabling on urban networks with the objective to minimize a weighted sum of travel time (including adaption time) and energy consumption. Their solution approach is based on sequential quadratic programming and a genetic algorithm, and tested on a small network with two cyclic lines.

Instead of including adaption time into the objective function, Gattermann et al. (2016) group passengers into time slices, and add a penalty to the objective function if passengers do not depart in the respective time slice. They propose a (non-linear) PESP-based mathematical programming formulation, and transfer it to a SAT formulation to solve it. While the model allows to group passengers into (pre-defined) time slices and penalize deviation from the respective time slice, a heuristic to include adaption time, only one time slice (that spans the whole period) is used in the numerical experiments reported in Gattermann et al. (2016).

Polinder et al. (2021) consider timetabling in the strategic railway planning phase. Like in the POT problem, they aim at finding a periodic timetable that minimizes perceived travel time (a weighted sum of in-train, transfer, and adaption time and transfer penalties) under the assumption that passenger demand is uniformly distributed over the period. However, in contrast to the POT problem, they do not consider infrastructure constraints, arguing that these are not relevant in the strategic planning phase. In this paper, we use the approach developed in Polinder et al. (2021) for strategic timetabling to compute an *ideal* timetable in the first phase of our solution approach. It is therefore described in more detail in Section 4.1.

Compared to the existing literature on passenger timetabling, we include the adaption time minimization in the objective, instead of having strict regularity constraints in order to gain flexibility. In addition, we consider a large railway network while most works tackle the problem on a single line or corridor.

3. Problem description

3.1. Input

The timetable that is to be designed is based on three items: First, the infrastructure network on which the trains operate. Second, an origin–destination matrix representing passenger demand. Third, a line plan that specifies line routes and frequencies.

As usual in tactical planning, we consider the infrastructure at the macroscopic level. That means that our infrastructure network consists of stations as nodes and tracks between the stations as arcs. Furthermore, we have estimates of driving times on tracks and dwell times at stations, and headway times between trains that consecutively use the same tracks.

Passenger demand is given in the form of an origin–destination (OD) matrix $(d_k)_{k \in \mathcal{OD}}$. For each OD-pair $k \in \mathcal{OD}$, the corresponding matrix entry d_k represents the number of passengers who want to travel from the origin to the destination in one period.

A line plan specifies a set of train lines that are to be operated on the given infrastructure network. Each train line consists of a route through this network, a list of stations where the train stops (a stopping pattern) and a frequency that specifies how often the line is operated per period (e.g., every hour). We assume that all lines are operated in both directions. Note that in the line planning phase, no timetable is known yet. Therefore, while line planning can take into account constraints on the infrastructure utilization, and on eligible frequencies, it is not ensured that there exists a feasible timetable where all trains specified in the line plan can be operated. Therefore, we allow our method to cancel trains if necessary.

3.2. Timetabling constraints

Based on the infrastructure network, the demand encoded in the OD-matrix $(d_k)_{k \in \mathcal{OD}}$, and the line plan, we aim at finding a periodic *timetable* π that assigns a timestamp to each arrival and departure of trains at/from every visited station during the considered period.

To ensure operational feasibility, the time difference between a departure of a train and its subsequent arrival needs to be at least the minimum driving time, and is only allowed to exceed it up to a certain extent. The time between an arrival and the subsequent departure of a train from a station must be equal to or higher than the minimum dwell time and is constrained by a maximum dwell time. When trains use the same infrastructure, headway times need to be respected. Furthermore, we require the timetable to be periodic, i.e., events reoccur every time period.

These constraints can be encoded by using the PESP approach (Serafini and Ukovich, 1989): in this case, departure and arrival events are represented as nodes in a so-called *Event-Activity Network*. The arcs of this network, called *activities*, represent driving and dwelling of the trains and headway constraints. In addition, *transfer activities* that represent the transferring of passengers from one train to another within one station, can be added to the PESP model. While these do not impose operational constraints on the timetable, they allow us to represent passenger routes as paths in the event-activity network. Details on how to encode standard timetabling constraints as PESP constraints can be found in Appendix A.1.

Another way to encode timetabling constraints is by using a *time-space graph*, in which each node corresponds to a potential departure or arrival time of a train from/at a station along a track, and arcs represent driving and dwelling between departures and arrivals. In this case, a *Time-index ILP Formulation* is used, in which the headway constraints are expressed as incompatibility between arcs of the time-space graph. An example of this graph is shown in Appendix A.2, while the Time-index ILP formulation will be introduced in Section 4, since it will be employed in our solution approach (and also requires more detail that will be presented in that section).

3.3. Quality of the timetable with respect to perceived passenger travel time

Among all operationally feasible timetables, we aim to find one that optimizes the total perceived passenger travel time.

To evaluate the quality of a timetable by computing the corresponding total perceived travel time, we first distribute the passenger demand given by the OD-matrix \mathcal{OD} on the network and compute total perceived travel time. This evaluation is based on a number of assumptions (see Polinder et al., 2021):

1. Passenger numbers for one period T are given in the OD-matrix $(d_k)_{k \in \mathcal{OD}}$. This demand is independent of the timetable.
2. Passenger demand per OD-pair is distributed uniformly over the period T . I.e., every time unit (in our experiments: every minute) $\frac{d_k}{T}$ passengers would like to depart from the origin station of OD-pair k to travel to the destination station of OD-pair k . The rationale behind this assumption is that the timetable is usually constructed a number of years to six months before the actual day of operation, and we cannot expect that the demand distribution over the period is known accurately at that time, in particular since we assume a relatively short period length of one hour.
3. Train capacity constraints do not play a role in the route choice of passengers. In fact, it is common to neglect capacity considerations in timetable design. Often these considerations are explicitly addressed in the next phase of public transport planning, the rolling stock planning, where capacity requirements can be met (to a certain extent) by assigning a sufficient number of train units to the train trips scheduled in the timetable.

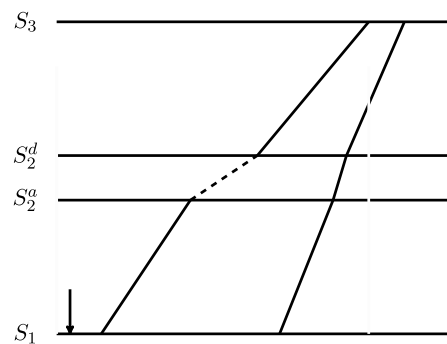


Fig. 1. Time-space diagram with two possible journeys from S_1 to S_3 .

4. Passengers are rational in the sense that they choose a route with minimum perceived travel time, based on the timetable and their desired departure time.
5. Perceived travel time is computed as a weighted sum of several components that are listed below. The weights are equal for each passenger.

The components of the perceived travel time are as follows:

In-train time The time the passenger actually spends in the train, both when the train is driving and when it dwells at a station.

Adaption time The time difference between the desired departure time of the passenger and the moment the train departs that brings him to his destination. The adaption time is weighted by a factor γ_w .

Transfer time The time a passenger has to spend on some station to transfer from one train to another. The transfer time is weighted by a factor γ_s .

Transfer penalty If the passenger needs to transfer from one train to another, a penalty of γ_l is added for each transfer, compare (de Keizer et al., 2015).

The different components are illustrated in Fig. 1, which displays a time-space diagram. A passenger arrives at S_1 at the time indicated by the arrow, and wants to travel to S_3 . He can either take the first departing train to S_2 , transfer (indicated by the dashed line) there to the train to S_3 , or he can take the direct train, and he then arrives later at S_3 . If the passenger takes the first route, he has a short adaption time, which is beneficial for his perceived travel time, but he needs a transfer, thus leading to a transfer penalty. He could also wait for the later train, which is a direct train, but then the adaption time is larger. Depending on the weights for the adaption time, transfer time and transfer penalty, he will choose a route that has the lowest perceived travel time.

Note that in-train time, transfer time, and transfer penalty are characteristics of the route and do not depend on the desired departure time of a passenger. We therefore refer to the weighted sum of these as *perceived route length*.

To compute the total perceived travel time for timetable π efficiently, we follow the approach described in Polinder et al. (2021):

We use a precomputed set of passenger routes. A route for passenger k is a sequence of drive, dwell, and transfer activities that form a path from the origin of passenger k to the destination of passenger k . We use the method described in Warmerdam (2004) to precompute the routes. Details can be found in Appendix A.3.

We assume that each passenger will take the route with lowest perceived travel time, based on his desired departure time. That can be the route that departs earliest after the desired departure time, but it can also be a route that departs later (e.g., if that route has significantly

shorter travel time or contains less transfers and is therefore preferable despite the higher adaption time, as illustrated in Fig. 1).

For each OD-pair k , we call the set of departure events at the origin of k that belong to a precomputed route *relevant* departure events. This set is denoted by V^k . We divide the period T based on the scheduled times π_i of relevant departure events i . Based on the assumption that perceived route length is the same for all passengers, we know that all passengers whose desired departure time falls between two relevant departure events v' and v will choose the same route (although their adaption time can differ, and therefore also their perceived travel time for the chosen route). Therefore, it is sufficient to compute for each OD-pair and each corresponding relevant departure event the route with lowest perceived travel time from departure time π_v on. We denote the perceived travel time of this route (measured from π_v on) as Y_v^k .

For each OD-pair k , we denote by $L_v^k := (\pi_v - \pi_{v'}) \pmod T$ the time before departure event v , but after the previous relevant departure event v' . Note that this calculation is made *modulo* the time period T . That is, also the 'first' relevant event of a period has a preceding relevant departure event (namely the last relevant departure event of the previous period) and the time difference between the two events is correctly calculated and assigned to L_v^k . Due to the assumption of uniformly distributed passenger arrivals, the number of passengers of OD-pair k arriving in this interval is $\frac{d_k}{T} \cdot L_v^k$. The expected waiting time (and hence the adaption time) of these passengers is $W_v^k = \frac{L_v^k}{2}$, based on the assumption of uniformly distributed customer arrivals. That is, the average perceived travel time for a passenger of OD-pair k , arriving between departure events v' and v , is $\gamma_w \cdot W_v^k + Y_v^k$. To compute the overall average perceived travel time for a passenger of OD-pair k we consider the total number of passengers $\frac{d_k}{T} \cdot L_v^k$ of OD-pair k arriving in each interval L_v^k , and consider all intervals in T (defined by the scheduled times of the relevant events $v \in V^k$), and then divide the expression by d_k to have the average value for a single passenger:

$$R_k(\pi) = \frac{1}{d_k} \sum_{v \in V^k} \frac{d_k}{T} \cdot L_v^k \cdot (\gamma_w \cdot W_v^k + Y_v^k) = \frac{1}{T} \sum_{v \in V^k} L_v^k \cdot (\gamma_w \cdot W_v^k + Y_v^k) \quad (3.1)$$

If, due to train cancellations, there is no route from origin to destination of OD-pair k , we set $R_k(\pi) := M$, where M is a (high) penalty value. In the remainder of this paper, we call the average perceived travel times of passengers of OD-pair k , weighted with the number of passengers on this OD-pair, $d_k \cdot R_k(\pi)$, the *evaluation contribution* of OD-pair $k \in \mathcal{OD}$.

Note that the evaluation contribution can also simply be interpreted as summing up the perceived travel time of all passengers of an OD-pair.

To evaluate the timetable we sum up the evaluation contributions of the OD-pairs and obtain the total perceived travel time

$$\sum_{k \in \mathcal{OD}} d_k \cdot R_k(\pi). \quad (3.2)$$

3.4. The passenger-oriented timetabling problem

To find a timetable that is optimal with respect to the minimization of function (3.2), we have to determine the timetable, the passenger routes, and the corresponding total perceived travel time simultaneously. We can summarize the POT problem as

$$\text{Minimize}_{\pi} \sum_{k \in \mathcal{OD}} d_k \cdot R_k(\pi) \quad (3.3)$$

$$\text{Such that } \pi \text{ is a feasible periodic timetable} \quad (3.4)$$

$$\text{passengers take best routes with respect to } \pi \quad (3.5)$$

$$R_k(\pi) \text{ is the average perceived travel time of OD-pair } k \quad \forall k \in \mathcal{OD} \quad (3.6)$$

Note that (3.4) means that all the timetabling constraints described in Section 3.2 are respected. In addition, by optimizing with respect to total perceived travel time, which includes the adaption time, we automatically obtain a timetable in which departures of trains that

serve the same line or whose corresponding lines share a substantial part of the physical route are spread relatively evenly over the period. For this reason, we can exclude additional 'synchronization constraints' that are often imposed in other timetabling models to synchronize these train runs.

4. Solution approach

In this section, we present our iterative heuristic approach. A graphical overview is shown in Fig. 2.

The approach consists of decomposing the POT problem into two subproblems: *make an ideal timetable* that determines an ideal timetable that minimizes the total passenger perceived travel time but neglects the infrastructure constraints, and *make a feasible timetable* that modifies the ideal timetable to make it satisfy all the infrastructure constraints, with the aim of applying as few as possible changes. These steps are solved in sequence, and a feedback mechanism on the second step allows to iteratively improve the feasible timetable in terms of the total passenger perceived travel time.

More precisely, in the first step, we construct an 'ideal' timetable using the solution approach for the strategic timetabling problem from Polinder et al. (2021). This approach is summarized in Section 4.1. An extension of the Lagrangian heuristic (LH) from Cacchiani et al. (2010) is used, in the second step, to modify the ideal timetable to make it feasible with respect to infrastructure constraints, while staying as close as possible to the ideal timetable. This approach is summarized in Section 4.3. This algorithm is chosen because it runs in relatively short computing times, and was designed to start from a given ideal (infeasible) timetable to derive a feasible similar one (although in a different context), therefore fitting to our aim. LH relies on a *profit structure* for trains used to define the objective function: the idea is that each change that modifies the ideal timetable is penalized but changes on different train services have different impacts on the timetable quality. Each train is assigned a profit, and each change reduces this profit by a different value, giving rise to the profit structure. Since LH requires rather short computing times, we adopt a multi-start method, i.e., we heuristically construct several different profit structures, based on the relative importance of the trains and on the changes that we allow to make a timetable feasible.

For each profit structure, we execute LH and find a feasible timetable. We evaluate these timetables with the evaluation function (3.2). The best feasible timetable(s) found are then compared to the ideal timetable found in the previous step: We check for which OD-pairs the evaluation contribution gets worse. This serves as input in a feedback mechanism to generate new updated profit structures which are used for generating new feasible timetables. Details can be found in Section 4.4.

In the remainder of this section, we explain each phase in detail.

4.1. Make an ideal timetable

We define an *ideal* timetable as a timetable that fulfils all constraints on driving time and dwell time and minimizes total perceived travel time. In many cases, an ideal timetable will not be feasible with respect to infrastructure constraints.

To find an ideal timetable, we apply an approach that has been proposed in Polinder et al. (2021) in the context of *Strategic timetabling*: We formulate the problem for finding an ideal timetable as a mixed integer (quadratic) program and linearize it. On the linearized model, we apply a relax-and-fix heuristic (see Belvaux et al., 1998; Wolsey, 1998). to find a sequence of timetables, which fulfil the timetabling constraints with respect to driving time and dwell time, of increasing quality. For the sake of completeness, the MIP model from Polinder et al. (2021) is given in Appendix A.4. The linearization and details of the heuristic can be found in Polinder et al. (2021). The basic idea of the relax-and-fix heuristic is as follows: First, we restrict the set of OD-pairs

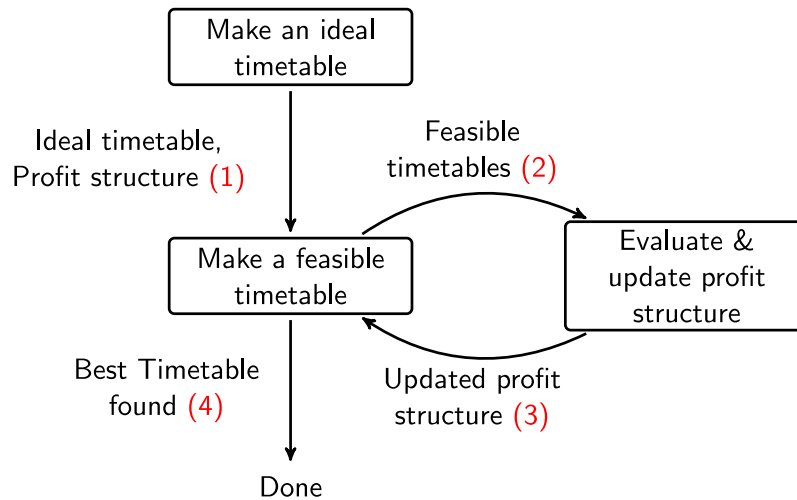


Fig. 2. Flow diagram of our approach.

to consider only the OD-pairs which are largest in size. In a first step, all variables are relaxed to continuous variables, except for the variables related to the timetabling constraints. This leads to a relatively easy model which we solve with a commercial solver until an optimal solution is found or a time limit is reached. Based on the timetable found, we use a dynamic programming approach to route passengers on routes with shortest perceived travel time, which, together with the timetable, provides a feasible solution for the MIP. In each further iteration step, more variables are changed into integers. The solution of the previous step is used as a warm start for the new step, which helps the solver in finding a (close-to) optimal solution for this step, which is then extended to a full solution by dynamic programming. After the heuristic finishes (when a time limit or optimality gap is reached), the resulting timetable is used as a warm start to solve the full ‘ideal timetabling model’.

4.2. Profit structure

LH relies on a profit structure that accounts for three operations applied to modify the ideal timetable and make it feasible: (i) *shift*, i.e., moving the departure time of some trains at their origin stations to an earlier or later point in time (and consequently moving the arrival and departure times, at all the stations visited by the train, by the same amount), (ii) *stretch*, i.e., increasing the train dwell time at some of the visited stations, and (iii) cancelling trains.

Each of these changes is undesirable from the perspective of an ideal timetable, and is thus penalized. Not all changes have the same impact: train cancellation has a stronger effect on passenger travel times but shift and stretch also affect the passenger travel times as they influence the adaption, in-train and transfer times. In addition, the severity of the consequences on passenger travel time will also depend on the type of train where a change is applied. To account for the impact of these changes on different trains, we define the following *profit structure*. We associate to each train $j \in \mathcal{P}$:

- a *train profit*: this value reflects the importance of scheduling the train, based on the train type (intercity, local, etc.) and on the line frequency (see Section 5.2.2 for details);
- a *shift penalty*: it decreases the profit for every minute of shift;
- a *stretch penalty*: it decreases the profit for every minute of stretch;
- a *maximum shift value*: it limits the (early or late) shift from the origin station defining a departure time window centred in the ideal departure time of train j ;
- a *maximum stretch value*: it limits the total stretch along the whole train path.

As detailed in Section 5.2.2, in our experiment, we use a multistart method, that is, we consider different values for shift and stretch penalties, as well as for maximum shift and stretch, since different profit structure produce different timetables.

4.3. Make a feasible timetable

To make a feasible timetable that satisfies all the headway constraints, we extend the method developed in Cacchiani et al. (2010).

Following that approach, we employ a time-index ILP model based on a time-space multi-graph, as the one reported in Appendix A.2. The goal is to find a feasible timetable that is as close to the ideal one as possible. The ILP model uses binary variables that represent whether a specific arc is chosen, with constraints ensuring that, for each train, at most one path (i.e., a timetable feasible for driving and dwelling times) in this time-space graph is selected, and that no two selected paths of different trains are in conflict to each other with respect to infrastructure. These constraints ensure that a timetable found with this method is feasible (Caprara et al., 2002).

In the ILP model, *profits* are assigned to the arcs, according to the profit structure defined in Section 4.2. These profits are highest if no change is applied, and decreased if the dwell time is *stretched* or the departure time from the origin station *shifted*, and the profit becomes null for a train if it is cancelled. In this way, the ideal timetable yields the highest possible profit, if it is feasible with respect to infrastructure, and changes are penalized. The time-index ILP model is reported in Appendix A.5.

As in Cacchiani et al. (2010), to have an efficient solution approach, we do not use a general-purpose ILP solver to solve the time-index formulation, but apply a Lagrangian-based heuristic algorithm. It consists of iteratively executing three steps: (i) apply the Lagrangian relaxation of the constraints that prevent infrastructure conflicts by using the current Lagrangian multipliers (initially set to 0), (ii) solve the relaxed problem by dynamic programming and update the Lagrangian multipliers, and (iii) compute a heuristic feasible solution.

At every iteration, the Lagrangian multipliers are updated through subgradient optimization by considering constraint violations or looseness (the relaxed problem is reported in Appendix A.6). As a consequence, arc profits are updated, and each train is associated with a Lagrangian profit (see Appendix A.6) that, in addition to the profit structure of Section 4.2, takes into account the penalties due to the relaxed constraints.

To compute a heuristic solution, at every iteration, the following steps are applied:

1. order trains based on their Lagrangian profits;
2. schedule one train at a time, in this order, in the most profitable and compatible way (avoid all conflicts with the previously scheduled trains) by dynamic programming;
3. apply a local search procedure to find a better path for trains that underwent shift, stretch or were cancelled.

We extend the approach in [Cacchiani et al. \(2010\)](#) to deal with periodic timetabling, train service regularity, and (simplified) rolling stock constraints, as described in the following.

To respect the periodicity requirement, if a train departs by the end of the time horizon and arrives later than T , then its arrival is mapped back to the beginning of the time horizon: i.e., the arrival node in the time-space graph is mapped to the corresponding node at the beginning of the time horizon. The same holds for train stops that traverse the end of the time horizon: in this case, the departure node after the stop is mapped to the corresponding node at the beginning of the time horizon. As explained above, we impose a limit on the time change that can be applied to each train departure or arrival: in particular, the departure of a train from its origin station is bounded (early or late) by the maximum shift value, while the total stretch along the whole train path is bounded by the maximum stretch value. Accordingly, the maximum change that train departure (or arrival) time can get is given by the sum of earlier shift, later shift and total stretch. As the timetable constructed in this paper has to be periodic, we impose this maximum change to be at most T : indeed, changing a train departure (or arrival) time by $T + K$ time units is equivalent to changing it by K units.

In contrast to the approach described in [Cacchiani et al. \(2010\)](#) that only penalizes shift (deviation from ideal timetable) at the first station of a train, our approach can also penalize shift at intermediate stations. In this way, we can penalize deviation from a regular pattern at intermediate stations. This is particularly relevant for improving the feasible timetable in the feedback mechanism, since irregular departures lead to significant worsening of the objective to minimize the total perceived travel time.

With respect to the method described in [Cacchiani et al. \(2010\)](#), our approach is able to take into account basic rolling stock constraints: in practice, trains of the same line (i.e., trains with the same origin and destination stations, and stopping at the same intermediate stations) are scheduled in “pairs”. In this way, when a train is scheduled in one direction, another train is also scheduled in the opposite direction, and the same rolling stock (physical train) is assigned to both services. This also leads to a more regular timetable having the same number of trains running in both directions. Since LH allows train cancellation, we need to guarantee that, if a train j_1 is cancelled, then also train j_2 , going on the same line in opposite direction, is cancelled. Clearly, this can be easily obtained by simply cancelling j_2 , but it is highly undesirable. Therefore, we modify LH by including a new procedure that first cancels trains to balance both directions of each line, and then tries to reschedule them. At each iteration, after a feasible timetable has been determined, for every train line, if needed we cancel additional trains to have the same number of trains scheduled in both directions. Once all train lines have been processed, we try to reschedule trains in pairs, for each train line: we compute, for each train, the most profitable path compatible with the previously scheduled ones. If there exists a feasible path for each direction of a line, the corresponding trains are inserted, otherwise they are cancelled again. Since this procedure can change the set of scheduled trains, at the end we apply again the local search procedure. Note that it is possible to schedule previously cancelled trains thanks to the additional train cancellation applied at the beginning of this procedure. We also observe that, for the considered instances, the number of trains per line is two or four, and thus this procedure can be executed efficiently.

4.4. Evaluate & update profit structure

The profit structures used in LH are heuristically chosen to penalize deviation from the ideal timetable, based on the severity of the deviation and the relative importance of trains. Different profit structures lead to a different order in which train timetables/paths are fixed in the steps of LH, which often leads to different timetables. For this reason, we do not use one, but several profit structures, which typically lead to different feasible timetables. We evaluate each timetable generated by LH using the evaluation function (3.2). The best timetable according to this evaluation function after running LH will be referred to as best pure Lagrangian (BPL) timetable in the remainder of this manuscript. However, there is no guarantee that the BPL timetable is the best feasible timetable with respect to our evaluation function. Therefore, as an attempt to find even better timetables, we update the profit structures based on a comparison of BPL timetable and ideal timetable, restart the LH with the updated profit structures, and find new feasible timetables.

By comparing the evaluation contributions of all OD-pairs in the ideal timetable π and in the BPL-timetable π' , we can identify the OD-pairs for which the evaluation contribution increased the most. We inspect the routes chosen for these OD-pairs and the corresponding trains to find the reason of the increase. In a ‘feedback’ step, we generate a new set Ψ of updated profit structures, based on the initially chosen profit structure S to penalize the undesired changes more. Since we are not able to predict an ‘optimal’ penalty for deviations, we use a set of penalty values $P = \{p_1, p_2, \dots, p_m\}$ and create several profit structures based on S and P . We proceed as follows:

1. Identify the OD-pairs with highest *excess evaluation contribution*. The *excess evaluation contribution* is computed as follows: Based on minimum drive, dwell, and transfer times and penalties, we compute lower bounds on perceived route lengths. Furthermore, we can compute a lower bound on the average waiting time of passengers of OD-pair k by assuming that the relevant departure events for this OD-pair are perfectly synchronized, and that each passenger chooses the route that departs earliest after his desired departure time. In this way, we can lower-bound the average waiting time for a passenger of OD-pair k by $\frac{T}{2 \cdot |V^k|}$. The sum of the lower bound for perceived route length plus the lower bound on adaption time, multiplied with the number of passenger for this OD-pair gives us a lower bound on the OD-pair’s evaluation contribution. The difference between the evaluation contribution of an OD-pair and the so-computed lower bound is the *excess evaluation contribution*. This increase with respect to the lower bound can be caused by a large number of passengers in the OD pair or by a high increase in the perceived travel time. We include the OD-pairs whose excess evaluation contribution exceeds a certain threshold in the set of *relevant* OD-pairs. To avoid that we increase penalties for many OD-pairs, we furthermore impose an upper bound B on the number of relevant OD-pairs. If there are more than B OD-pairs exceeding the threshold, we include the B OD-pairs with highest evaluation value contribution in the set of relevant OD-pairs. Let $\mathcal{O} = \{o_1, o_2, \dots, o_k\}$ be the set of origin-stations for the relevant OD-pairs.
2. We create $|\mathcal{O}| \cdot |P|$ new profit structures, one for each penalty value $p_j \in P$ assigned to each station $o_i \in \mathcal{O}$. More precisely, we create the corresponding profit structure as follows: for each station $o_i \in \mathcal{O}$ and for each $p_j \in P$, we create a new profit structure that is based on S , with an additional intermediate shift penalty of value p_j that is assigned to all trains passing station o_i . Furthermore, all trains for which o_i is an origin or terminal station and which are *relevant* for the corresponding OD-pair identified in Step 1, receive shift penalty $\max\{s(t), p_j\}$,

where $s(t)$ is the regular shift penalty for train t in profit structure S .

If $|\mathcal{O}| > 1$, we generate additional profit structures. In these new profit structures we apply the same principle as above, but now we apply the penalties to all *pairs* of stations. In particular, for each pair of stations $o_i, o_h \in \mathcal{O}$ and for each pair of penalty values $p_j, p_l \in P$, we assign (intermediate) shift penalty p_j to station o_i and p_l to station o_h .

For example, if $|\mathcal{O}| = 2$ and $|P| = 2$, we create 4 profit structures, as described above, that correspond to assigning p_1 to station o_1 or o_2 , and similarly p_2 to station o_1 or o_2 . Moreover, we generate 4 additional profit structures: two are obtained by assigning p_1 both to stations o_1 and o_2 , or p_2 both to stations o_1 and o_2 , while two other profit structures are obtained by assigning p_1 to station o_1 and p_2 to station o_2 , or p_2 to station o_1 and p_1 to station o_2 .

Given the set of updated profit structures Ψ , we again run LH. Each of these profit structures leads to a new timetable which we evaluate. If any of these timetables gives a better evaluation value, we stop the feedback process and finish with the best timetable generated using Ψ . Else, execute steps 1 and 2 again with the original profit structure as input, but now also identify OD-pairs as relevant for which the evaluation contribution increased the most in the best timetable generated using the profit structures in Ψ . The timetable that is the best after providing feedback is referred to as the *best after feedback* (BF). The feedback process is repeated until either we find an improved timetable or we reach a given maximum number of iterations.

We underline that the intermediate shift penalty is not adopted at every intermediate station where there are irregular departure headway times, but only at those that cause a significant increase in the evaluation value. Indeed, it would not be effective to penalize shifts at every station, since some changes are needed in order to get a feasible timetable. Therefore, we aim at penalizing the changes that have most impact on the evaluation value of the timetable. For the same reason, we do not use the intermediate shift penalty when LH is applied to the ideal timetable in the first round before the feedback process.

The rationale behind our feedback approach is that adaption times have a strong influence on the evaluation value of a timetable. If LH causes a higher irregularity in the new timetable compared to the ideal timetable, the evaluation value is likely to increase. For this reason, we focus on the shift penalties in the feedback process. However, many alternative strategies to provide feedback in order to reduce the evaluation contributions of OD-pairs can be thought of, as we can update initial train profits, shift and stretch penalties, maximum shift and maximum stretch, as well as adding intermediate shift penalties. We experimented with different strategies on how to update profit structures before we identified this one which was successful on our three test instances and that is presented here.

5. Case study

In this section we perform three case studies of the Dutch railway network with increasing complexity in terms of considered number of trains and OD-pairs. First, we describe the three instances in Section 5.1. Next, in Section 5.2 we describe the parameters that are used in our approach. In Section 5.3, we describe and discuss the obtained results. Finally, in Section 5.4 we benchmark our iterative approach with solving the POT model as an integer program (see the extended model in Appendix A.4).

5.1. Instances

Here we describe the instances that we consider in more detail. Each instance consists of a railway network displayed in Fig. 3: as can be seen, Fig. 3(a) contains a central corridor with few connected lines, while Fig. 3(b) and Fig. 3(c) are larger networks. Some of the stations

in Fig. 3 are labelled, because we refer to them in the presentation and discussion of the results.

5.1.1. A2 instance

The first instance that we consider is the ‘A2’ that consists of a main corridor between the stations Eindhoven (Ehv) and Amsterdam Central (Asd), and few additional lines from Hdr, Ut and Ehv. The network contains 34 stations. We consider five Intercity train lines that are operated on this network with a frequency of two trains per hour in both directions, so there are 20 trains in total. In this instance, we consider 891 OD-pairs.

The underlying event-activity network contains 1344 events and 1460 drive and dwell activities. To build the model for constructing the ideal timetable, 376 transfer activities (which do not impose operational constraints on the timetable, but facilitate passenger routing) are included. An inclusion of infrastructure constraints in the PESP framework leads to 2964 additional activities.

5.1.2. Rotterdam–Groningen instance

The second instance covers part of the 2019 line plan of Netherlands Railways (NS, 2019). It is centred on the line between Rotterdam (Rtd, in the South-West) and Groningen (Gn, in the North-East), and also includes all connected lines that share a part of their route with this main line. Fig. 3(b) shows the corresponding network. The network contains 77 stations. We consider 3810 OD-pairs and 60 trains in the network.

The underlying event-activity network contains 1664 events and 1716 drive and dwell activities. 1402 transfer activities are added to enable passenger routing. To model infrastructure constraints, 4004 additional activities are needed.

Note that, although in this instance we have three times as many trains compared to the A2 instance, the number of events, as well as of drive and dwell activities, increases only slightly. The reason for this is that we have a number of trains on short lines (and thus less events are needed per train line). The main increase is seen in the number of transfer activities, to generate all possible transfer routes, as well as in the number of activities modelling infrastructure constraints.

5.1.3. Extended A2

The third instance is an extension of the A2 instance. It contains all train lines in the 2019 network of Netherlands Railways that share a part of their route with the main line between Amsterdam Central (Asd) and Eindhoven (Ehv). As can be seen from Fig. 3(c), this is a large network with many interconnected lines. The network contains 140 stations. We consider 88 trains on the network and 11121 OD-pairs.

The event-activity network contains 3160 events and 3308 drive and dwell activities. 3592 transfer activities are added to enable passenger routing. Adding the infrastructure constraints leads to 8360 additional activities.

5.2. Parameters

To solve the mixed integer linear programs in the computation of ideal timetables, and to solve the benchmark model (see Section 5.4), we use a machine with an Intel Xeon Silver 4110 2.10 GHz processor with 96 GB of RAM installed. The mathematical programs are solved by Cplex 12.9.0 under default settings, using up to 15 parallel threads (IBM, 2019). We use a time limit of two hours for the ‘A2’ instance, and a time limit of four hours for the ‘Rotterdam–Groningen’ instance and the ‘Extended A2’ instance. In the heuristic approach to solve the MIP models, we use a subset of the OD-pairs: we include as few OD-pairs as possible, with the restriction that at least 30% of all passengers are included in this smaller model.

In all our experiments we discretize time to minutes and use a period length of one hour, i.e., $T = 60$.

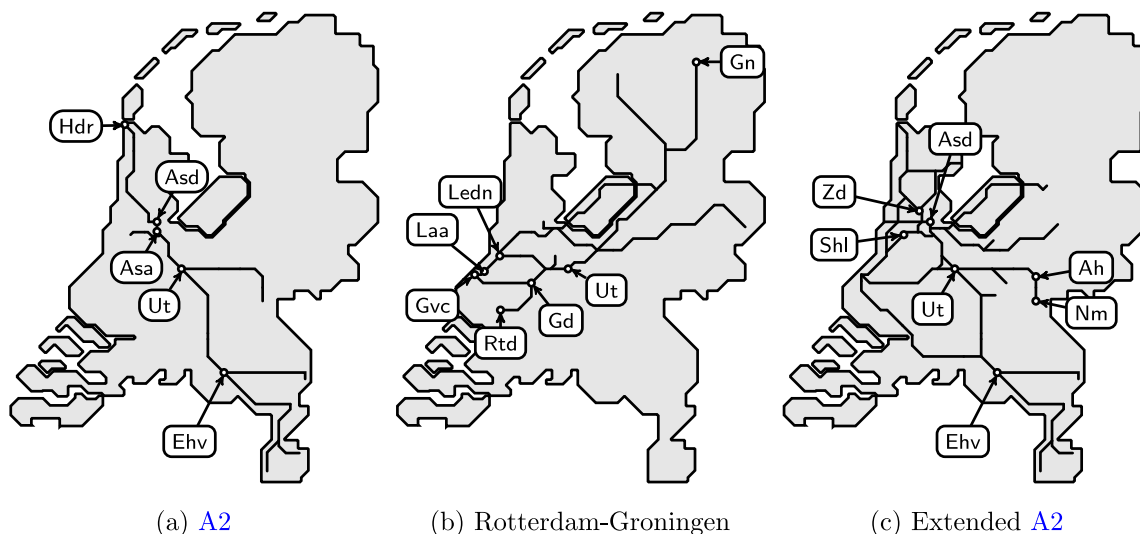


Fig. 3. Networks of the three instances considered.

In the remainder of this section, we specify the parameters that we use in our experiments. First, we describe the parameters for the objective function. Next, we detail the profit structure that we use for LH.

5.2.1. Objective function parameters

In line with Polinder et al. (2021) and de Keizer et al. (2015), we set $\gamma_t = 20$, i.e., the transfer penalty equals 20 min. We use $\gamma_s = 1$, i.e., transfer time weights as much as in-train time as we already have a transfer penalty. We set $\gamma_w = 3$, i.e., adaption time weights three times as much as in-train time.

If passengers no longer have a travel option when trains are cancelled, we count a penalty of value $M = 24 \cdot T$ (a full day) as the perceived travel time of these passengers.

Note that in the presentation of our results, for confidentiality reasons we normalize evaluation values with the ideal timetable. That is, an increase in the evaluation value by one unit means the evaluation value is 1% higher than that of the ideal timetable.

5.2.2. Profit structure and feedback parameters

The Lagrangian heuristic LH requires the specification of a profit structure (see Section 4.2). In our experiments, we initially create 9 profit structures for the multistart approach. These profit structures are based on the same initial *train profits*, but differ in the stretch and shift penalties and the in the maximum values for stretch and shift.

Train profits are based on the train type (Intercity, local, etc.) and on line frequencies. For the train type ‘Intercity’, we consider a base profit of 4000. For the train type ‘local train’, the base profit is reduced by 10% to 3600 and for trains that partly operate as an Intercity and partly as a local train, it is reduced by 5% (to 3800). In this way, Intercity trains have priority in the scheduling process over the other train types, hence are more likely scheduled as in the ideal timetable. To account for different line frequencies, for each train we identify for each pair of consecutive stations on the corresponding line the number of trains that travel between the stations, and take the minimum m of these numbers along the line. The train profit is computed as the base profit divided by m . For example, if we consider a local train line with frequency two that is the only train line offering a service on some part of the network, we have $m = 2$ and the profit for the trains in this line is $3600/2 = 1800$. If another line with frequency two is present as well on the considered part of the network, we have $m = 4$ and the profit is 900. Thus, trains with lower frequency become more important in the scheduling process than more frequent trains, so that it is more probable that the former are not cancelled.

For the shift and stretch penalties, we consider equal values for all trains, and use three different settings: (1) shift penalty set to 20 and stretch penalty to 10; (2) both shift and stretch penalty set to 15; (3) shift penalty set to 10 and stretch penalty to 20. Namely, we assign more importance to the shift penalty in the first case, same importance to both changes in the second case, and more importance to the stretch penalty in the third case. Indeed, it is not known a priori whether a shift or a stretch is worse: it depends on the location where this happens and on the influence it has on the regularity of trains in general. In addition, we want to explore a rather broad spectrum of profit structures, because it is not a priori known which changes in the timetable have the least negative effect on the evaluation of the timetable according to evaluation function (3.2).

For maximum shift and maximum stretch, we consider three settings: (1) maximum stretch and maximum shift both equal five: this means that each train can have its departure time from its origin station up to 5 minutes earlier or 5 minutes later, and a total stretch along its route of up to 5 minutes, (2) maximum shift is 10 and maximum stretch is 5, (2) maximum shift is 5 and maximum stretch is 10.

Based on preliminary experiments, the total number of iterations for each run of LH is set to 250.

In the *evaluate and update* step we use a threshold of +0.03 to identify the set of relevant OD-pairs, together with a bound $B = 4$ on the size of this set. Indeed, by modifying the profit structure of a small number of trains, we are able to focus on the locations that show higher irregularity of service. We adapt the values for the intermediate shift penalties at the corresponding stations, as well as for the initial shift penalty of a train starting in such a station. We use values of 10, 20 and 30 for these penalties. We set the maximum number of iterations for the feedback mechanism to 5: this limit was never reached in our experiments, since we found an improved timetable for all instances within 2 feedback iterations.

5.3. Results of the algorithm

In this section we illustrate the different steps of our approach on the three described instances, providing insights on critical points in the network and trade-offs in timetable construction.

5.3.1. A2 instance

Make an ideal timetable. Within the time limit of two hours, we are not able to solve the optimization problem of finding an ideal timetable to optimality. The lower bound that is proven by CPLEX is 97% of the objective value, hence the remaining gap is 3%.

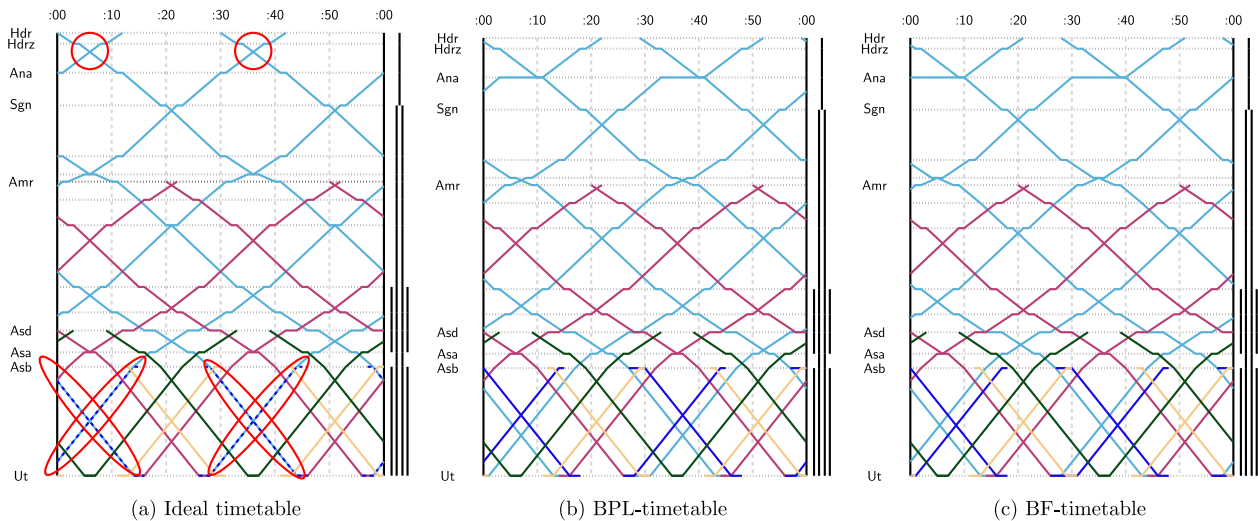


Fig. 4. Time-space diagrams for A2.

To illustrate the best ‘ideal’ timetable π found within the time limit of two hours, we show a time-space diagram for the corridor between Hdr and Ut in Fig. 4(a). Time is shown on the horizontal axis, between 0 and 60, i.e., one cycle period is displayed. Space is shown on the vertical axis, where several stations are mentioned. The lines on the right of the figures display, for each section, the number of tracks that are present on that section.

Even though no regularity constraints are added to the model, we see in Fig. 4(a) that the trains are spread over time rather regularly in this network thanks to the inclusion of the adaption time in the objective function. However, the ideal timetable does not satisfy the constraints imposed by the infrastructure. There are two conflicts, which are indicated by red circles in Fig. 4(a). At one location, two trains are scheduled at the exact same time: the light blue and dark blue trains are scheduled at the exact same time between Ut and Asb. At another location two trains are scheduled to cross each other in a single track area between Hdr and Sgn.

Make a feasible timetable. We run LH with the nine different parameter sets specified in Section 5.2.2, thus obtaining nine feasible timetables in which all trains are scheduled. The best timetable found in this step, *BPL*, has an evaluation value of 100.18, i.e., the evaluation value increased by 0.18% with respect to the ideal timetable. For the *BPL* timetable, the time-space diagram is shown in Fig. 4(b), where it can clearly be seen that the conflicts are resolved. Trains that crossed on a single track area are now stretched such that they pass each other at a station. Secondly, the trains that were scheduled at the same time are now moved away from each other. Fig. 4(c) shows the best timetable after feedback, reported here for a more direct comparison with the ideal and *BPL* timetables. The process to obtain it is discussed in the following.

Evaluate & update profit structure. Fig. 5(a) shows the difference in excess evaluation contribution for individual OD-pairs between timetables π and *BPL*. Each OD-pair is represented by a star, the stars are sorted from left to right according to the excess evaluation contribution of the corresponding OD-pair in the ideal timetable. As can be seen, for many OD-pairs the excess evaluation contribution is small in the ideal timetable, for only a few OD-pairs it is large. These OD-pairs are selected for profit structure updating according to the considered threshold $+0.03$ and bound $B = 4$. In this instance, these OD-pairs correspond to those having a high demand and a small irregularity in their departure pattern: recall that the evaluation contribution is weighted by the number of passengers, i.e., a large increase for only

a few passengers can count less than a small increase for many passengers. In Fig. 5(a), the vertical coordinate indicates the difference in evaluation value between π and *BPL*. In particular, if a star lies above 0, the evaluation contribution of the OD-pair has increased after applying LH. But there are also some OD-pairs which have a lower evaluation contribution after applying LH, these can be found below 0. Two OD-pairs are labelled in the figure: these are the OD-pairs with the increase in evaluation contribution over the considered threshold, and on which we base the feedback. The OD-pair Ut-Asd has an excess evaluation contribution of 0.38 in the ideal timetable, and that evaluation contribution increased to 0.47 in the *BPL*-timetable, due to a more irregular departure pattern at Ut, giving an excess evaluation contribution of 0.09. Similarly, the OD-pair Ut-Asa gives an excess evaluation contribution of 0.07.

Fig. 5(b) contains a different visualization of the differences between π and *BPL* with respect to the evaluation contribution. All 891 OD-pairs are shown on the horizontal axis sorted by their corresponding increase in evaluation contribution. As can be seen well in this figure, there are many OD-pairs for which the evaluation contribution hardly changed, only for a minority there are major changes. Hence, also in this figure it can be seen that the *BPL* timetable is very close to the ideal one, and only few OD-pairs were subject to an increase in evaluation contribution.

First feedback step. As mentioned above, two OD-pairs are identified as relevant, based on the considered threshold: Ut-Asd and Ut-Asa (see Fig. 5).

These two OD-pairs have the same set of travel options, as the trains from Ut to Asd first pass Asa (see Fig. 3(a)). In fact, Asa is the only station where the trains from Ut to Asd stop. Passengers on these OD-pairs can choose from six different trains. Thus, in a timetable that minimizes adaption time upon departure in Ut, the headway times between consecutive trains would be 10 min. In the ideal timetable we have headway times of 12 min (2 times) and 9 min (4 times), as is visible in Fig. 4(a). As these OD-pairs have high passenger numbers, the deviation from the ideal timetable causes high excess evaluation contributions. In the *BPL*-timetable, the departure pattern in Ut becomes even less regular. The headway times now are 15 min (twice), 9 min (twice) and 6 min (twice), i.e., excess evaluation contributions are even higher.

We add intermediate shift penalties, according to the values defined in Section 5.2.2, at Ut to improve the timetable for Ut-Asd and Ut-Asa, and run LH with the three new profit structures. Unfortunately, none of these timetables gives an overall improvement. For the OD-pairs in the other direction (Asd-Ut and Asa-Ut) the situation becomes worse

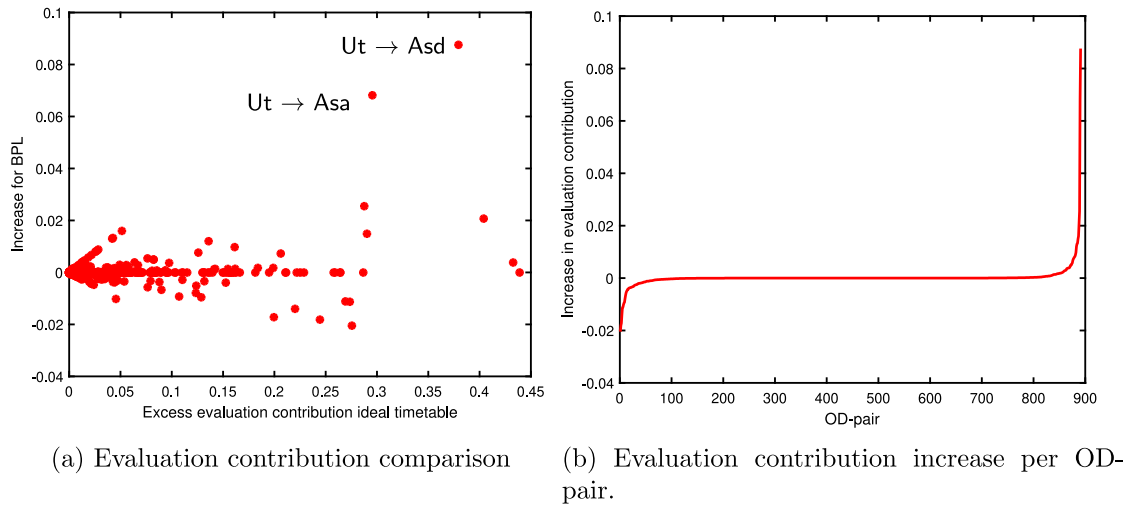


Fig. 5. A2: ideal timetable vs BPL-timetable.

after the feedback because the regularity at Asd (and in line with that also at Asa) is lost. Indeed, the excess evaluation contribution of Asd-Ut becomes 0.093 and that of Asa-Ut becomes 0.067. Hence, we identify these two OD-pairs as the new set of relevant OD-pairs, and apply a second feedback step.

Second feedback step. Starting with the initial profit structure that was used to find the BPL timetable, we add shift penalty values at Ut and Asd, which have been identified as relevant OD-pairs for feedback. This leads to 12 new profit structures: 3 for adding the three different penalty values 10, 20, and 30 at Asd and 9 for the combinations of Ut and Asd. Note that a shift penalty at Asd can also improve the timetable at Ut, since these two stations are close to each other and there is only one stop in between.

The best evaluation value among the 12 created timetables is 100.10, which is an improvement with respect to the BPL timetable. The time–space diagram for this timetable is shown in Fig. 4(c), where we can see the improved regularity between Ut and Asd.

By preliminary experiments, we observed that additional feedback loops do not achieve further improvement. Indeed, the objective value found after this improvement is very close to the ideal one (100.1 versus 100), and to obtain a better timetable we would rather need to restart with a completely new profit structure. Therefore, we stop the feedback loop after the first improvement, as explained in Section 4.4. We refer to the timetable found in the second feedback iteration as the *Best after feedback* (BF).

To visualize the changes in the timetable, we show in Fig. 6 the same plots as in Fig. 5 extended by adding blue stars, showing the evaluation contributions in the BF-timetable. Note that the OD-pairs Ut-Asa and Ut-Asd significantly improve in the second iteration, as the blue stars representing these OD-pairs are now on the horizontal axis (the indicated arrows show the obtained improvements). However, there are also some OD-pairs for which the evaluation contribution increases in comparison to the BPL timetable. This is best seen in Fig. 6(b). Here, the same red line as in Fig. 5(b) is shown. The excess evaluation contributions for the OD-pairs in the timetable after feedback are added in blue. It is visible that some OD-pairs for which the evaluation contribution was the same as in the BPL timetable are now changed: for some the evaluation contribution decreases, but for others it increases. It is also visible that the excess evaluation contributions are now smaller than in the BPL timetable, thus leading to an overall improved timetable.

Summary. A summary of the progress of our approach on the A2 instance is given in Table 1. The table displays the evaluation values for the best timetables found in the steps of the algorithm. The last row of the table shows the lower bound that is found by CPLEX when solving

Table 1
Evaluation values for A2.

Timetable	Evaluation value
Ideal	100
Best Pure Lagrangian	100.18
Feedback step 1 (FB-1)	100.23
Feedback step 2 (FB-2)	100.10
Lower bound	97.00

the integer programming model for finding an ideal timetable, which is, of course, also a lower bound on the objective value of a feasible timetable. We can see that the BF timetable, that respects all headway constraints, is very close to the ideal one.

A visual summary of our approach is displayed in Fig. 7. Here, the evaluation values of all computed timetables are displayed. The horizontal axis displays the step in the algorithm. The vertical axis shows the evaluation value of the timetable. First, the evaluation value of the ideal timetable is shown at the bottom left. Then, the blue lines and dots link this evaluation value to the evaluation values of the nine timetables computed by LH. Next, the red lines and dots display the values of the timetables computed during the feedback process. The evaluation values of the three timetables computed in the first and second feedback step (FB-1 and FB-2) are shown, with lines linking this evaluation value to the evaluation value of the BPL timetable. It is clearly visible that the evaluation value of the BF timetable is lower than that of the BPL timetable.

5.3.2. Rotterdam–Groningen instance

Make an ideal timetable. Within the time limit of four hours, the optimality gap of the best found ‘ideal’ timetable to the lower bound is 7.31%.

Figs. 8(a) and 8(b) show the time–space diagrams for best found ideal timetable π on two relevant corridors: Rotterdam (Rtd) to Utrecht (Ut) and Leiden Central (Ledn) to The Hague Central (Gvc).

The ideal timetable violates infrastructure constraints. E.g., between stations Gd and Wd, two trains are scheduled at the exact same time on the same track and hence do not satisfy the headway constraints (see the circled area in Fig. 8(a)). Also on the other corridor violations of the headway constraints occur: e.g., two trains from the same direction arrive in Gvc at the same time, while there is only one track available for them, so the headway constraint upon arrival of two trains is not satisfied (see the circled area in Fig. 8(b)).

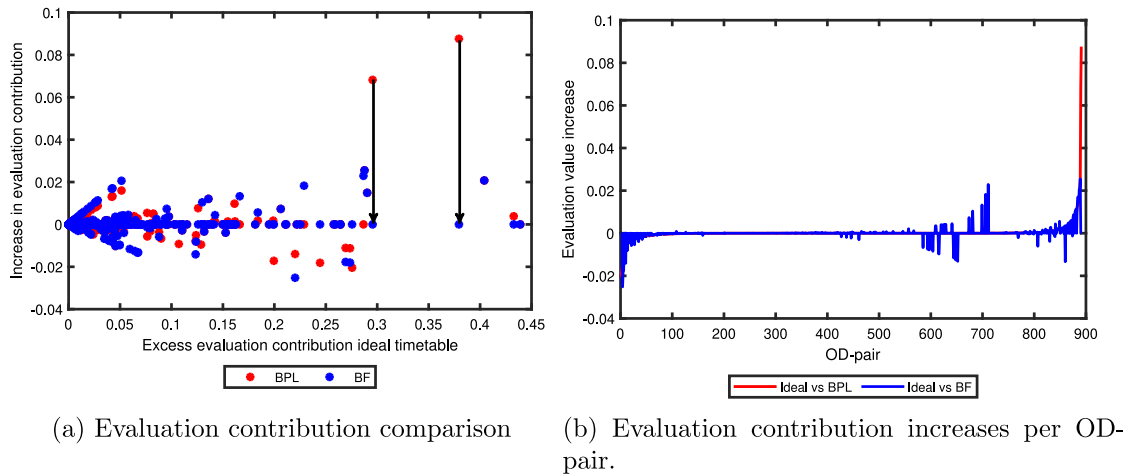


Fig. 6. A2: Timetable comparisons after feedback.

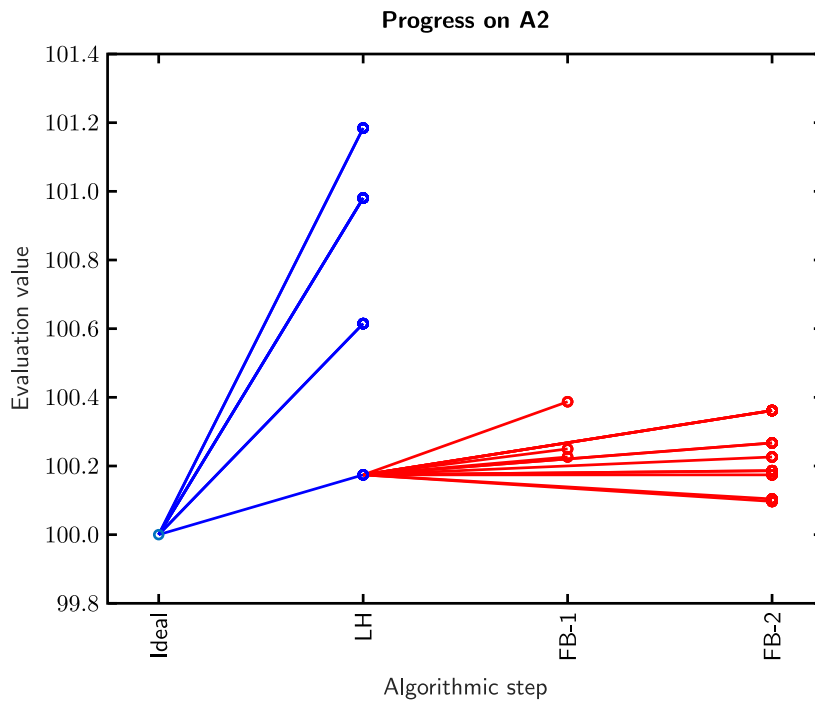


Fig. 7. Overview of the progress in A2.

Make a feasible timetable. In seven out of the nine timetables obtained with the different parameter settings (Section 5.2.2) of the multistart approach, all trains are scheduled. In two of them, two trains are cancelled (however, also in these timetables there are still travel options for all passengers). The best evaluated timetable has an evaluation value of 100.59 with all train being scheduled. The time–space diagrams for this BPL-timetable on the two aforementioned corridors are displayed in Figs. 8(c) and 8(d). In Figs. 8(e) and 8(f), we show the corresponding best timetables obtained after feedback through the process described in the following.

Evaluate & update profit structure. To illustrate the differences between the ideal timetable π and the BPL-timetable, we show, in Figs. 9(a) and 9(b), the same plots as for the previous instance, showing the differences in evaluation contribution per OD-pair. Note that for many OD-pairs the evaluation contribution does not change.

In the BPL-timetable, there are three OD-pairs above the considered threshold, as is indicated in Fig. 9(a): Ledn-Laa and Ledn-Gvc are

two OD-pairs that have very similar routes, have the same origin and their destinations are close to each other (see also Fig. 3(b)). The corresponding excess evaluation contributions increased by +0.056 and +0.044 respectively with respect to the ideal timetable. The third OD-pair for which the evaluation value is above the threshold is Rtd-Ut: its excess evaluation contribution increased by +0.050 with respect to the ideal timetable. This OD-pair is located on a different part of the network than the other two OD-pairs.

First feedback step. Based on the parameters defined in Section 5.2.2, we make new profit structures, by adding the intermediate shift penalties of values 10, 20 and 30 for trains starting in or traversing Ledn and Rtd. This leads to 15 new profit structures: 3 with a penalty only at Ledn, 3 with a penalty only at Rtd, and 9 for all combinations of penalties at the stations.

We run LH on these new profit structures and evaluate the resulting timetables. We find the best timetable to have an evaluation value of 100.55. This timetable is referred to as the *best after feedback* (BF).

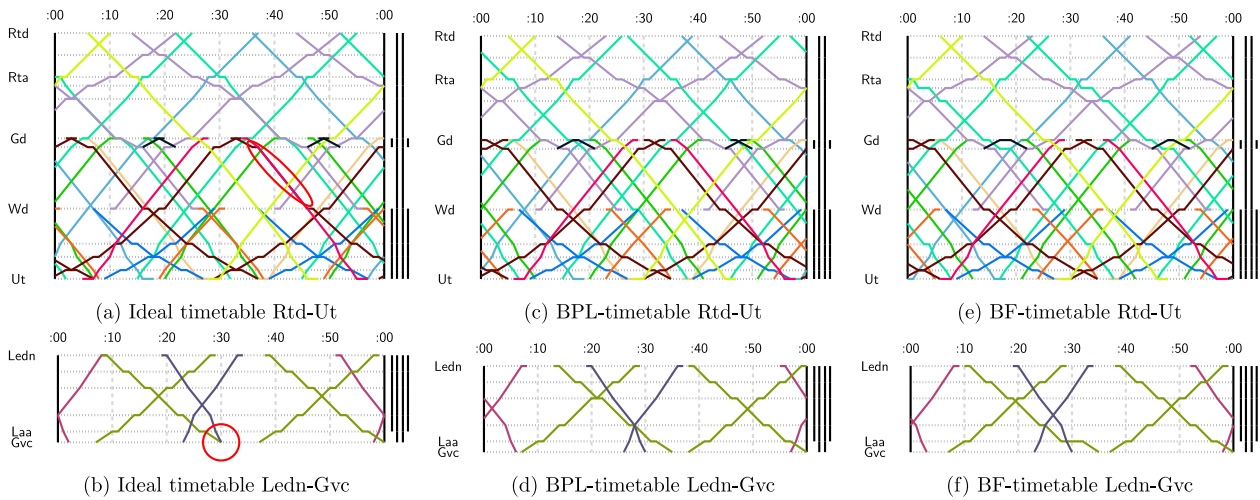


Fig. 8. Time-space diagrams for the Rotterdam-Groningen instance.

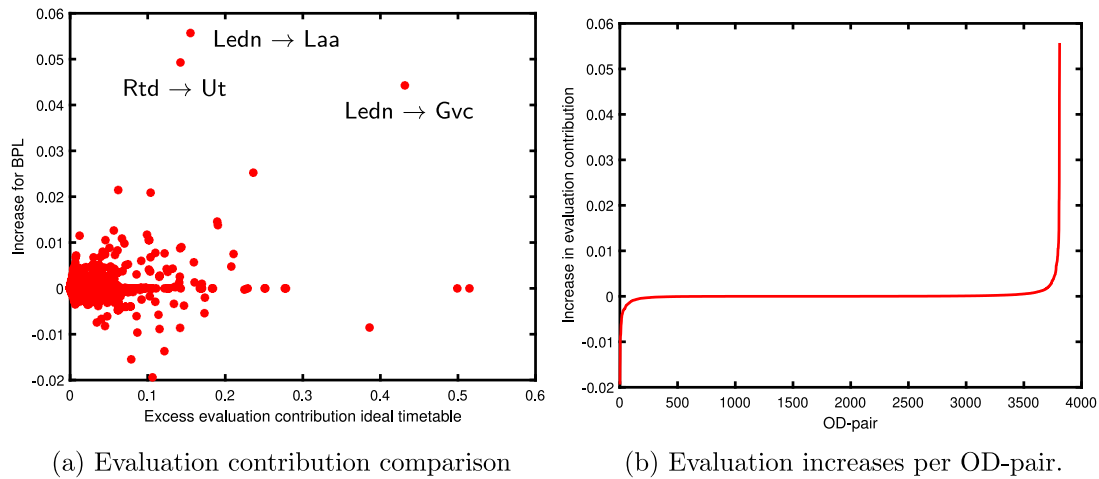


Fig. 9. Rotterdam-Groningen: Ideal timetable vs BPL-timetable.

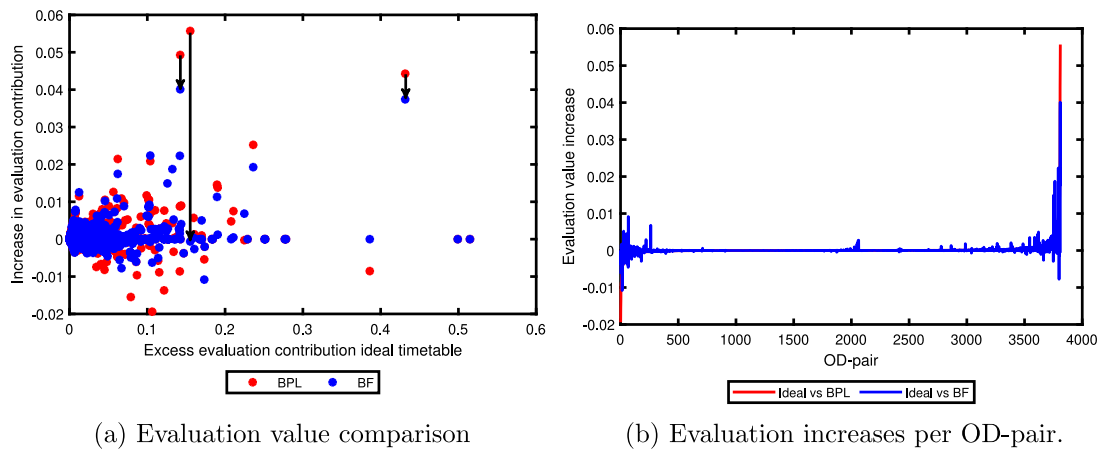


Fig. 10. Rotterdam-Groningen: Timetable comparisons after feedback.

We show in Fig. 10 the evaluation contributions of the OD-pairs in the BF-timetable to illustrate the differences with respect to the BPL-timetable. The increase in evaluation contribution with respect to the ideal timetable of Ledn-Laa reduces to -0.001 , i.e., the timetable for this OD-pair is better than the ideal timetable. The increase in evaluation

contribution for Ledn-Gvc now reduces to $+0.037$ and Rtd-Ut reduces to $+0.040$.

Similar to the previous instance, we stop after the first improvement step, as also in this case we obtain an objective value very close to the ideal one.

Table 2
Evaluation values for Rotterdam–Groningen instance.

Timetable	Evaluation value
Ideal	100
Best Pure Lagrangian (BPL)	100.59
Feedback step 1 (FB-1)	100.55
Lower bound	92.69

Table 3
Evaluation values for extended A2 instance.

Timetable	Evaluation value
Ideal	100
Best Pure Lagrangian (BPL)	101.51
Feedback step 1 (FB-1)	101.28
Lower bound	93.00

As an example of how the feedback process allows for obtaining improvements in the timetable, we refer to OD-pair Ledn-Laa, and examine the time–space diagrams in Figs. 8(b), 8(d) and 8(f). In the ideal timetable, passengers can either take a local train directly from Ledn to Laa, or an Intercity train that travels to Gvc, where passengers can transfer, with 5 min of transfer time, to a local train back to Laa.

In the BPL-timetable though, the Intercity train is shifted while the local train is not shifted: therefore, passengers have to wait half an hour for the connection, leading to a very high excess evaluation contribution.

In the feedback, trains get an intermediate shift penalty at Ledn. This causes the Intercity train still to arrive in Gvc later than in the ideal timetable, but earlier than in the BPL-timetable. Thus, passengers can again make the connection to the local train and we are in a similar situation as in the ideal timetable.

This illustrates how the additional penalties lead to a different timetable and how feedback can be used to improve the timetable that is found.

Summary. Table 2 and Fig. 11 summarize the progress of our approach on the Rotterdam–Groningen instance. The evaluation values are displayed for each step of our algorithm. We can observe that the obtained timetable is very close to the ideal one.

Note that two lines are drawn with a dash-dotted line. These correspond to the timetables where some trains are cancelled. We observe that the evaluation value of these timetables is not extremely high: indeed, all OD-pairs still have a travel option, hence, penalty M (defined in Section 5.2) does not have to be considered. We can also see that these timetables have a better value than two timetables in which all trains are scheduled (solid lines with evaluation value around 101.4): this happens because the cancelling of trains gives us more freedom to schedule other trains, that can thus have a timetable more similar to the ideal one. However, we can also clearly see that the best timetables do not have any cancelled train: indeed, train cancellation reduces the travel options for the passengers and usually increases their adaption time.

5.3.3. Extended A2

Make an ideal timetable. After the time limit of four hours, we find a solution with optimality gap 7.0%.

Time–space diagrams displaying the best found ideal timetable are shown in Figs. 12(a)–12(c) for three corridors in the network: Arnhem (Ah) to Nijmegen (Nm), Amsterdam Central (Asd) to Schiphol (Shl) and Zaandam (Zd) to Utrecht (Ut). See Fig. 3(c) for the location of these stations. In this dense network, there are numerous conflicts between trains that have to be resolved in the next step (hence we do not show red circles here).

Make a feasible timetable. Using the same initial profit structures as in the earlier cases, in the first step of LH nine feasible timetables are computed. In three of these timetables, all trains are scheduled. In the other six, trains are cancelled in a way that causes some passengers not to have a travel option any longer. This is highly penalized in the evaluation function and as a consequence these six timetables have a bad evaluation value. However, the evaluation value of the best feasible timetable found is 101.51, that is rather close to the ideal value. The corresponding time–space diagrams are shown in Figs. 12(d)–12(f). We

show in Figs. 12(g)–12(i) the best timetables obtained after feedback through the process described in the following.

Evaluate & update profit structure. The changes in the evaluation contribution for each OD-pair are pictured in Fig. 13, in a similar way as in the previous two cases. We observe that for many OD-pairs there are only few or little changes (see Fig. 13(b)). However, there are a few OD-pairs for which the evaluation contribution increases significantly as is visible in Fig. 13(a). The highest increase occurs for the OD-pair Ah-Nm. Its evaluation contribution increases by +0.062. Besides Ah-Nm, Asd-Ut (+0.045), Asd-Zd (+0.039) and Asd-Shl (+0.038) account for evaluation contributions above the threshold. Indeed, as illustrated in Fig. 12, we see that departure and arrival patterns have changed at Asd, trains depart in a less regular pattern. Also the pattern of departures at Ah is less regular in the BPL-timetable than in the ideal timetable (compare Figs. 12(a) and 12(d)).

First feedback step. The relevant OD-pairs with excess evaluation contribution above the threshold are: Ah-Nm, Asd-Ut, Asd-Zd, and Asd-Shl.

As described in Section 4.4, we update the profit structure for trains at the stations Ah and Asd with penalty 10, 20, or 30, leading two 15 new profit structures (3 for the shift penalties at Asd, 3 for the shift penalties at Ah, and 9 for the combinations).

We run LH with the new profit structures and obtain 15 new timetables. When evaluating these solutions, we find an improved timetable with an evaluation value of 101.28, i.e., a reduction of 0.23 with respect to the BPL-timetable. This new timetable is referred to as the BF-timetable. Similar to the previous cases, we stop after the first improvement.

Fig. 14 displays the new evaluation contributions, both for the BPL-timetable and BF-timetable in a similar way as in previous sections. The OD-pairs that had a high increase in evaluation contribution now improved significantly: the increase in evaluation contribution for Ah-Nm reduced from +0.062 to 0 (see also the time–space diagram in Fig. 12(g)). For Asd-Ut, the increase in evaluation contribution reduced from +0.045 to +0.014 and for Asd-Zd it reduced from +0.039 to +0.027. The increase for Asd-Shl remained the same. The improvements for the first three mentioned OD-pairs are shown by means of arrows in Fig. 14(a).

Although the overall evaluation value of the new timetable improved and the contributions of the aforementioned three OD-pairs improved as well, now other OD-pairs have a higher evaluation contribution, as already indicated in Fig. 14(b). An improvement for some OD-pairs can indeed imply a worsening for others. This is also caused by the large number of trains considered in this network: when moving the schedule of one train to improve it, some other trains are affected. However, also for this instance the objective value of the BF-timetable is close to the ideal one (101.28 versus 100).

Summary. Table 3 and Fig. 15 summarize the progress of our approach on the extended A2 instance in terms of evaluation values for each of the steps.

Note that all timetables with comparatively bad evaluation values are timetables in which not all passengers can travel. Although this instance is more congested than the previous ones, also in this case the best timetable after feedback is close to the ideal one.

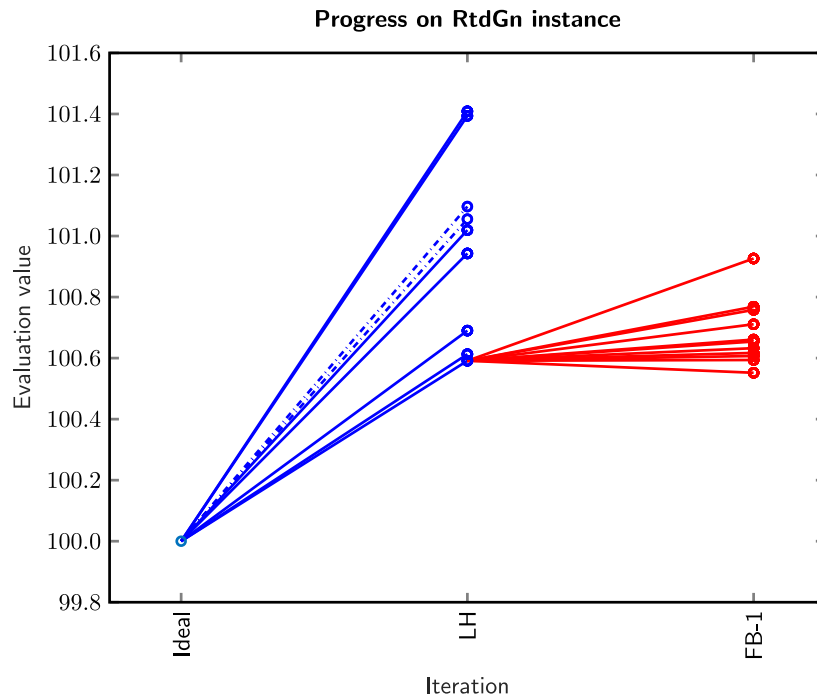


Fig. 11. Overview of the progress in Rotterdam–Groningen instance.

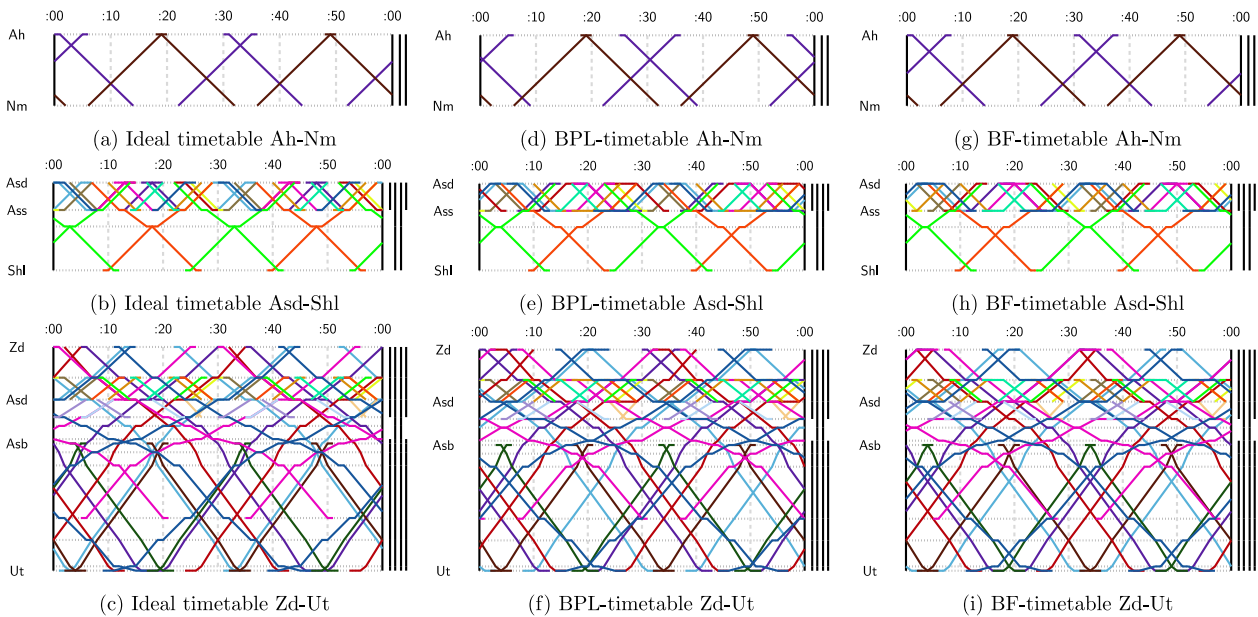


Fig. 12. Time-space diagrams for the extended A2 instance.

5.4. Comparison to a benchmark approach

In this section, we compare our approach to solving POT with the benchmark approach of modelling POT as a mixed-integer linear program. To make this comparison, we extend the MIP model proposed in Polinder et al. (2021) by infrastructure constraints (see the extended model in Appendix A.4). As is the PESP modelling approach, infrastructure constraints are modelled in the same way as driving time and dwelling time constraints, this adds many constraints to the model, but does not change the type of constraints included. We can therefore linearize the model in the same way as described in Polinder et al. (2021) (see also Section 4.1) and apply the heuristic stated there to attempt to solve it. Note that as solutions with cancelled trains were

not competitive in the instances that we consider, we do not implement the possibility to cancel trains in our benchmark approach.

The results of the comparison are displayed in Table 4. In this table we show, for each instance, the results obtained by applying several alternative approaches to find a feasible timetable. For each approach, we report the evaluation value of the best timetable it obtained, and the corresponding computing time. In particular, the first result (Ideal+LH) is the value of the best timetable obtained after computing an ideal timetable, and then running LH. The table reports the evaluation value of the timetable in the third column. The fourth column shows the time it took to compute this timetable: the time is split up in two parts, the first number shows the time spent on computing the ideal timetable, the other number shows the time spent on LH.

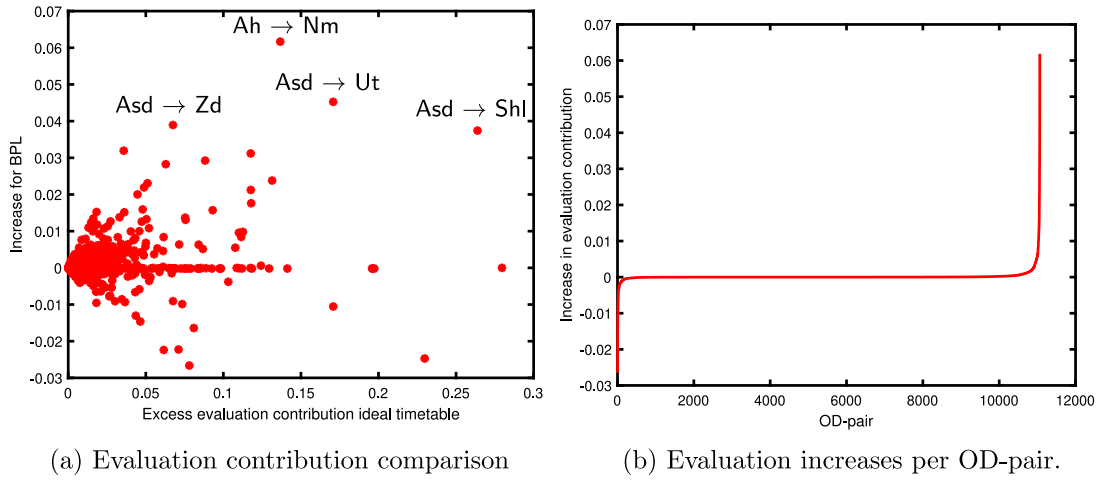


Fig. 13. Extended A2: Ideal timetable vs BPL-timetable.

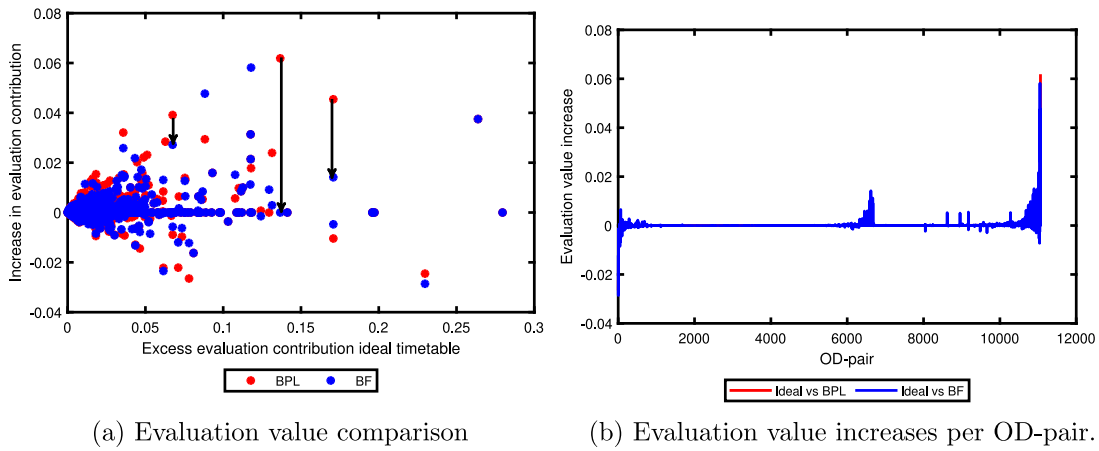


Fig. 14. Extended A2: Timetable comparisons after feedback.

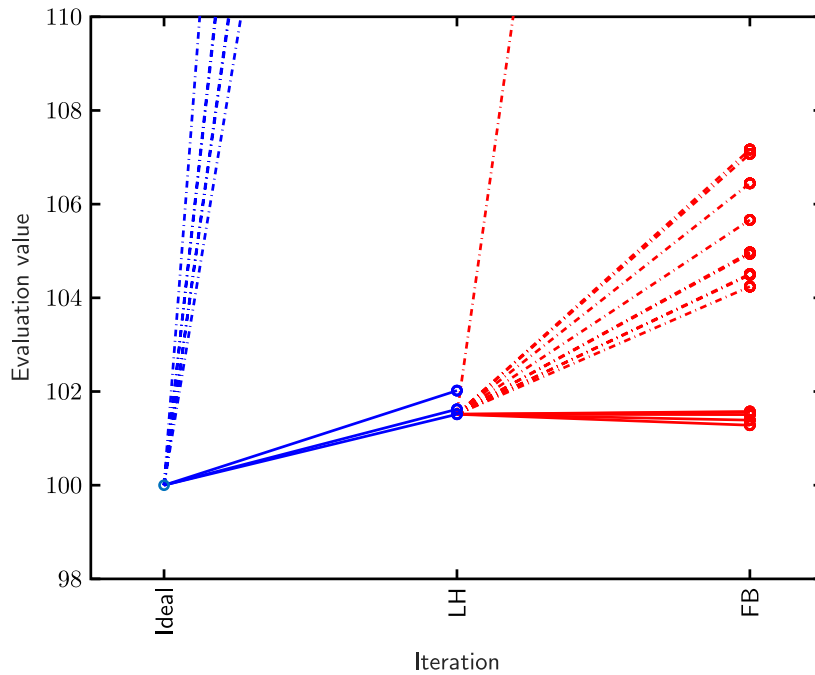


Fig. 15. Overview of the progress in the extended A2 instance.

Table 4
Benchmark results.

Instance	Approach	Evaluation value	Time (h)
A2	Ideal + LH	100.18	2 + 0.03
	Ideal + LH + FB	100.10	2 + 0.11
	POT as PESP		
	- After 2.11 h	105.80	2.11
	- After 8 h	104.88	8
	Lower bound CPLEX	97.09	
Rotterdam Groningen	Ideal + LH	100.59	4 + 0.06
	Ideal + LH + FB	100.55	4 + 0.18
	POT as PESP		
	- After 4.18 h	105.64	4.18
	- After 16 h	103.69	16
	Lower bound CPLEX	92.72	
Extended A2	Ideal + LH	101.51	4 + 0.14
	Ideal + LH + FB	101.28	4 + 0.49
	POT as PESP		
	- After 4.49 h	-	4.49
	- After 16 h	-	16
	Lower bound CPLEX	93.00	

The second result (Ideal+LH+FB) shown for each instance is the value of the best timetable obtained after computing an ideal timetable, then running LH, and finally applying the feedback process (i.e., the proposed iterative heuristic approach). In this case, the time reported in the fourth column includes the computing time needed for the feedback. The third result (POT as PESP) concerns the best timetable obtained after solving the integer programming formulation for POT including infrastructure constraints. In particular, we report the values of the best timetables obtained, respectively, in the same computing time required by the iterative heuristic approach (Ideal+LH+FB) and in the given time limit (8 or 16 h). For example, for the A2 instance, the iterative heuristic approach required 2.11 h, and, in the same computing time, the full POT model found a solution of value 105.80, while in the time limit (8 h) it obtained a solution of value 104.88. Finally, we display the lower bound as computed by CPLEX when solving the POT model until the time limit is reached. Note that the CPLEX lower bounds are stronger than those mentioned in Section 5.3 where we report lower bounds on the MILP model for finding the *ideal* timetable, because the POT model is more restrictive and therefore, combined with a longer computation time, a stronger lower bound is more likely to be obtained.

We observe that our approach is able to find better solutions in less time, even when no feedback is included. In particular, for the extended A2 instance, we were not able to find any feasible timetable within 16 h using the MILP formulation for POT, while the approach of this paper generates a reasonably good one within a bit more than two hours.

6. Conclusion and further research

In this paper, we proposed an approach to solve the tactical timetabling problem. Hereby we specifically focused on the quality of the timetable for the passengers.

In order to find a feasible passenger-oriented timetable for challenging real-world instances, for which the timetabling model itself already is challenging, we used variants of two existing approaches. These two approaches are combined into an algorithmic framework. First, an ideal timetable is computed, thereby neglecting infrastructure-related constraints. Next, through a Lagrangian heuristic, this timetable is modified to obtain a feasible timetable with respect to infrastructure. A feedback mechanism is used to improve the found solutions.

We showed that for real-life instances, based on the network operated by Netherlands Railway, we can obtain satisfying results. Furthermore, we showed that the provided feedback indeed leads to (overall) better timetables.

For the first step of our method, the construction of an ideal timetable, we use the approach described in Polinder et al. (2021). This approach linearizes the quadratic objective function, and then uses a ‘Relax-and-Fix’ heuristic to solve the formulation. It may be promising, in particular to improve time consumption, to investigate different approaches to find an ideal timetable, possibly working directly on the quadratic formulation of the problem.

One merit of our approach is that passenger demand is taken into account more accurately due to the inclusion of adaption time in the perceived travel time. On the other hand, however, the POT model relies on the assumption that demand would be uniformly distributed over the period in absence of a known timetable. It may be worthwhile to investigate to what extent this assumption is realistic, and, if it is not, to develop models that can handle other demand distributions, or that explicitly include the transitions between peak-demand and low-demand periods.

Train capacity (for transporting passengers) is not explicitly addressed in our timetabling problem, as this is traditionally solved in a subsequent step, the rolling stock scheduling, by assigning sufficient train units to individual train trips to meet the passenger demand. It could be an interesting extension to our model to take into account train capacity by explicitly scheduling individual train units, possibly over several periods. This extension would also allow evaluating the impact of train cancellations on the passenger demand, during the timetabling process.

CRedit authorship contribution statement

Gert-Jaap Polinder: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Valentina Cacchiani:** Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Marie Schmidt:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Dennis Huisman:** Conceptualization, Methodology, Supervision.

Acknowledgements

We thank Erasmus Trustfonds for supporting this work. Furthermore, we thank Gábor Maróti for providing several figures used in this paper. Valentina Cacchiani acknowledges the support by the Air Force Office of Scientific Research under award number FA8655-20-1-7019.

Appendix

A.1. Modelling timetabling constraints using the PESP model

A periodic timetable $(\pi_i)_{i \in V}$ is an assignment of times to events $i \in V$ that represent departures and arrivals of trains. To represent a feasible timetable, π has to fulfil a number of constraints. These are formulated based on the line plan, the infrastructure network, and time bounds on activities like driving, dwelling, and headway (when trains share part of the infrastructure).

It is convenient to represent these constraints using an *Event-Activity Network (EAN)*, in which the nodes are *events* and the arcs, called *activities*, model driving, dwelling, headways, and transfers, and correspond to the constraints of the PESP model.

Minimum driving times between stations, constrain the arrival time of a train to the preceding departure time. Furthermore, in stations we have minimum dwell times to let passengers board and alight. In macroscopic timetabling, a safe operation is ensured through headway constraints upon departure and arrival at stations. Headway constraints ensure that a minimum time between two consecutive trains is respected upon departure and arrival at a station, in order to avoid

collisions. We also consider overtaking constraints, ensuring a minimum headway time is respected on tracks between stations, when the headway constraints only are not enough to prevent overtakings. Finally, crossing constraints ensure that if two trains use a track in opposite directions, they can do this safely, without meeting each other halfway. As we consider a periodic timetable, these events are periodic, i.e., they re-occur every time period. In the most common mixed-integer linear programming (MILP) formulation for PESP, these constraints are expressed as

$$\ell_{ij} \leq \pi_j - \pi_i + T p_{ij} \leq u_{ij}. \tag{A.1}$$

Here, p_{ij} is a binary variable denoting the modulo operator, i.e., a shift from one period to the next, and ℓ_{ij} and u_{ij} are the lower and upper bounds on the activity durations. All aforementioned constraints can be formulated as a PESP-constraint. An overview of other activities that can be formulated in a PESP context can be found in [Liebchen and Möhring \(2007\)](#), [Kroon et al. \(2014\)](#).

As we use the event-activity network also for passenger routing, we also model transfer possibilities as activities in the event-activity network. To ensure that these do not impose constraints on timetable feasibility, we set the lower bound of these activities ℓ_{ij} to the minimum required transfer time, and the upper bound of these activities to $u_{ij} = \ell_{ij} + T - 1$. This choice ensures that the activity duration $\pi_j - \pi_i + T p_{ij}$ can take any value in the interval $[0, T - 1]$.

When for example $\pi_j = \pi_i + 2$ and $\ell_{ij} = 3$, the time difference between events i and j is two minutes, but also any multiple of T minutes can be added (due to the $T p_{ij}$ term in (A.1)). In this case, the transfer time is not 2 min, but 62 (when $T = 60$), since $\ell_{ij} = 3$ en $u_{ij} = 62$. By this way of modelling, the proper transfer times for passengers can be determined. The same principle holds when upper bounds of, for example, trip and dwell times are omitted. In this way, no operational constraints are added. In Section 4, when our model is explained, these transfer activities are used to determine the correct passenger paths and their perceived duration.

A.2. Time-space graph

To represent the train timetabling problem, we can use a time-space multi-graph $G = (W, Q)$, in which every node $w \in W$ is either a *departure node* or an *arrival node*, i.e., it corresponds to a departure or an arrival time of a train from/at a station along a track. We define W^j as the set of nodes of train j , $W_0^j \subset W^j$ the set of departure nodes of train j from its origin station and $W_f^j \subset W^j$ the set of arrival nodes of train j at its destination station ($j \in \mathcal{P}$). Arcs in set Q represent the travel of a train between two consecutive stations (*travel arcs*) or the stop of a train at a station (*dwell arcs*), and are partitioned into arc sets, one for each train. We call $Q^j \subset Q$ the set of arcs of train $j \in \mathcal{P}$, $QT^j \subset Q^j$ the set of travel arcs of train j , $QT_0^j \subset QT^j$ the set of travel arcs of train j from its origin station, and QS^j the set of dwelling arcs of train j .

A simple example of graph G for an instance with 4 stations and 2 trains is reported in Fig. 16. For sake of clarity, only very few arcs are reported. Train 1 has origin station S_1 and destination station S_4 , while train 2 has origin station S_2 and destination station S_4 . We use black colour for Train 1 and blue for Train 2. Set W^1 contains nodes w_1, \dots, w_8 and set W^2 nodes w_9, \dots, w_{16} . Node w_1 belongs to set W_0^1 , and nodes w_6 and w_8 to set W_f^1 . Nodes w_9 and w_{13} belong to set W_0^2 and nodes w_{12} and w_{16} to set W_f^2 . Travel arcs q_1, q_3, q_5 in set QT^1 represent the travel of train 1, while dwelling arcs q_2 and q_4 in QS^1 its stops at stations S_2 and S_3 , respectively. Dotted arc q_6 represents an increase of the dwelling time at station S_3 that can be used as an alternative arc to avoid train conflicts, and is followed by travel arc q_7 . For train 2, q_8 and q_{10} are travel arcs, while q_9 is a dwelling arc. Dotted arcs q_{11}, q_{12} and q_{13} correspond to alternative departure times from station S_2 that can be used to avoid train conflicts (in this case, the arrival and departure

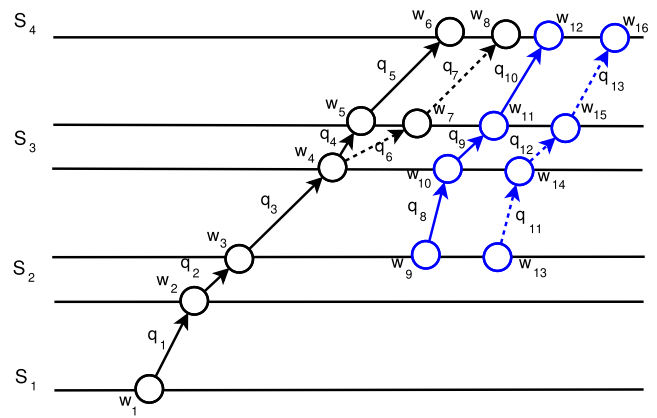


Fig. 16. An example of time-space graph G .

times, at all the stations visited by the train, are modified by the same amount). Set QT_0^1 contains q_1 , while QT_0^2 includes q_8 and q_{11} . A path in this time-space multi-graph corresponds to a timetable for one train that respects the train travel and dwell times, while, to determine a timetable that respects infrastructure constraints, additional constraints that avoid arc incompatibilities must be imposed (see A.5).

A.3. Precomputation of routes

In order to route passengers through the railway network when no timetable is known yet, we use a method described in [Warmerdam \(2004\)](#). The routes are generated based on a given line plan (specifying train lines, frequencies, and possible transfer options).

First, all direct travel options are determined. After this, all travel options with exactly one transfer are determined, then with two transfers, until a predefined maximum number of transfer options is reached. This leads to a large set of possible travel options. For each travel option, the expected duration is computed, which is the sum of the minimum travel and dwell times, plus for each transfer an estimate of transfer time. The result is multiplied by a percentage (generally 5%), to take uncertainty in trip durations into account.

We estimate transfer times for a transfer at a certain station s based on the number of direct connections from the origin to s , n_1 , and the number of direct connection from s to the destination, n_2 . More precisely, we set the estimated transfer time for transfer at s to $T/(2 \max\{n_1, n_2\})$. After generating routes as described above, we exclude dominated routes.

A.4. MIP model and approach to compute an ideal timetable

To compute an ideal timetable, we apply the approach described in [Polinder et al. \(2021\)](#). This approach makes use of mixed integer program (MIP) (A.2a)–(A.2n) to find an ideal timetable that meets driving time constraints and dwell time constraints and minimizes the total perceived travel time of passengers.

As before, also in this model variable π_i denotes the times of departure and arrival events $i \in V$. The timetabling constraints (A.1) for driving times and dwell times (with lower and upper bounds on activity durations) are rewritten as (A.2b)–(A.2c), introducing a new variable y_{ij} that stands for the duration of activity $(i, j) \in A$. As before, p_{ij} is a binary variable denoting the modulo operator, i.e., a shift from one period to the next. Note that although all our infrastructure-related constraints can be modelled as PESP constraints and could therefore be included in the form of constraints (A.2b)–(A.2c), we explicitly exclude them from the model formulation in this first step, to keep the model tractable.

The model uses a precomputed set of routes R as input, details can be found in Section A.3. Constraints (A.2d) set Y_r to the perceived duration of route r , where γ_r is the transfer penalty and 1_r is an indicator function that takes the value 1 if activity a is a transfer activity and 0 if it is not. For each OD-pair k and each relevant departure event for this OD-pair (that is, each departure event that denotes the start of a precomputed route for OD-pair k), variable L_v^k denotes the time difference between event v and the previous relevant departure event. These variables are set in constraints (A.2e)–(A.2f). Note that in constraints (A.2e), the time difference between π_v and $\pi_{v'}$ is computed modulo T : For $v, v' \in V$, $v \neq v'$, the binary variable $\alpha_{v',v}$ takes the value 1 if and only if $\pi_{v'} > \pi_v$, due to constraints (A.2e), (A.2f), and the non-negativity requirement on L_v^k in (A.2i). As a consequence, if for example $T = 60$, $\pi_{v'} = 40$ and $\pi_v = 9$, then $\pi_v - \pi_{v'} + T\alpha_{v',v} = 9 - 40 + 60 = 29$. Furthermore, note that (A.2f) makes sure that in case two events v and v' happen at the same time, exactly one is considered to ‘precede’ the other.

The average expected waiting time for all passengers of OD-pair k that depart on a route starting at event v is denoted as W_v^k and computed in (A.2g), based on the assumption that passenger arrivals are uniformly distributed over the period. Finally, in constraints (A.2h), for each OD-pair k and each relevant departure event v , we set variable Y_v^k to the perceived travel time of the best route from event v on towards the destination of OD-pair k . Note that the best route may start in event v , but it may also start in a later event v' : in the latter case, the term $(\pi_{v'} - \pi_v + T\alpha_{v',v})$ that represents the time difference between v and v' is non-zero, and enters the perceived duration, weighted with adaption time weight γ_w . Constraints (A.2i)–(A.2n) state the domains of the variables. In the objective (A.2a), we compute the total perceived travel time of passengers given timetable π as described in Section 3.3.

The model from Polinder et al. (2021) is summarized in (A.2a)–(A.2n).

$$\min \sum_{k \in \mathcal{OD}} \frac{d_k}{T} \sum_{v \in V^k} L_v^k \cdot (\gamma_w \cdot W_v^k + Y_v^k) \quad (\text{A.2a})$$

$$\text{s.t. } y_{ij} = \pi_j - \pi_i + T p_{ij} \quad \forall (i, j) \in A \quad (\text{A.2b})$$

$$\ell_{ij} \leq y_{ij} \leq u_{ij} \quad \forall (i, j) \in A \quad (\text{A.2c})$$

$$Y_r = \sum_{a \in r} (y_a + \gamma_r \cdot 1_r(a)) \quad \forall r \in \mathcal{R} \quad (\text{A.2d})$$

$$L_v^k = \min_{v' \in V^k \setminus \{v\}} \{\pi_v - \pi_{v'} + T\alpha_{v',v}\} \quad \forall k \in \mathcal{OD}, v \in V^k \quad (\text{A.2e})$$

$$\alpha_{v,v'} + \alpha_{v',v} = 1 \quad \forall k \in \mathcal{OD}, v \in V^k, v' \in V^k \setminus \{v\} \quad (\text{A.2f})$$

$$W_v^k = \frac{1}{2} L_v^k \quad \forall k \in \mathcal{OD}, \forall v \in V^k \quad (\text{A.2g})$$

$$Y_v^k = \min_{v' \in V^k} \min_{r \in \mathcal{R}_v^k} \{Y_r + \gamma_w \cdot (\pi_{v'} - \pi_v + T\alpha_{v',v})\} \quad \forall k \in \mathcal{OD}, v \in V^k \quad (\text{A.2h})$$

$$L_v^k \in [0, T] \quad \forall k \in \mathcal{OD}, v \in V^k \quad (\text{A.2i})$$

$$W_v^k \in [0, T/2] \quad \forall k \in \mathcal{OD}, v \in V^k \quad (\text{A.2j})$$

$$Y_r, Y_v^k \in \mathbb{R}_{\geq 0} \quad \forall r \in \mathcal{R}, k \in \mathcal{OD}, v \in V^k \quad (\text{A.2k})$$

$$\pi_v \in \{0, \dots, T-1\} \quad \forall v \in V \quad (\text{A.2l})$$

$$p_{ij} \in \mathbb{Z}_{\geq 0} \quad \forall (i, j) \in A \quad (\text{A.2m})$$

$$\alpha_{v,v'} \in \{0, 1\} \quad \forall k \in \mathcal{OD}, v \in V^k, v' \in V^k \setminus \{v\}. \quad (\text{A.2n})$$

Extensions to model POT as PESP

Model (A.2a)–(A.2n) can be extended to formulate the POT problem, by adding the infrastructure constraints, expressed in the same form as (A.2b)–(A.2c). In addition, it can be generalized by including the possibility of not scheduling all trains. In order to do so, we introduce some necessary notation. First of all, let L be the set of trains that have to be scheduled. Next, $V_l \subseteq V$ be the set of events that correspond to departures or arrivals of train $l \in L$. Then, let $A_l \subseteq A$ be the set of activities for which at least one of the involved events is

in V_l , i.e.:

$$A_l = \{(i, j) \in A : i \in V_l \text{ and/or } j \in V_l\}$$

Finally, we define an indicator function

$$1_l(a) = \begin{cases} 1 & \text{if } a \in A_l \\ 0 & \text{else} \end{cases}$$

The model in (A.2) can now be adapted by introducing binary variables z_l , indicating whether train $l \in L$ is scheduled or not. We replace (A.2c) by

$$\ell_{ij} - 1_l(a) \cdot T \cdot (1 - z_l) \leq y_{ij} \leq u_{ij} + 1_l(a) \cdot T \cdot (1 - z_l).$$

This means that activity $a \in A$ imposes no restriction if at least one of the involved trains is not scheduled. In order to ensure that no passenger is routed on a cancelled train, we update (A.2d) to

$$Y_r = \sum_{a \in r} \left(y_a + \gamma_r \cdot 1_r(a) + M \cdot \sum_{l \in L} 1_l(a) \cdot (1 - z_l) \right),$$

where M is a sufficiently large value. This ensures that passengers will not be routed in a route that involves an activity related to a train that is not scheduled (compare also Section 3.3).

A.5. Time-index ILP model

The time-index ILP model, based on the time–space graph introduced in A.2, contains one binary variable x_q^j for each arc $q \in Q^j$, $j \in \mathcal{P}$, that assumes value one if the arc is selected in the solution. The model requires to select arcs to form the best path for each train that is as close as possible to the ideal one. To this aim, each arc is assigned a profit φ_q^j , based on the profit structure defined in Section 4.2. For each train $j \in \mathcal{P}$, let $\bar{\varphi}^j$ be the train profit, β^j the shift penalty, ψ^j the stretch penalty, ms^{hj} the maximum shift value, and mst^j the maximum stretch value. In addition, let $\theta(w)$ be the time instant associated with node $w \in W$, and $\bar{\theta}^j(w)$ the ideal time instant of train j at node w (e.g., in Fig. 16, the ideal time instant at node w_7 is $\bar{\theta}^1(w_7) = \theta(w_5)$). In addition, let $q = (w_i, w_h) \in Q$ be an arc with tail node w_i and head node w_h . The arc profit is defined as follows:

- the travel arc profit φ_q^j for arc $q = (w_i, w_h)$ from the origin station of train j ($q \in QT_0^j$) is given by the train profit decreased by the shift penalty counted for every minute of shift incurred by that departure time: $\varphi_q^j = \bar{\varphi}^j - \beta^j |\theta(w_i) - \bar{\theta}^j(w_i)|$;
- the dwelling arc profit φ_q^j for station arc $q = (w_i, w_h)$ of train j ($q \in QS^j$) is zero, if that arc corresponds to the minimum dwell time, and otherwise is decreased by the stretch penalty counted for every minute of stretch incurred at that station: $\varphi_q^j = -\psi^j (\theta(w_h) - \theta(w_i) - (\bar{\theta}^j(w_h) - \bar{\theta}^j(w_i)))$, where $(\bar{\theta}^j(w_h) - \bar{\theta}^j(w_i))$ is the dwelling time in the ideal timetable.

In this way, the total train profit $\bar{\varphi}^j$ is gained if train j is scheduled according to its ideal timetable (path), while the profit is decreased for each operation that changes the ideal timetable. Clearly, if a train is cancelled, no profit is obtained.

Shift, stretch and cancellation are applied when train conflicts are present in the ideal timetable. Let C be the family of all sets C of pairwise incompatible arcs: two arcs $q_a = (w_{ia}, w_{ha})$ and $q_b = (w_{ib}, w_{hb})$ are incompatible if $|\theta(w_{ia}) - \theta(w_{ib})|$ or $|\theta(w_{ha}) - \theta(w_{hb})|$ is strictly smaller than the minimum headway time, or if the two arcs represent an overtaking between two trains along the same track, or they correspond to a crossing between two trains in opposite directions along the same track. Let b_{qC} be a binary value that is 1 if arc q is contained in set C , and $\delta^{-j}(w)$ and $\delta^{+j}(w)$ be, respectively, the set of entering and outgoing arcs of node $w \in W^j \setminus (W_0^j \cup W_f^j)$.

The ILP model reads as follows:

$$\max \sum_{j \in \mathcal{P}} \sum_{q \in Q^j} \varphi_q^j x_q^j \quad (\text{A.3})$$

$$\sum_{q \in Q^j} x_q^j \leq 1, \quad \forall j \in \mathcal{P}, \quad (\text{A.4})$$

$$\sum_{q \in \delta^-(w)} x_q^j = \sum_{q \in \delta^+(w)} x_q^j, \quad \forall w \in W^j \setminus (W_0^j \cup W_f^j), j \in \mathcal{P}, \quad (\text{A.5})$$

$$\sum_{j \in \mathcal{P}} \sum_{q \in Q^j} b_{qC} x_q^j \leq 1, \quad \forall C \in \mathcal{C}, \quad (\text{A.6})$$

$$x_q^j \in \{0, 1\}, \quad \forall q \in Q^j, j \in \mathcal{P}. \quad (\text{A.7})$$

The objective (A.3) is to maximize the total profit of all trains. The constraints require to select, for each train, arcs that form a path in the time–space graph (constraints (A.4) to allow at most one travel arc for each train from its origin station, and constraints (A.5) to build a path), and do not conflict with arcs selected for any other train, i.e., satisfy all infrastructure constraints (A.6).

A.6. Lagrangian Relaxation

Obviously, the difficulty of solving the problem comes from the latter constraints: therefore, LH applies a Lagrangian relaxation of all constraints (A.6).

Let λ_C ($C \in \mathcal{C}$) be non-negative Lagrangian multipliers associated with constraints (A.6). The Lagrangian relaxed model reads as follows:

$$\sum_{C \in \mathcal{C}} \lambda_C + \max \sum_{j \in \mathcal{P}} \sum_{q \in Q^j} (\varphi_q^j - \sum_{C \in \mathcal{C}} \lambda_C b_{qC}) x_q^j \quad (\text{A.8})$$

(A.4), (A.5), (A.7),

where $\xi_q^j = \varphi_q^j - \sum_{C \in \mathcal{C}} \lambda_C b_{qC}$ is the Lagrangian profit of arc $q \in Q^j$, $j \in \mathcal{P}$. In ξ_q^j , profit φ_q^j is defined as described in Appendix A.5, while λ_C ($C \in \mathcal{C}$) are the Lagrangian multipliers. The latter are initialized to zero, and then iteratively updated through subgradient optimization, based on whether each constraint (A.6) is violated or loose in the relaxed solution. In particular, to speed-up the process of multipliers updating, as in Caprara et al. (2002) and Cacchiani et al. (2010), constraints (A.6) are reformulated by using variables associated with the nodes of the time–space graph (instead of arc variables): indeed, the number of arc variables is very large, and each one would have associated its Lagrangian profit. Thanks to this reformulation, Lagrangian profits are instead associated with nodes of the time–space graph, and can be updated more quickly. In addition, at each iteration of the subgradient optimization process, only the constraints (A.6) violated by the current Lagrangian-relaxed solution are added to a constraint pool to avoid handling a large number of inactive constraints. We refer the reader to Caprara et al. (2002) and Cacchiani et al. (2010) for more details on the reformulation and on the developed subgradient optimization procedure.

References

Ait-Ali, A., Lindberg, P.O., Eliasson, J., Nilsson, J.-E., Peterson, A., 2020. A disaggregate bundle method for train timetabling problems. *J. Rail Transp. Plan. Manag.* 16, 100200.

Barrena, E., Canca, D., Coelho, L.C., Laporte, G., 2014a. Exact formulations and algorithm for the train timetabling problem with dynamic demand. *Comput. Oper. Res.* (ISSN: 0305-0548) 44, 66–74. <http://dx.doi.org/10.1016/j.cor.2013.11.003>.

Barrena, E., Canca, D., Coelho, L.C., Laporte, G., 2014b. Single-line rail rapid transit timetabling under dynamic passenger demand. *Transp. Res. B* (ISSN: 0191-2615) 70, 134–150. <http://dx.doi.org/10.1016/j.trb.2014.08.013>.

Belvaux, G., Boissin, N., Sutter, A., Wolsey, L.A., 1998. Optimal placement of add/drop multiplexers static and dynamic models. *Eur. J. Oper. Res.* (ISSN: 0377-2217) 108 (1), 26–35. [http://dx.doi.org/10.1016/S0377-2217\(97\)00021-0](http://dx.doi.org/10.1016/S0377-2217(97)00021-0).

Borndörfer, R., Hoppmann, H., Karbstein, M., 2017. Passenger routing for periodic timetable optimization. *Public Transp.* 9 (1–2), 115–135. <http://dx.doi.org/10.1007/s12469-016-0132-0>.

Brännlund, U., Lindberg, P.O., Nou, A., Nilsson, J.-E., 1998. Railway timetabling using Lagrangian relaxation. *Transp. Sci.* 32 (4), 358–369.

Cacchiani, V., Caprara, A., Toth, P., 2008. A column generation approach to train timetabling on a corridor. *4OR* 6 (2), 125–142.

Cacchiani, V., Caprara, A., Toth, P., 2010. Scheduling extra freight trains on railway networks. *Transp. Res. B* 44 (2), 215–231. <http://dx.doi.org/10.1016/j.trb.2009.07.007>.

Cacchiani, V., Toth, P., 2012. Nominal and robust train timetabling problems. *Eur. J. Oper. Res.* 219 (3), 727–737. <http://dx.doi.org/10.1016/j.ejor.2011.11.003>.

Caimi, G., Kroon, L., Liebchen, C., 2017. Models for railway timetable optimization: Applicability and applications in practice. *J. Rail Transp. Plan. Manag.* (ISSN: 2210-9706) 6 (4), 285–312. <http://dx.doi.org/10.1016/j.jrtpm.2016.11.002>.

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. *Oper. Res.* 50 (5), 851–861. <http://dx.doi.org/10.1287/opre.50.5.851.362>.

de Keizer, B., Kouwenhoven, M., Hofker, F., 2015. New insights in resistance to interchange. *Transp. Res. Proc.* (ISSN: 2352-1465) 8, 72–79. <http://dx.doi.org/10.1016/j.trpro.2015.06.043>, Current practices in transport: appraisal methods, policies and models - 42nd European Transport Conference Selected Proceedings.

de Dios Ortúzar, J., Willumsen, L.G., 2011. *Modelling Transport*. John Wiley & Sons.

Farina, F., 2018. *Optimization of Operations in Public Transportation*. (Ph.D. thesis). DTU.

Gattermann, P., Großmann, P., Nachtigall, K., Schöbel, R.A., 2016. Integrating passengers' routes in periodic timetabling: A SAT approach. In: Goerigk, M., Werneck, R. (Eds.), 16th Workshop On Algorithmic Approaches For Transportation Modelling, Optimization, And Systems, Vol. 54. ATMOS 2016, In: OpenAccess Series in Informatics (OASISs), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 3:1–3:15. <http://dx.doi.org/10.4230/OASISs.ATMOS.2016.3>, URL <http://drops.dagstuhl.de/opus/volltexte/2016/6527>.

Hartleb, J., Schmidt, M., Friedrich, M., Huisman, D., 2019. A good or a bad timetable: Do different evaluation functions agree? ERIM report series research in management erasmus research institute of management. ERS-2019-002-LISURL <http://hdl.handle.net/1765/115831>.

IBM, 2019. *IBM ILOG CPLEX Optimization studio 12.9.0*.

Kroon, L.G., Peeters, L.W.P., Wagenaar, J.C., Zuidwijk, R.A., 2014. Flexible connections in PESP models for cyclic passenger railway timetabling. *Transp. Sci.* 48 (1), 136–154. <http://dx.doi.org/10.1287/trsc.1120.0453>.

Liebchen, C., 2008. The first optimized railway timetable in practice. *Transp. Sci.* 42 (4), 420–435. <http://dx.doi.org/10.1287/trsc.1080.0240>.

Liebchen, C., Möhring, R., 2007. The modeling power of the periodic event scheduling problem: Railway timetables – and beyond. In: Geraets, F., Kroon, L., Schöbel, A., Wagner, D., Zoroiliadis, C. (Eds.), *Algorithmic Methods For Railway Optimization: International Dagstuhl Workshop, Dagstuhl Castle, Germany, June 20-25, 2004, 4th International Workshop, ATMOS 2004, Bergen, Norway, September 16-17, 2004, Revised Selected Papers*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 3–40. http://dx.doi.org/10.1007/978-3-540-74247-0_1.

Liebchen, C., Peeters, L., 2009. Integral cycle bases for cyclic timetabling. *Discrete Optim.* (ISSN: 1572-5286) 6 (1), 98–109. <http://dx.doi.org/10.1016/j.disopt.2008.09.003>.

Lübbe, J., 2009. *Passagierrouting und Taktfahrplanoptimierung*. (Ph.D. thesis). Technische Universität Berlin.

Lusby, R., Larsen, J., Bull, S., 2018. A survey on robustness in railway planning. *Eur. J. Oper. Res.* (ISSN: 0377-2217) 266 (1), 1–15. <http://dx.doi.org/10.1016/j.ejor.2017.07.044>.

Martin-Iradi, B., Røpke, S., 2021. A column-generation-based matheuristic for periodic and symmetric train timetabling with integrated passenger routing. *Eur. J. Oper. Res.* 2, 511–531.

Matos, G.P., Albino, L.M., Saldanha, R.L., Morgado, E.M., 2018. Solving periodic timetabling problems with SAT and machine learning. In: *Conference On Advanced Systems In Public Transp.*

Nachtigall, K., 1994. A branch and cut approach for periodic network programming. In: *Hildesheimer Informatik-Berichte. Technical Report 29*, Univ., Inst. für Mathematik.

Nachtigall, K., Opitz, J., 2008. Solving periodic timetable optimization problems by modulo simplex calculations. In: Fischetti, M., Widmayer, P. (Eds.), 8th Workshop On Algorithmic Approaches For Transportation Modeling, Optimization, And Systems. ATMOS'08, In: OpenAccess Series in Informatics (OASISs), vol. 9, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, ISBN: 978-3-939897-07-1, <http://dx.doi.org/10.4230/OASISs.ATMOS.2008.1588>.

Niu, H., Zhou, X., Gao, R., 2015. Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: Nonlinear integer programming models with linear constraints. *Transp. Res. B* 76, 117–135.

NS, 2019. Website Nederlandse Spoorwegen, <http://www.ns.nl>.

Odijk, M., 1996. A constraint generation algorithm for the construction of periodic railway timetables. *Transp. Res. B* (ISSN: 0191-2615) 30 (6), 455–464. [http://dx.doi.org/10.1016/0191-2615\(96\)00005-7](http://dx.doi.org/10.1016/0191-2615(96)00005-7).

Peeters, L., 2003. *Cyclic Railway Timetable Optimization*. (Ph.D. thesis). Erasmus University Rotterdam.

Polinder, G.J., Schmidt, M., Huisman, D., 2021. Timetabling for strategic passenger railway planning. *Transp. Res. B* (ISSN: 0191-2615) 146, 111–135. <http://dx.doi.org/10.1016/j.trb.2021.02.006>, URL <https://www.sciencedirect.com/science/article/pii/S0191261521000278>.

Radtke, A., 2014. Infrastructure modelling. In: Hansen, I., Pachl, J. (Eds.), *Railway Timetable & Operations*. DDV Media Group.

- Schiewe, P., Schöbel, A., 2020. Periodic timetabling with integrated routing: Toward applicable approaches. *Transp. Sci.* 54 (6), 1714–1731.
- Schmidt, M., Schöbel, A., 2015. Timetabling with passenger routing. *OR Spectrum* (ISSN: 0171-6468) 37 (1), 75–97. <http://dx.doi.org/10.1007/s00291-014-0360-0>.
- Serafini, P., Ukovich, W., 1989. A mathematical model for periodic scheduling problems. *SIAM J. Discrete Math.* 2 (4), 550–581. <http://dx.doi.org/10.1137/0402049>.
- Siebert, M., Goerigk, M., 2013. An experimental comparison of periodic timetabling models. *Comput. Oper. Res.* (ISSN: 0305-0548) 40 (10), 2251–2259. <http://dx.doi.org/10.1016/j.cor.2013.04.002>.
- Wang, Y., Tang, T., Ning, B., van den Boom, T.J., De Schutter, B., 2015. Passenger-demands-oriented train scheduling for an urban rail transit network. *Transp. Res. C* 60, 1–23.
- Warmerdam, J., 2004. Specificaties TRANS toedeler. Technical report, QQQ Delft.
- Wolsey, L., 1998. *Integer Programming*. In: *Wiley Series in Discrete Mathematics and Optimization*, Wiley, ISBN: 9780471283669, URL <https://books.google.nl/books?id=x7RvQgAACAAJ>.
- Yin, J., Yang, L., Tang, T., Gao, Z., Ran, B., 2017. Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. *Transp. Res. B* 97, 182–213.
- Zhang, Y., Peng, Q., Yao, Y., Zhang, X., Zhou, X., 2019. Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and alternating direction method of multipliers methods. *Transp. Res. B* 128, 344–379.
- Zhu, Y., Mao, B., Bai, Y., Chen, S., 2017. A bi-level model for single-line rail timetable design with consideration of demand and capacity. *Transp. Res. C* 85, 211–233.