A Matching Approach for Periodic Timetabling*

Julius Pätzold[1] and Anita Schöbel[2]

1 Institut für Numerische und Angewandte Mathematik, Georg August University Göttingen, Göttingen, Germany
paetzold@stud.uni-goettingen.de

2 Institut für Numerische und Angewandte Mathematik, Georg August University Göttingen, Göttingen, Germany
schoebel@math.uni-goettingen.de

Abstract

The periodic event scheduling problem (PESP) is a well studied problem known as intrinsically hard, but with important applications mainly for finding good timetables in public transportation. In this paper we consider PESP in public transportation, but in a reduced version (r-PESP) in which the driving and waiting times of the vehicles are fixed to their lower bounds. This results in a still NP-hard problem which has less variables, since only one variable determines the schedule for a whole line. We propose a formulation for r-PESP which is based on scheduling the lines. This enables us on the one hand to identify a finite candidate set and an exact solution approach. On the other hand, we use this formulation to derive a matching-based heuristic for solving PESP. Our experiments on close to real-world instances from LinTim show that our heuristic is able to compute competitive timetables in a very short runtime.

1998 ACM Subject Classification G.1.6 Optimization, G.2.2 Graph Theory, G.2.3 Applications

Keywords and phrases PESP, Timetabling, Public Transport, Matching, Finite Dominating Set

Digital Object Identifier 10.4230/OASIcs.ATMOS.2016.1

1 PESP: The Periodic Event Scheduling Problem

The *Periodic Event Scheduling Problem* (PESP) in which events have to be scheduled periodically is a complex and well-known discrete problem with interesting real-world applications. It has been introduced in [17]. The PESP is known to be NP hard - in fact, even finding a feasible solution is so. The PESP can be formulated as linear mixed-integer program and has been extensively studied. Still, even heuristics are rare and suffer under high empirical run times. Nevertheless using constraint programming techniques, [7] were able to support the decision process of the Netherlands Railway (NS) using the PESP model, and the basic concept of the 2005 timetable of Berlin Underground has been computed in [9]. Solution approaches include constraint generation [14], techniques using the cycle space (see [11, 16, 8]), or the modulo-simplex heuristic [12, 3]. Recently SAT-solvers proved to be successful for solving the PESP [4]. Under research is the construction of timetables under uncertainty, see, e.g., [6, 1].

We start by giving the mathematical formulation of PESP, its interpretation in the context of public transportation will be provided in Section 2. Let an event-activity network $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ with nodes (or events) $\mathcal{E}$ and directed arcs (or activities) $\mathcal{A}$ be given. We want to assign a time π_i to every event $i \in \mathcal{E}$. For setting up feasibility constraints, we furthermore

* This work was partially supported by DFG under grant SCHO1140/8-1

16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'16).
Editors: Marc Goerigk and Renato Werneck; Article No. 1; pp. 1:1–1:15
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

assume time spans $\Delta_a = [L_a, U_a]$ with a lower bound L_a and an upper bound U_a for all activities $a \in \mathcal{A}$, and weights w_a which represent the importance of activity $a \in \mathcal{A}$. Finally, we need a period $T \in \mathbb{N}$. An instance I of PESP is hence given by $\mathcal{N}, w, L, U, T$. Defining

$$[x]_T := \min\{x - zT : z \in \mathbb{Z}, x - zT \geq 0\},$$

PESP can be formulated as

$$\begin{array}{lllll} (\textbf{PESP}) & \min \displaystyle\sum_{a=(i,j)\in\mathcal{A}} w_a[\pi_j - \pi_i - L_a]_T & & & \\ & \text{s.t. } [\pi_j - \pi_i - L_a]_T & \in & [0, U_a - L_a] & \text{for all } a \in \mathcal{A} \\ & \pi_i & \in & \{0, 1, \ldots, T-1\} & \text{for all } i \in \mathcal{E}. \end{array}$$

The variables π_i assign a point of time to each event $i \in \mathcal{E}$. This time is usually assumed to be integer (in minutes) and takes only values in $\{0, 1, \ldots, T-1\}$ since it is repeated periodically with a period of T. Note that the PESP only looks at the differences of the π values, hence one of the variables can always be fixed, e.g., $\pi_1 := 0$.

The objective function minimizes the sum of slack times over all activities of the resulting periodic schedule while the constraints ensure that the minimal duration L_a and maximal duration U_a of all activities $a = (i, j) \in \mathcal{A}$ are respected by the periodic schedule. Note that $[\pi_j - \pi_i - L_a]_T \in [0, U_a - L_a]$ is equivalent to $L_a \leq \pi_j - \pi_i + z_a T \leq U_a$ for some integer $z_a \in \mathbb{Z}$ which can be used to linearize the formulation given above to receive a linear integer program. For details on the periodicity and the meaning of the time spans Δ_a we refer to the extensive literature on PESP.

Our contribution. In this paper we study the PESP in the context of its main application, namely for timetabling in public transportation. We use the special underlying structure of the event-activity network to design an exact and a heuristic approach for solving the PESP in this case.

2 r-PESP: The reduced periodic event scheduling problem in public transportation

We first repeat how the event-activity network is constructed for the case of periodic timetabling in public transportation.

Given a set of traffic lines $\mathcal{L}$, the event-activity network $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ consists of nodes $\mathcal{E} = \mathcal{E}_{arr} \cup \mathcal{E}_{dep}$ which are called *arrival and departure events* and of edges $\mathcal{A} = \mathcal{A}_{drive} \cup \mathcal{A}_{wait} \cup \mathcal{A}_{trans}$ called *driving activities, waiting activities* and *transfer activities*. These are constructed as follows (see, e.g., [11, 8]):

- Let $l \in \mathcal{L}$ be a line passing through stations $s_1, s_2, \ldots, s_p$. Such a line corresponds to $p-1$ arrival and to $p-1$ departure events (s_1, l, dep), (s_2, l, arr), (s_2, l, dep), …, (s_p, l, arr).
- A departure event (s_i, l, dep) and its consecutive arrival event (s_{i+1}, l, arr) on the same line l at its next station are linked by a directed driving activity. Waiting activities link an arrival event of a line (s_i, l, arr) and its consecutive departure event (s_i, l, dep) at the same station.
- Transfer activities connect an arrival event (s, l, arr) of one line l at some station s to a departure event (s, k, dep) of another line k at the same station s if a transfer for the passengers should be possible here.

Note that in railway applications also headway activities are needed which ensure a minimal distance between two consecutive trains on the same piece of infrastructure.

In the PESP formulation, the L_a describe lower bounds on the activities, i.e., the minimal driving time for driving activities, the minimal dwell time at stations for waiting activities and the minimal time needed for a transfer (i.e., getting off the train, changing the platform and boarding the next train) for the passengers for transfer activities. The weights w_a give the number of passengers who use activity $a \in \mathcal{A}$. Minimizing the sum of all slack times in PESP hence can be interpreted as minimizing the sum of all traveling times over the passengers.

2.1 r-PESP in public transportation

In public transportation it is often assumed that there are no upper bounds for transfer activities, since a passenger can always take the train of the next period, and the objective function aims at minimizing the transfer slack times anyway. We will also use this assumption here, i.e., that

$$\Delta_a = [L_a, L_a + T - 1] \text{ for all } a \in \mathcal{A}_{trans}. \tag{1}$$

As mentioned above, in the practice of public transportation planning, every event $i \in \mathcal{E}$ belongs to exactly one line $l \in \mathcal{L}$. Hence, the events $\mathcal{E}$ of the event-activity network can be partitioned into the lines they belong to, i.e.,

$$\mathcal{E} = \bigcup_{l \in \mathcal{L}} \mathcal{E}_l.$$

Every line l induces a subgraph $\mathcal{N}_l = (\mathcal{E}_l, \mathcal{A}_l)$ with $\mathcal{A}_l \subseteq \mathcal{A}_{wait} \cup \mathcal{A}_{drive}$, i.e., $\mathcal{A}_l$ consists only of waiting and driving activities. Solving PESP on such a subgraph is easy: The optimal solution is to fix all driving and waiting activities to their lower bounds. This motivates the formulation of a *reduced PESP* in which we require that all driving and waiting times are fixed to their lower bounds. This can formally be done by setting

$$\Delta_a := [L_a, L_a] \text{ for all } a \in \mathcal{A}_{wait} \cup \mathcal{A}_{drive}. \tag{2}$$

Using both the assumptions (1) and (2) we obtain the following straightforward formulation for the *reduced PESP* in which we fix the length of all waiting and driving activities to their lower bounds and do not have any restriction on the transfer activities. The latter are the only activities which are then relevant in the objective function.

$$\begin{aligned}
(\mathbf{r-PESP}) \quad \min \sum_{a=(i,j) \in \mathcal{A}_{trans}} & w_a [\pi_j - \pi_i - L_a]_T \\
\text{s.t. } [\pi_j - \pi_i - L_a]_T &= 0 \text{ for all } a \in \mathcal{A}_{drive} \cup \mathcal{A}_{wait} \\
\pi_i &\in \{0, 1, \ldots, T-1\} \text{ for all } i \in \mathcal{E}.
\end{aligned}$$

Fixing the driving and waiting activities to their lower bounds has been done in other publications before. In [13] it has been shown that the resulting problem is still NP-hard. A theoretical analysis of the error which is made by fixing the values of the waiting and driving activities to their lower bounds is provided in the next section, an experimental evaluation can be found in Section 4.

2.2 Comparing PESP and r-PESP

We are interested in a bound on the error which is made by fixing the driving and waiting activities to their lower bounds. To this end, we denote the objective values of PESP and r-PESP by PESP(I), or r-PESP(I), respectively. Then the *gap* between the reduced version of PESP and the original PESP is specified by

$$\mathrm{Gap} := \sup_{\text{Instances I}} \text{r-PESP(I)} - \text{PESP(I)},$$

where for an instance $I = (\mathcal{N}, w, L, U, T)$ r-PESP(I) is defined by (2), i.e., the time windows for waiting and driving activities a are set to $\Delta_a := [L_a, L_a]$.

▶ **Lemma 1.** *If (1) holds we have*

$$0 \leq \mathrm{Gap} \leq \frac{\mathrm{T}}{2} \sum_{\mathrm{a} \in \mathcal{A}_{\mathrm{trans}}} \mathrm{w_a}$$

Proof.

- Under assumption (1) we have that every feasible solution of r-PESP is also feasible for PESP and satisfies $\sum_{a=(i,j)\in\mathcal{A}} w_a[\pi_j - \pi_i - L_a]_T = \sum_{a=(i,j)\in\mathcal{A}_{trans}} w_a[\pi_j - \pi_i - L_a]_T$, hence PESP is a relaxation of r-PESP. This gives that Gap ≥ 0.
- On the other hand, it can be shown that for every instance I of r-PESP we have r-PESP(I) $\leq \frac{T}{2} \sum_{a\in\mathcal{A}_{trans}} w_a$, i.e., any optimal solution is bounded by the waiting time which would be received if trains are scheduled according to a uniform distribution (for details, see [15, 13]). We hence obtain r-PESP(I) $-$ PESP(I) $\leq \frac{T}{2} \sum_{a\in\mathcal{A}_{trans}} w_a - 0$ for all instances I, and hence Gap $\leq \frac{\mathrm{T}}{2} \sum_{\mathrm{a}\in\mathcal{A}_{\mathrm{trans}}} \mathrm{w_a}$.

◀

We will also see in the experiments that solving r-PESP seems to be a very good heuristic for finding PESP solutions.

2.3 An equivalent formulation for r-PESP and a finite candidate set

We now consider some line $l \in \mathcal{L}$ with its corresponding events $\mathcal{E}_l$. We need the following notation.

- For every line l, let *first*(l) denote the first event of line l.
- For every event $i \in \mathcal{E}_l$ of line l define *dur*(i) as the length of the (unique) path from *first*(l) to i in the subnetwork $\mathcal{N}_l$ with edge weights L_a. This is the duration which a vehicle of line l needs from its start to event i. Note that *dur*(i) is well defined since every event i belongs to exactly one line l.
- For two lines $k, l \in \mathcal{L}$ define $\mathcal{A}_{trans}(k,l) := \{a = (i,j) \in \mathcal{A}_{trans} : i \in \mathcal{E}_k, j \in \mathcal{E}_l\}$ as the (possibly empty) set of transfer activities from line k to line l.

If $\pi_{\mathit{first}(l)}$ is fixed for the first event of line l, the resulting arrival and departure times for all other events $i \in \mathcal{E}_l$ in r-PESP can be determined as

$$\pi_i := \pi_{\mathit{first}(l)} + \mathit{dur}(i) \quad \text{for all } i \in \mathcal{E}_l.$$

Plugging this into r-PESP, the objective function can be transformed to

$$\begin{aligned}
&\sum_{a=(i,j)\in\mathcal{A}_{trans}} w_a[\pi_j - \pi_i - L_a]_T \\
&\quad = \sum_{k,l\in\mathcal{L}} \sum_{a=(i,j)\in\mathcal{A}_{trans}(k,l)} w_a[\pi_{\mathit{first}(l)} + \mathit{dur}(j) - \pi_{\mathit{first}(k)} - \mathit{dur}(i) - L_a]_T \\
&\quad := \sum_{k,l\in\mathcal{L}} \sum_{a\in\mathcal{A}_{trans}(k,l)} w_a[d_a + \pi_{\mathit{first}(l)} - \pi_{\mathit{first}(k)}]_T
\end{aligned}$$

where

$$d_a := [\mathit{dur}(j) - \mathit{dur}(i) - L_a]_T \text{ for } a = (i,j) \in \mathcal{A}_{trans}.$$

We abbreviate $\pi_l := \pi_{\mathit{first}(l)}$ emphasizing that we now determine only one point of time for every line $l \in \mathcal{L}$. Given π_l for all lines $l \in \mathcal{L}$ we furthermore denote

$$\begin{aligned}
f_{k,l}(\pi) &:= \sum_{a\in\mathcal{A}_{trans}(k,l)} w_a[d_a + \pi_l - \pi_k]_T \\
f(\pi) &:= \sum_{k,l\in\mathcal{L}} f_{k,l}(\pi)
\end{aligned}$$

r-PESP can hence be equivalently formulated as

$$\begin{aligned}
(\mathbf{r-PESP}) \quad \min & \sum_{a\in\bigcup_{k,l\in\mathcal{L}}\mathcal{A}_{trans}(k,l)} w_a[\pi_l - \pi_k + d_a]_T \\
\text{s.t. } & \pi_l \in \{0,1,\ldots,T-1\} \text{ for all } l \in \mathcal{L}
\end{aligned}$$

which is a PESP on a reduced event-activity network, without any feasibility requirements, but with possibly multiple activities between every pair of events. In the following, when we talk about a solution to r-PESP we mean a solution $\pi \in \{0,\ldots,T-1\}^{|\mathcal{L}|}$ to the above reformulation r-PESP. We now illustrate this reformulation on two special cases which will be used later in our algorithmic approach.

An optimal timetable for the case of two lines. For only two lines $\mathcal{L} = \{k,l\}$ we receive a problem with two variables $\pi_k, \pi_l \in \{0,1,\ldots,T-1\}$. We can further reduce its objective function to only one variable by computing

$$\begin{aligned}
f(\pi_k,\pi_l) &= f_{k,l}(\pi_k,\pi_l) + f_{l,k}(\pi_k,\pi_l) \\
&= \sum_{a\in\mathcal{A}_{trans}(k,l)} w_a[d_a + \pi_l - \pi_k]_T + \sum_{a\in\mathcal{A}_{trans}(l,k)} w_a[d_a + \pi_k - \pi_l]_T
\end{aligned}$$

and substituting $t := \pi_k - \pi_l$ due to the fact that we can set e.g., $\pi_l := 0$ and then receive $t := \pi_k - \pi_l = \pi_k$. We obtain

$$\begin{aligned}
\min_{t=0,\ldots,T-1} g(t) &:= \sum_{a\in\mathcal{A}_{trans}(k,l)} w_a[d_a - t]_T + \sum_{a\in\mathcal{A}_{trans}(l,k)} w_a[d_a + t]_T \\
&= \sum_{a\in\mathcal{A}_{trans}(k,l)} w_a[\bar{d}_a - t]_T + \sum_{a\in\mathcal{A}_{trans}(l,k)} w_a[\bar{d}_a + t]_T \qquad (3)
\end{aligned}$$

with $\bar{d}_a := [d_a]_T \in \{0,\ldots,T-1\}$ for all $a \in \mathcal{A}_{trans}$, since adding an integer multiple of T in $[d_a - t]_T$ or in $[d_a + t]_T$ does not change their values.

Optimal adjustment of two line clusters. A similar situation appears if we have a partition of the set of all lines $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ into two disjoint line clusters $\mathcal{L}_1, \mathcal{L}_2$. Suppose, a timetable $\pi_l, l \in \mathcal{L}$ is given. We want to adjust the two clusters such that they fit as good as possible to each other without changing the synchronization between any pair of lines within the same cluster. This can be done by shifting all lines in $\mathcal{L}_1$ by an amount of t minutes. The new timetable $\pi(\mathcal{L}_1, t)$ is then given by

$$\pi(\mathcal{L}_1, t)_l := \begin{cases} \pi_l & \text{if } l \in \mathcal{L}_2 \\ \pi_l + t & \text{if } l \in \mathcal{L}_1 \end{cases} \tag{4}$$

We are now interested in the best t, i.e., the optimal shift between the two clusters. The objective function $f(\pi(\mathcal{L}_1, t))$ is only dependent on t and can hence be simplified to

$$\begin{aligned} g(t) &:= g(\pi(\mathcal{L}_1, t)) \\ &= \sum_{k,l \in \mathcal{L}_1} f_{k,l}(\pi) + \sum_{k,l \in \mathcal{L}_2} f_{k,l}(\pi) + \sum_{k \in \mathcal{L}_1, l \in \mathcal{L}_2} \sum_{a \in \mathcal{A}_{trans}(k,l)} w_a[\pi_l - (\pi_k + t) + d_a]_T \\ &\quad + \sum_{k \in \mathcal{L}_2, l \in \mathcal{L}_1} \sum_{a \in \mathcal{A}_{trans}(k,l)} w_a[(\pi_l + t) - \pi_k + d_a]_T \\ &= const + \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_1, l \in \mathcal{L}_2}} w_a[\bar{d}_a - t]_T + \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_2, l \in \mathcal{L}_1}} w_a[\bar{d}_a + t]_T \end{aligned} \tag{5}$$

with $\bar{d}_a := [d_a + \pi_l - \pi_k]_T \in \{0, \ldots, T-1\}$ for all $a \in \mathcal{A}_{trans}(k, l)$, $k, l \in \mathcal{L}$.

Note that using this formula one directly sees that

$$g(\pi(\mathcal{L}_1, t)) = g(\pi(\mathcal{L}_2, T - t)). \tag{6}$$

The following lemma applies to solving problems of type (3) or (5).

▶ **Lemma 2.** *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two disjoint sets and let $d_a \in \{0, \ldots, T-1\}$, $w_a \geq 0$ for all $a \in \mathcal{A}_1 \cup \mathcal{A}_2$. Consider the optimization problem*

$$(P) \qquad \min_{t \in \{0,1,\ldots,T-1\}} g(t) := \sum_{a \in \mathcal{A}_1} w_a[d_a - t]_T + \sum_{a \in \mathcal{A}_2} w_a[d_a + t]_T.$$

Then there exists an optimal solution t^* to (P) which satisfies

$$t^* \in \{d_a : a \in \mathcal{A}_1\} \cup \{T - d_a : a \in \mathcal{A}_2\}.$$

Furthermore,

- *$t^* \in \{d_a : a \in \mathcal{A}_1\}$ for* all *optimal solutions t^* to (P) if $\sum_{a \in \mathcal{A}_1} w_a > \sum_{a \in \mathcal{A}_2} w_a$,*
- *$t^* \in \{T - d_a : a \in \mathcal{A}_2\}$ for* all *optimal solutions t^* to (P) if $\sum_{a \in \mathcal{A}_1} w_a < \sum_{a \in \mathcal{A}_2} w_a$.*

Proof. The first part of the lemma was already observed by [13]. For the second part, let $\sum_{a \in \mathcal{A}_1} w_a > \sum_{a \in \mathcal{A}_2} w_a$ and consider $t \in \{0, 1, \ldots, T-1\}$. Let $t \notin \{d_a : a \in \mathcal{A}_1\}$. Increasing t to $t+1$ gives $[d_a - t]_T - [d_a - (t+1)]_T = 1$ and

$$[d_a + t]_T - [d_a + (t+1)]_T = \begin{cases} -1 & \text{if } t \neq T - d_a - 1 \\ T - 1 \geq -1 & \text{if } t = T - d_a - 1, \end{cases}$$

hence

$$\begin{aligned} g(t) - g(t+1) &= \sum_{a \in \mathcal{A}_1} w_a \left([d_a - t]_T - [d_a - (t+1)]_T \right) \\ &\quad + \sum_{a \in \mathcal{A}_2} w_a \left([d_a + t]_T - [d_a + (t+1)]_T \right) \\ &\geq \sum_{a \in \mathcal{A}_1} w_a - \sum_{a \in \mathcal{A}_2} w_a > 0, \end{aligned}$$

i.e., increasing t improves the objective function value and t can hence not be optimal. The other direction works analogously. ◀

This means the problem for two lines can be solved by testing all $t = d_a$ in the case that $\sum_{a \in \mathcal{A}_{trans(l,k)}} w_a \geq \sum_{a \in \mathcal{A}_{trans(k,l)}} w_a$ and all $t = T - d_a$ otherwise. The same holds for problem (5) with two line clusters where we have to test all $t = \bar{d}_a$ or all $t = T - \bar{d}_a$. In both cases we have a finite candidate set of possible solutions to be checked. Using the Lemma 2 we can even derive a finite candidate set for any instance of (r-PESP). To this end, we define the *line graph*

$$G_{\mathcal{L}} = (\mathcal{L}, E_{\mathcal{L}})$$

as the graph with nodes corresponding to the lines $\mathcal{L}$ and undirected edges

$$E_{\mathcal{L}} := \{\{k, l\} \subseteq \mathcal{L} : \mathcal{A}_{trans}(k, l) \cup \mathcal{A}_{trans}(l, k) \neq \emptyset\}.$$

▶ **Theorem 3.** *There exists an optimal solution $\pi \in \{0, 1, \ldots, T-1\}^{|\mathcal{L}|}$ to r-PESP and a spanning tree S in the line graph $G_{\mathcal{L}}$ such that for every edge $e = \{k, l\} \in S$ there exists some $a \in \mathcal{A}_{trans}(k, l) \cup \mathcal{A}_{trans}(l, k)$ with $\pi_l - \pi_k = [d_a]_T$.*

Proof. We give a sketch of the proof here, its details can be found in the appendix.

In the proof we start with some timetable π. We determine the set $E(\pi)$ of all edges $\{k, l\}$ in the line graph which satisfy the condition of the theorem. If the set of these edges does not contain a spanning tree, we determine a largest connected component $\mathcal{L}_1$ in $(\mathcal{L}, E(\pi))$ and adjust the timetable optimally between the two line clusters $\mathcal{L}_1$ and $\mathcal{L} \setminus \mathcal{L}_1$. We receive a new timetable $\tilde{\pi}$. We then show that the resulting graph $(\mathcal{L}, E(\tilde{\pi}))$ w.r.t the new timetable $\tilde{\pi}$ has a strictly larger connected component. We can repeat this procedure until we find a timetable π^* such that $E(\pi^*)$ contains a spanning tree. ◀

The result shows that for every optimal solution there exists a spanning tree for which the tension $x_{kl} := \pi_l - \pi_k$ of its (directed) edge $\{k, l\}$ comes from a finite set $\{[d_a]_T : a \in \mathcal{A}_{trans}(k, l) \cup \mathcal{A}_{trans}(l, k)\}$ of values. We hence can enumerate over all trees and all such tensions to find an optimal timetable which will be formulated as Algorithm 1 in the next section.

We remark that the structure of the line graph $G_{\mathcal{L}} = (\mathcal{L}, E_{\mathcal{L}})$ may also be exploited for decomposing an r-PESP instance into two smaller instances in the following case.

▶ **Lemma 4.** *Let $\{\bar{k}, \bar{l}\} \in E_{\mathcal{L}}$ be a bridge of the line graph $G_{\mathcal{L}}$, i.e., an edge such that $(\mathcal{L}, E_{\mathcal{L}} \setminus \{e\})$ decomposes into two components $G_{\mathcal{L}_1} = (\mathcal{L}_1, E_{\mathcal{L}_1})$ and $G_{\mathcal{L}_2} = (\mathcal{L}_2, E_{\mathcal{L}_2})$. Let π^1 be an optimal solution to r-PESP on $\mathcal{L}_1$ and π^2 be an optimal solution to r-PESP on $\mathcal{L}_2$. Let furthermore t^* be the optimal adjustment between the two line clusters $\mathcal{L}_1$ and $\mathcal{L}_2$. Then*

$$\pi(\mathcal{L}_1, t^*)_l := \begin{cases} \pi^1_l + t^* & \text{if } l \in \mathcal{L}_1 \\ \pi^2_l & \text{if } l \in \mathcal{L}_2 \end{cases}$$

is an optimal solution to r-PESP on $\mathcal{L}$.

Proof. For the case of a bridge $\{\bar{k},\bar{l}\}$ with $\bar{k} \in \mathcal{L}_1, \bar{l} \in \mathcal{L}_2$ the objective function (5) for the adjustment of the two line clusters $\mathcal{L}_1$ and $\mathcal{L}_2$ with timetables π^1 and π^2 simplifies to

$$\begin{aligned} g(t) &= const + \sum_{\substack{a\in\mathcal{A}_{trans}(k,l)\\ k\in\mathcal{L}_1, l\in\mathcal{L}_2}} w_a[d_a + \pi^2_l - \pi^1_k - t]_T + \sum_{\substack{a\in\mathcal{A}_{trans}(k,l)\\ k\in\mathcal{L}_2, l\in\mathcal{L}_1}} w_a[d_a + \pi^2_l - \pi^1_k + t]_T \\ &= const + \sum_{a\in\mathcal{A}_{trans}(\bar{k},\bar{l})} w_a[d_a + \pi^2_{\bar{l}} - \pi^1_{\bar{k}} - t]_T + \sum_{a\in\mathcal{A}_{trans}(\bar{k},\bar{l})} w_a[d_a + \pi^2_{\bar{l}} - \pi^1_{\bar{k}} + t]_T \end{aligned}$$

Now consider any timetable π^*. We compare $f(\pi^*)$ with $f(\pi(\mathcal{L}_1, t^*))$:

$$\begin{aligned} f(\pi^*) &= \sum_{k,l\in\mathcal{L}_1} f_{k,l}(\pi^*) + \sum_{k,l\in\mathcal{L}_2} f_{k,l}(\pi^*) + f_{\bar{k},\bar{l}}(\pi^*) + f_{\bar{l},\bar{k}}(\pi^*) \\ f(\pi(\mathcal{L}_1, t^*)) &= \sum_{k,l\in\mathcal{L}_1} f_{k,l}(\pi^1 + t^*) + \sum_{k,l\in\mathcal{L}_2} f_{k,l}(\pi^2) + f_{\bar{k},\bar{l}}(\pi(\mathcal{L}_1, t^*)) + f_{\bar{l},\bar{k}}(\pi(\mathcal{L}_1, t^*)) \end{aligned}$$

For the first two terms we receive due to the optimality of π^1 and π^2 directly that

$$\sum_{k,l\in\mathcal{L}_1} f_{k,l}(\pi^1 + t^*) = \sum_{k,l\in\mathcal{L}_1} f_{k,l}(\pi^1) \leq \sum_{k,l\in\mathcal{L}_1} f_{k,l}(\pi^*) \quad \text{and} \quad \sum_{k,l\in\mathcal{L}_2} f_{k,l}(\pi^2) \leq \sum_{k,l\in\mathcal{L}_2} f_{k,l}(\pi^*).$$

For the third term we know that

$$\begin{aligned} & f_{\bar{k},\bar{l}}(\pi(\mathcal{L}_1, t^*)) + f_{\bar{l},\bar{k}}(\pi(\mathcal{L}_1, t^*)) \\ = & \sum_{a\in\mathcal{A}_{trans}(\bar{k},\bar{l})} w_a[d_a + \pi^2_{\bar{l}} - \pi^1_{\bar{k}} - t^*]_T \;\; + \sum_{a\in\mathcal{A}_{trans}(\bar{l},\bar{k})} w_a[d_a + \pi^1_{\bar{k}} - \pi^2_{\bar{l}} + t^*]_T \\ \leq & \sum_{a\in\mathcal{A}_{trans}(\bar{k},\bar{l})} w_a[d_a + \pi^2_{\bar{l}} - \pi^1_{\bar{k}} - t]_T \;\; + \sum_{a\in\mathcal{A}_{trans}(\bar{l},\bar{k})} w_a[d_a + \pi^1_{\bar{k}} - \pi^2_{\bar{l}} + t]_T \end{aligned}$$

for all $t \in \{0, \ldots, T-1\}$ since t^* is a minimizer of $g(t)$. In particular, this holds for $t := [\pi^2_{\bar{l}} - \pi^1_{\bar{k}} - \pi^*_{\bar{l}} + \pi^*_{\bar{k}}]_T$. Plugging this in, we receive

$$\begin{aligned} & f_{\bar{k},\bar{l}}(\pi(\mathcal{L}_1, t^*)) + f_{\bar{l},\bar{k}}(\pi(\mathcal{L}_1, t^*)) \\ \leq & \sum_{a\in\mathcal{A}_{trans}(\bar{k},\bar{l})} w_a[d_a + \pi^*_{\bar{l}} - \pi^*_{\bar{k}}]_T + \sum_{a\in\mathcal{A}_{trans}(\bar{l},\bar{k})} w_a[d_a + \pi^*_{\bar{k}} - \pi^*_{\bar{l}}]_T \\ = & f_{\bar{k},\bar{l}}(\pi^*) + f_{\bar{l},\bar{k}}(\pi^*), \end{aligned}$$

which finally shows that $f(\pi(\mathcal{L}_1, t^*) \leq f(\pi^*)$. ◀

3 Algorithms for r-PESP

3.1 An exact approach

The naive approach to solve r-PESP would be to enumerate brute-force and evaluate all possible $T^{|\mathcal{L}|-1}$ timetables of the reduced formulation r-PESP in $O(T^{|\mathcal{L}|-1} \cdot |\mathcal{A}_{trans}|)$. The result of Theorem 3 provides a finite set of values for the tensions on the edges of a specific spanning tree. Recall (e.g., from [11]) that fixing the tensions on a spanning tree (with directed edges) already determines the timetable π: It can be found by setting, e.g, $\pi_1 := 0$ and then iteratively choosing a neighbor i of a node j with already assigned time π_j and setting $\pi_i = [\pi_j + x_{1i}]_T$ if $(1,i) \in E$ and $\pi_i = [\pi_j - x_{1i}]_T$ if $(i,1) \in E)$ until all events have a time assigned. This approach works since a tree does not contain a cycle. We use this for proposing the following new exact approach for solving r-PESP:

Algorithm 1: Exact approach for finding an optimal solution to r-PESP

1. For every spanning tree S of the line graph $G_{\mathcal{L}}$ with edges E_S find an optimal timetable π_S for S by
   a. fixing an (arbitrary) direction of every edge $\{k, l\}$ of the tree S
   b. computing the corresponding timetable for all combinations of possible tensions values on the directed edges (k, l)
      - $\{d_a : a \in \mathcal{A}_{trans}(k, l)\}$ if $\sum_{a \in \mathcal{A}_{trans}(k,l)} w_a > \sum_{a \in \mathcal{A}_{trans}(l,k)} w_a$
      - $\{T - d_a : a \in \mathcal{A}_{trans}(l, k)\}$ if $\sum_{a \in \mathcal{A}_{trans}(k,l)} w_a \leq \sum_{a \in \mathcal{A}_{trans}(l,k)} w_a$.
2. Choose π as minimizer of $\min\{f(\pi_s) : S$ is a spanning tree of $G_{\mathcal{L}}\}$.

Using Cayley's formula saying that the number of spanning trees in a complete graph with n nodes is n^{n-2}, and that evaluating a timetable is of order $O(\mathcal{A}_{trans})$ it turns out that the complexity if Algorithm 1 is $O(|\mathcal{L}|^{|\mathcal{L}|-2} \cdot \eta^{|\mathcal{L}|-1} \cdot |\mathcal{A}_{trans}|)$ in a complete line graph $G_{\mathcal{L}}$ with η transfers between any pair of lines, i.e. $\eta = |\mathcal{A}_{trans}(k, l)|$ for all $k, l \in \mathcal{L}$. Note that for this time complexity we make use of Lemma 2, namely that we only need to evaluate all $d_a, a \in \mathcal{A}_{trans}(k, l)$ or $T - d_a, a \in \mathcal{A}_{trans}(l, k)$.

Even in this worst case we end up with a smaller time complexity than the naive brute-force approach if $\eta|\mathcal{L}| \leq T$ which will be the case in small to medium-size metro systems, assuming a period of $T = 60$ minutes. In practice, the line graph is usually not a complete graph, and the number of possible transfers η from a line l to another line k is usually small (often even zero) such that the complexity can be significantly reduced.

3.2 A heuristic based on matching

The idea of the matching heuristic is taken from [10] where a similar approach was used for aperiodic timetabling based on given vehicle routes. In every iteration we use a partition of the set of lines into line clusters $\mathcal{C} = \{\mathcal{L}_1, \ldots, \mathcal{L}_k\}$ with $\mathcal{L} = \mathcal{L}_1 \cup \ldots \cup \mathcal{L}_k$ and the $\mathcal{L}_p$ are pairwise disjoint. In the first step each line cluster consists of one single line only. In every iteration, the line clusters are matched pairwise. For every pair of clusters being matched one looks for the optimal adjustment of them by solving the optimization problem (5).

Algorithm 2: Matching-based heuristic for finding a solution to r-PESP

1. Initialization: Define the initial cluster graph $G_{\mathcal{C}} = (\mathcal{C}, E_{\mathcal{C}})$ as the line graph graph: $\mathcal{C} = \mathcal{L}$ (each line makes up one cluster), and $E_{\mathcal{C}} := E_{\mathcal{L}}$, i.e., two such clusters are connected if a transfer between their lines is possible.
   For every line $l \in \mathcal{L}$, define $\pi_l = 0$.
2. While $|\mathcal{C}| > 1$ do
   a. For every edge $\{\mathcal{L}_1, \mathcal{L}_2\} \in E_{\mathcal{C}}$ determine $eval(\mathcal{L}_1, \mathcal{L}_2)$ as in (10).
   b. Determine a matching $M \subseteq E_{\mathcal{C}}$ with maximal weight in $G_{\mathcal{C}}$.
   c. For every edge $\{\mathcal{L}_1, \mathcal{L}_2\} \in M$ do
      i. Find an optimal adjustment of the timetable of the two line clusters $\mathcal{L}_1$ and $\mathcal{L}_2$ (using the result of Lemma 2).
      ii. Merge the nodes $\mathcal{L}_1$ and $\mathcal{L}_2$ to one node $\mathcal{L}_1 \cup \mathcal{L}_2$ in $G_{\mathcal{C}}$.

The algorithm runs in polynomial time. The main question is how to evaluate two line clusters $\mathcal{L}_1$ and $\mathcal{L}_2$. As in the case of two clusters, we look at all transfers a between a line $k \in \mathcal{L}_1$ and another line $l \in \mathcal{L}_2$ and vice versa. If a timetable π is already given, the evaluation of the timetable $\pi(\mathcal{L}_1, t)$ (as in (4)) is done by computing

$$g_{\mathcal{L}_1,\mathcal{L}_2}(t) := + \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_1, l \in \mathcal{L}_2}} w_a [\bar{d}_a - t]_T + \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_2, l \in \mathcal{L}_1}} w_a [\bar{d}_a + t]_T$$

with (as usual) $\bar{d}_a := d_a + \pi_l - \pi_k$. As evaluation functions we tested

$$\begin{aligned} best(\mathcal{L}_1, \mathcal{L}_2) &:= \min_{t \in \{1,\dots,T-1\}} g_{\mathcal{L}_1,\mathcal{L}_2}(t) && (7) \\ worst(\mathcal{L}_1, \mathcal{L}_2) &:= \max_{t \in \{1,\dots,T-1\}} g_{\mathcal{L}_1,\mathcal{L}_2}(t) && (8) \\ span(\mathcal{L}_1, \mathcal{L}_2) &:= worst(\mathcal{L}_1, \mathcal{L}_2) - best(\mathcal{L}_1, \mathcal{L}_2) && (9) \\ expected(\mathcal{L}_1, \mathcal{L}_2) &:= \frac{1}{T} \left(\sum_{t=0}^{T-1} g_{\mathcal{L}_1,\mathcal{L}_2}(t) \right) - best(\mathcal{L}_1, \mathcal{L}_2) && (10) \end{aligned}$$

Note that all these evaluation functions are symmetric in $\mathcal{L}_1$ and $\mathcal{L}_2$, i.e., it does not matter if we look at $g_{\mathcal{L}_1,\mathcal{L}_2}(t)$ or at $g_{\mathcal{L}_2,\mathcal{L}_1}(t)$ when determining the values of (7)-(10).

▶ **Lemma 5.** *All the evaluation functions (7) - (10) are symmetric, i.e.,*

$$\begin{aligned} best(\mathcal{L}_1, \mathcal{L}_2) = best(\mathcal{L}_2, \mathcal{L}_1), \qquad & worst(\mathcal{L}_1, \mathcal{L}_2) = worst(\mathcal{L}_2, \mathcal{L}_1), \\ span(\mathcal{L}_1, \mathcal{L}_2) = span(\mathcal{L}_2, \mathcal{L}_1), \qquad & expected(\mathcal{L}_1, \mathcal{L}_2) = expected(\mathcal{L}_2, \mathcal{L}_1). \end{aligned}$$

Proof. As in (6) we can easily verify that $g_{\mathcal{L}_1,\mathcal{L}_2}(t) = g_{\mathcal{L}_2,\mathcal{L}_1}(T-t)$. Using furthermore that $g_{\mathcal{L}_2,\mathcal{L}_1}(0) = g_{\mathcal{L}_2,\mathcal{L}_1}(T)$, we receive that

$$\{g_{\mathcal{L}_1,\mathcal{L}_2}(t) : t = 0, \dots, T-1\} = \{g_{\mathcal{L}_2,\mathcal{L}_1}(t) : t = 0, \dots, T-1\},$$

hence the result follows from the fact that

$$\begin{aligned} \min_{t \in \{1,\dots,T-1\}} g_{\mathcal{L}_1,\mathcal{L}_2}(t) &= \min_{t \in \{1,\dots,T-1\}} g_{\mathcal{L}_2,\mathcal{L}_1}(t), \\ \max_{t \in \{1,\dots,T-1\}} g_{\mathcal{L}_1,\mathcal{L}_2}(t) &= \max_{t \in \{1,\dots,T-1\}} g_{\mathcal{L}_2,\mathcal{L}_1}(t), \\ \sum_{t=0}^{T-1} g_{\mathcal{L}_1,\mathcal{L}_2}(t) &= \sum_{t=0}^{T-1} g_{\mathcal{L}_2,\mathcal{L}_1}(t). \end{aligned}$$

◀

The different evaluation functions follow different strategies. *best* (7) matches the lines first which are most expensive to get adjusted even in the best case. *worst* (8) matches the lines which could make the objective value really bad later. *span* (9) and *expected* (10) consider how much the objective value for adjusting two line clusters can change between the best and the worst case, or between the expected and the best case. If the change is rather low there is no need to match such a pair.

Our pre-evaluation show that *span* and *expected* perform better than *best* and *worst* with *expected* providing the overall best results. We hence used *expected* (10) in Algorithm 2.

3.3 A hybrid algorithm

We can combine the exact and the matching approach by starting with the matching approach and changing to the exact approach when the complexity for solving the instance with an exact approach gets small enough. In case that brute-force is used as exact approach, we check the size of $T^{|\mathcal{C}|}$ where $\mathcal{C}$ contains the remaining clusters. When using Algorithm 1 as exact approach the decision is based on the number of spanning trees in the remaining graph.

Algorithm 3: Hybrid heuristic for finding a solution to r-PESP

1. Initialization: Define the initial cluster graph $G_{\mathcal{C}} = (\mathcal{C}, E_{\mathcal{C}})$ as the line graph graph: $\mathcal{C} = \mathcal{L}$ (each line makes up one cluster), and $E_{\mathcal{C}} := E_{\mathcal{L}}$, i.e., two such clusters are connected if a transfer between their lines is possible.
   For every line $l \in \mathcal{L}$, define $\pi_l = 0$.
2. While Complexity is too large do
   a. For every edge $(\mathcal{L}_1, \mathcal{L}_2) \in E_{\mathcal{C}}$ determine $eval(\mathcal{L}_1, \mathcal{L}_2)$ as in (10).
   b. Determine a matching $M \subseteq E_{\mathcal{C}}$ with maximal weight in $G_{\mathcal{C}}$.
   c. For every edge $(\mathcal{L}_1, \mathcal{L}_2) \in M$ do
      i. Find an optimal adjustment of the timetable of the two line clusters $\mathcal{L}_1$ and $\mathcal{L}_2$ as in Lemma 2.
      ii. Merge the nodes $\mathcal{L}_1$ and $\mathcal{L}_2$ to one node $\mathcal{L}_1 \cup \mathcal{L}_2$ in $G_{\mathcal{C}}$.
3. Solve the remaining instance exactly by using Algorithm 1 or another exact procedure.

Our experiments show that the runtime of the exact approaches is still too large for more than five lines; hence this is approximately the size of $\mathcal{C}$ when we switch from Algorithm 2 to an exact approach.

4 Experimental results

For our experiments we used data from the LinTim library [2, 5]. Besides a toy example this includes close-to real world data from the metro network of Athens, the German high-speed train network with different line concepts, and the bus network of the local bus company in Göttingen. The characteristics of the data used are summarized in Table 1. Note that all of these instances have no restriction on the upper bounds of transfer activities, so they satisfy assumption (1). On the other hand, none of the instances fixes the waiting or driving times of the activities to their lower bounds (2) as we do in r-PESP. It will be observed that even with this variable fixing the resulting outcomes of Algorithm 2 are competitive.

In our first evaluation we tested Algorithm 3 (the hybrid strategy) with three different settings: We either changed to an exact approach and took the naive brute-force enumeration or Algorithm 1, or we did not use any exact approach but only performed Algorithm 2. The results are shown in Table 2. We see that in all but one instance Algorithm 1 is faster than brute-force while the clear winner in runtime is (as expected) Algorithm 2, i.e., the polynomial matching heuristic. We also see that there is nearly no benefit in terms of the objective value solving the reduced final instances exactly instead of just continuing Algorithm 2. Note that the objective function values between using brute-force or Algorithm 1 as exact approach differ (although both alternatives are exact approaches) since the point *when* we switch to the exact approach depends on the algorithm chosen as explained at the beginning of Section 3.3.

Table 1 The characteristics of the test instances. V denotes the set of stations, $\mathcal{L}$ the set of lines, $\mathcal{E}$ the set of all arrival and departure events, and $\mathcal{A}_{trans}$ the set of transfer activities.

Name	$\lvert V\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{A}_{trans}\rvert$
toy	8	6	64	8
athens	51	20	592	115
bahn-eq-f	250	53	3444	1761
bahn-01	250	65	4184	4370
bahn-02	280	80	5048	3397
bahn-04	319	115	6368	7986
goevb	257	76	3044	9029

Table 2 Different versions of Algorithm 3.

	Brute-force in Step 3		Algorithm 1 in Step 3		No exact approach	
Instance	objective	runtime	objective	runtime	objective	runtime
toy	22094	10s	22094	$<$1s	22094	$<$1s
athens	12274246	30min	12274246	1s	12725934	1s
bahn-eq-f	66473861	1min	66473861	70min	66462971	1s
bahn-01	541120106	29min	540759985	20s	540759985	2s
bahn-02	675040353	35min	675206688	7min	675206688	2s
bahn-04	742637615	1min	742772838	1min	742637615	3s
goevb	20147281	18min	20191871	43s	20191871	1s

We finally compared Algorithm 2 to another procedure for periodic timetabling, namely to the modulo simplex ([12, 3]) in its implementation within LinTim [2, 5]. We compared the result of the matching-based heuristic (Algorithm 2) directly to the result of the modulo simplex, but also used it as a starting solution to check if the modulo simplex is able to further improve it. The runtime of the modulo Simplex was bounded to 60 minutes.

Note that in our instances slack times of driving and waiting activities are allowed, i.e., assumption (2) is not satisfied. This means that Algorithm 2 can only provide a heuristic solution also from this point of view. However, as we see in Table 3, the objective function values obtained by Algorithm 2 are highly competitive. The objective function values obtained by Algorithm 2 were surprisingly good, in one case even better than the result of the modulo simplex. But the main advantage of Algorithm 2 is its very fast runtime (which is also shown in the table). It furthermore turns out that the modulo simplex is able to further improve the solution obtained by Algorithm 2 in all cases, and that it cannot predicted which starting solution leads to the overall best solution after performance of the modulo simplex in the end.

5 Extension and conclusion

We presented a new formulation of the PESP in public transportation networks which is based on the characteristics of instances from timetabling. We show that this formulation can be used to derive a finite candidate set which is smaller than enumerating all possibilities in a brute-force approach. We also used the formulation to derive a matching-based heuristic for the PESP. Our experiments show promising results: the heuristic is competitive with

Table 3 Comparison with the Modulo Simplex.

	Algorithm 2		Algorithm 2 + ModSim		ModSim	
Instance	objective	runtime	objective	runtime	objective	runtime
toy	22094	< 1s	18236	< 1s	18236	< 1s
athens	12725934	1s	10215458	4s	10215458	30s
bahn-eq-f	66462971	1s	65430934	5min	65388991	13min
bahn-01	540759985	2s	536208754	1h	537965995	1h
bahn-02	675206688	2s	663179698	1h	668819275	1h
bahn-04	742637615	3s	737959364	1h	746841914	1h
goevb	20191871	1s	19691541	11min	18984122	19min

solutions obtained by the modulo simplex but with a runtime only in seconds. We currently investigate a heuristic in which the starting times π_l of the lines are fixed one after another in a Greedy manner (as proposed in [13]), in particular which of the evaluation functions (7)-(10) performs best in such an approach for choosing the sequence in which the lines are processed.

In our study we neglected headway constraints. However, they can be incorporated by adding feasibility constraints also in r-PESP meaning that constraints on t have to be taken into account when adjusting two lines or two line clusters. The implication of headway constraints and the performance of the matching-based approach in this case are subject of future research. Another interesting point is the further exploitation of Theorem 3. Since r-PESP is a special case of a PESP on the line graph $G_{\mathcal{L}} = (\mathcal{L}, E_{\mathcal{L}})$, the modulo simplex can directly applied to the reduced formulation. Using the special properties of the line graph $G_{\mathcal{L}}$ together with the finite candidate set on every tree is likely to yield further improvements for the modulo simplex. This is another point which is interesting to be studied in the future.

Finally, since the set of lines $\mathcal{L}$ is used explicitly in Algorithm 2, this seems to be a promising approach also for solving the integrated line-planning and timetabling problem in which a line plan and a timetable are optimized simultaneously.

References

1 M. Goerigk. Exact and heuristic approaches to the robust periodic event scheduling problem. *Public Transport*, 7(1):101–119, 2015.

2 M. Goerigk, M. Schachtebeck, and A. Schöbel. Evaluating line concepts using travel times and robustness: Simulations with the lintim toolbox. *Public Transport*, 5(3), 2013.

3 M. Goerigk and A. Schöbel. Improving the modulo simplex algorithm for large-scale periodic timetabling. *Computers and Operations Research*, 40(5):1363–1370, 2013.

4 P. Großmann, S. Hölldobler, N. Manthey, K. Nachtigall, J. Opitz, and P. Steinke. Solving periodic event scheduling problems with sat. In H. Jiang, W. Ding, M. Ali, and X. Wu, editors, *Advanced Research in Applied Artificial Intelligence*, volume 7345, pages 166–175. Springer, 2012.

5 J. Harbering, A. Schiewe, and A. Schöbel. LinTim - Integrated Optimization in Public Transportation. Homepage. see http://lintim.math.uni-goettingen.de/.

6 L. Kroon, G. Maróti, M. R. Helmrich, M. Vromans, and R. Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, 42(6):553 – 570, 2008.

7 L.G. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maroti, A. Shrijver, A. Steenbeek, and R. Ybema. The new Dutch timetable: The OR Revolution. *Interfaces*, 39:6–17, 2009.
8 C. Liebchen. *Periodic Timetable Optimization in Public Transport*. dissertation.de – Verlag im Internet, Berlin, 2006.
9 C. Liebchen. The first optimized railway timetable in practice. *Transportation Science*, 42(4):420–435, 2008.
10 M. Michaelis and A. Schöbel. Integrating line planning, timetabling, and vehicle scheduling: A customer-oriented approach. *Public Transport*, 1(3):211–232, 2009.
11 K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables*. PhD thesis, University of Hildesheim, 1998.
12 K. Nachtigall and J. Opitz. Solving periodic timetable optimisation problems by modulo simplex calculations. In *Proc. ATMOS*, 2008.
13 K. Nachtigall and S. Voget. A genetic approach to periodic railway synchronization. *Computers Ops. Res.*, 23(5):453–463, 1996.
14 M. A. Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research*, 30B:455–464, 1996.
15 J. Pätzold. Periodic timetabling with fixed driving and waiting times. Master's thesis, Fakultät für Mathematik und Informatik, Georg August University Göttingen, 2016. (in German).
16 L. Peeters and L. Kroon. A cycle based optimization model for the cyclic railway timetabling problem. In S. Voß and J. Daduna, editors, *Computer-Aided Transit Scheduling*, volume 505 of *Lecture Notes in Economics and Mathematical systems*, pages 275–296. Springer, 2001.
17 P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematic*, 2:550–581, 1989.

A Proof of Theorem 3

Theorem 3. *There exists an optimal solution $\pi \in \{0, 1, \ldots, T-1\}^{|\mathcal{L}|}$ to r-PESP and a spanning tree S in the line graph $G_{\mathcal{L}}$ such that for every edge $e = \{k, l\} \in S$ there exists some $a \in \mathcal{A}_{trans}(k, l) \cup \mathcal{A}_{trans}(l, k)$ with $\pi_l - \pi_k + d_a = 0$.*

Proof. Let $\pi^* \in \{0, \ldots, T-1\}^{|\mathcal{L}|}$ be a given timetable. Without loss of generality assume that $d_a \in \{0, \ldots, T-1\}$, otherwise just use $[d_a]_T$ instead of d_a. Define

$$E(\pi) := \{e = \{k, l\} \in E_{\mathcal{L}} : \pi_l - \pi_k - d_a = 0 \text{ for some } a \in \mathcal{A}_{trans}(k, l) \cup \mathcal{A}_{trans}(l, k)\}$$

and consider the largest connected component of $\mathcal{L}_1$ of $(\mathcal{L}, E(\pi))$ (which may consist of one node $l \in \mathcal{L}$ only). Note that $E(\pi)$ does not contain any edge between $\mathcal{L}_1$ and $\mathcal{L}_2$, i.e.,

$$E(\pi) \subseteq \{\{k, l\} \in E_{\mathcal{L}} : k, l \in \mathcal{L}_1\} \cup \{\{k, l\} \in E_{\mathcal{L}} : k, l \in \mathcal{L}_2\}. \tag{11}$$

If $\mathcal{L}_1 = \mathcal{L}$, the line graph $G_{\mathcal{L}}$ contains a tree which satisfies the condition of the theorem and we are done. Otherwise we construct a tree and a timetable which is at least as good as π and satisfies the condition.

To this end, consider again $(\mathcal{L}, E(\pi))$ with its largest connected component $\mathcal{L}_1$, and let $\mathcal{L}_2 := \mathcal{L} \setminus \mathcal{L}_1$. Let S be a spanning tree of $\mathcal{L}_1$ using only edges of $E(\pi)$. We now construct a new timetable

$$\tilde{\pi} := \pi(\mathcal{L}_1, t) = \begin{cases} \pi_l & \text{if } l \in \mathcal{L}_2 \\ \pi_l + t & \text{if } l \in \mathcal{L}_1 \end{cases}$$

(see also (4)) which optimally adjusts the two clusters $\mathcal{L}_1$ and $\mathcal{L}_2$. According to (5) we hence have to find the minimum of $g(t)$ with

$$g(t) := \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_1, l \in \mathcal{L}_2}} w_a[\bar{d}_a - t]_T + \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_2, l \in \mathcal{L}_1}} w_a[\bar{d}_a + t]_T$$

and $\bar{d}_a := d_a + \pi_l - \pi_k$ as in (5). To this problem we apply Lemma 2 with $\mathcal{A}_1 := \{a \in \mathcal{A}_{trans}(k,l) : k \in \mathcal{L}_1, l \in \mathcal{L}_2\}$ and $\mathcal{A}_2 := \{a \in \mathcal{A}_{trans}(k,l) : k \in \mathcal{L}_2, l \in \mathcal{L}_1\}$. We distinguish two cases:

Case 1 There exists a minimum t^* of $g(t)$ with $t^* = \bar{d}_a$ for some $a \in \mathcal{A}_{trans}(k,l)$ with $k \in \mathcal{L}_1, l \in \mathcal{L}_2$ (if $\sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_1, l \in \mathcal{L}_2}} w_a \geq \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_2, l \in \mathcal{L}_1}} w_a$):
For the resulting timetable $\tilde{\pi} := \pi(\mathcal{L}_1, t^*)$ we compute

$$\tilde{\pi}_l - \tilde{\pi}_k + d_a = \pi_l - (\pi_k + t^*) + d_a = \pi_l - \pi_k - \bar{d}_a + d_a = 0.$$

Case 2 There exists a minimum t^* of $g(t)$ with $t^* = T - \bar{d}_a$ for some $a \in \mathcal{A}_{trans}(k,l)$ with $k \in \mathcal{L}_2, l \in \mathcal{L}_1$ (if $\sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_1, l \in \mathcal{L}_2}} w_a < \sum_{\substack{a \in \mathcal{A}_{trans}(k,l) \\ k \in \mathcal{L}_2, l \in \mathcal{L}_1}} w_a$):
For the resulting timetable $\tilde{\pi} := \pi(\mathcal{L}_1, t^*)$ we again receive

$$\tilde{\pi}_l - \tilde{\pi}_k + d_a = \pi_l + t^* - \pi_k + d_a = \pi_l + T - \bar{d}_a - \pi_k + d_a = 0.$$

(Note that this is the same as $t^* = T - (T - \bar{d}_a) = \bar{d}_a$ for $\tilde{\pi} = \pi(\mathcal{L}_2, \bar{d}_a)$ according to (6).)
We now consider $E(\tilde{\pi}) = \{e = \{k,l\} \in E_{\mathcal{L}} : \tilde{\pi}_l - \tilde{\pi}_k - d_a = 0$ for some $a \in \mathcal{A}_{trans}(k,l) \cup \mathcal{A}_{trans}(l,k)\}$. Observe that

- for k,l both in $\mathcal{L}_1$ or for k,l both in $\mathcal{L}_2$ we have that $\tilde{\pi}_l - \tilde{\pi}_k - d_a = 0$ if and only if $\pi_l - \pi_k - d_a = 0$,
- for $k \in \mathcal{L}_1, l \in \mathcal{L}_2$ or vice versa, no edge $\{k,l\}$ is contained in $E(\pi)$ (see (11)), while we have just seen that the optimal adjustment of the two clusters $\mathcal{L}_1$ and $\mathcal{L}_2$ yields at least one transfer activity a in $\mathcal{A}_{trans}(k,l) \cup \mathcal{A}_{trans}(l,k)$ for some edge $\{k,l\}$ between $\mathcal{L}_1$ and $\mathcal{L}_2$ with $\tilde{\pi}_l - \tilde{\pi}_k + d_a = 0$. Hence $\{k,l\} \in E(\tilde{\pi}) \setminus E(\pi)$.

We conclude that the largest connected component $\mathcal{L}_1'$ of $E(\tilde{\pi})$ contains $\mathcal{L}_1$ and at least one additional node $l \in \mathcal{L}_2$. We may proceed with $\mathcal{L}_1 := \mathcal{L}_1'$ and $\pi := \tilde{\pi}$ and continue the whole procedure until $\mathcal{L}_1 = \mathcal{L}$. ◀