



# Introducing TimPassLib – A Library for Integrated Periodic Timetabling and Passenger Routing

Philine Schiewe<sup>1</sup> · Marc Goerigk<sup>2</sup> · Niels Lindner<sup>3,4</sup>

Received: 31 January 2023 / Accepted: 10 August 2023 / Published online: 29 August 2023  
© The Author(s) 2023

## Abstract

Classic models to derive a timetable for public transport often face a chicken-and-egg situation: A good timetable should offer passengers routes with small travel times, but the route choice of passengers depends on the timetable. While models that fix passenger routes were frequently considered in the literature, integrated models that simultaneously optimize timetables and passenger routes have seen increasing attention lately. This creates a growing need for a set of instances that allows to test and compare new algorithmic developments for the integrated problem. Our paper addresses this requirement by presenting TimPassLib, a new benchmark library of instances for integrated periodic timetabling and passenger routing.

**Keywords** Periodic timetabling · Optimization in public transport · Data sets · Benchmarking

## 1 Introduction

Improvements in public transport systems have the potential to better the life of millions. In the sustainable development goals of the United Nations Development Programme, goal 11 (“Sustainable Cities and Communities”) states the need to “provide access to safe, affordable, accessible and sustainable transport systems for all, improving road safety, notably by expanding public transport” [1].

---

This article is part of the Topical Collection on *Public Transport Optimization: From Theory to Practice*

✉ Philine Schiewe  
philine.schiewe@aalto.fi

<sup>1</sup> Department of Mathematics and Systems Analysis, Aalto University, Espoo, Finland

<sup>2</sup> Business Decisions and Data Science, University of Passau, Passau, Germany

<sup>3</sup> Institut für Mathematik, Freie Universität Berlin, Berlin, Germany

<sup>4</sup> Department Network Optimization, Zuse Institute Berlin, Berlin, Germany

Operations Research can help achieve these goals by making better use of the existing infrastructure.

One part of the process is to design timetables that allow passengers short travel times. Since the introduction of the periodic event scheduling problem (PESP) in the late 80s (see [2]), periodic timetables have been a major challenge to optimize. In this model, we assume that services are offered periodically (e.g., there is a train departure of a specific line from a station at 20 min past the hour throughout the day). Periodic timetabling models are surprisingly flexible in the aspects that can be included [3], and have been driving algorithmic development through the theoretical (see, e.g. [4, 5]) and computational challenge that they pose (see, e.g., [6–11]). Algorithmic development requires instances that allow the research community to compare performance results. The PESPlib benchmark set [12] has been offering such an opportunity in the case of pure periodic timetabling problems.

At an increasing rate, recent research has gone beyond the possibilities of PESP by looking at more integrated models for public transport [13–15]. This approach is particularly natural in the case of periodic timetabling. To find a good timetable, we need to know the routes that the passengers wish to use. But passenger routes depend on the timetable. In classic timetabling problems, we simply fix a passenger routing that is calculated in a heuristic way. This can potentially lead to solutions that are far from optimal, see [16, 17], even if we follow an iterative timetabling and passenger routing strategy, see for example the discussion in [18]. For this reason, integrated models need to be considered, that optimize both the timetable and the passenger routes simultaneously. In many recent publications, solution methods for integrated models have been discussed, see, e.g., [17, 19–24].

This creates a demand for new benchmark instances that allow researchers to compare the results that solution methods can achieve. The purpose of this paper is to address this demand by introducing a new, freely available benchmark set of instances for the integrated periodic timetabling and passenger routing: the TimPassLib [25].

The remainder of this paper is structured as follows. In Sect. 2, we give a formal definition for the integrated problem that we study. The problem instances that constitute the TimPassLib are described in Sect. 3. We conclude our work in Sect. 4 and described the data format in Appendix A.

## 2 Problem Definition

We first introduce the classic periodic timetabling problem where passenger routes are fixed, which is based on the periodic event scheduling problem [2]. We consider a set of events  $\mathcal{E}$ . For each event  $i \in \mathcal{E}$  we would like to schedule a time  $\pi_i$  in the discrete interval  $\{0, \dots, T - 1\}$ , where  $T$  denotes the period length. The schedule is supposed to be repeated every  $T$  time units. In timetabling, typical events are the arrival or the departure of a train at a station. Additionally, there is a set of activities  $\mathcal{A} \subseteq \mathcal{E} \times \mathcal{E}$  that connect events with each other. Such an activity may model a train driving from one station to the next, a train waiting at a station for passengers to embark or disembark, or a group of passengers changing from one train arrival to

the departure of another train. Another class of activities, which is not related to passengers, could model headway between trains or synchronization of lines running more frequently than once every  $T$  time units. Events and activities together result in the event-activity network  $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ . Each activity  $a \in \mathcal{A}$  has a lower bound  $\ell_a \in \mathbb{N}$  and upper bound  $u_a \in \mathbb{N}$  that reflect requirements on minimum and maximum durations of activities. We denote by  $y_a$  the slack of an activity, i.e., difference of an activity duration to its lower bound. Finally, we assume that some weight  $w_a$  is known for each activity, which represents the number of passengers who wish to use this activity.

The periodic timetabling problem can then be formulated as the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{a \in \mathcal{A}} w_a (y_a + \ell_a) \\ \text{s.t.} \quad & y_a = [\pi_j - \pi_i - \ell_a]_T \quad \forall a = (i, j) \in \mathcal{A}, \\ & 0 \leq y_a \leq u_a - \ell_a \quad \forall a \in \mathcal{A}, \\ & \pi_i \in \{0, \dots, T - 1\} \quad \forall i \in \mathcal{E}. \end{aligned}$$

Solution approaches based on mixed-integer programming typically focus on one set of variables; either the timetable variables  $\pi_i$  (node-based formulations) or the slack variables  $y_a$  (cycle-based formulations) [26]. The symbol  $[\cdot]_T$  denotes the modulo bracket  $[x]_T = \min\{x + zT : x + zT \geq 0, z \in \mathbb{Z}\}$  and is usually modeled by introducing additional integer variables.

In the *integrated* periodic timetabling and passenger routing problem, we choose to model the activity weights  $w_a$  in more detail. We assume that an origin–destination (OD) matrix is given, where every entry  $d_{st}$  denotes the number of passengers who wish to travel from origin station  $s \in V$  to destination station  $t \in V$ , where  $V$  denotes the set of all stations. Note that multiple events in the event-activity network can be assigned to the same station. Let  $P_{st}$  denote all simple paths in  $\mathcal{N}$  that connect an event corresponding to station  $s$  with an event corresponding to station  $t$ , and use exclusively drive, wait or change activities as described above. For each path  $p \in P_{st}$ , let  $f_p$  denote the fraction of passengers that travel from  $s$  to  $t$  along  $p$ . This means that for each OD pair  $(s, t)$  with  $d_{st} > 0$ , we require

$$\sum_{p \in P_{st}} f_p = 1.$$

The passenger weights  $w_a$  for each activity  $a \in \mathcal{A}$  are then determined as

$$w_a = \sum_{(s,t) \in V \times V : d_{st} > 0} \sum_{p \in P_{st} : a \in p} d_{st} f_p \quad \forall a \in \mathcal{A}.$$

From a passenger’s perspective, there are more criteria which connection to choose, apart from just the travel time. Another frequently considered criterion is the number of transfers along a journey [27]. This can be included in this model by using an additional penalty if passengers need to use a change activity, which represents the discomfort in comparison to a direct connection. Let  $c$  denote this penalty

value and let  $\mathcal{A}_{\text{change}} \subseteq \mathcal{A}$  denote the set of change activities in the event-activity network. The objective function is extended by the term

$$\sum_{a \in \mathcal{A}_{\text{change}}} cw_a$$

to include change penalties.

The integrated periodic timetabling and passenger routing problem is now to find both the timetable  $(\pi, y)$  as well as passenger routes and weights  $(f, w)$  simultaneously. The resulting model is thus non-linear because of the bilinear term in the objective. Observe that we can assume that variables  $f_p$  are binary, as there is always an optimal solution where passengers are routed along a single shortest path, since the model does not contain capacity constraints. Hence, the overall model for our problem can be summarized as follows [16, 18, 21, 28]:

$$\begin{aligned} \min \quad & \sum_{a \in \mathcal{A}} w_a(y_a + \ell_a) + \sum_{a \in \mathcal{A}_{\text{change}}} cw_a \\ \text{s.t.} \quad & y_a = [\pi_j - \pi_i - \ell_a]_T \quad \forall a = (i, j) \in \mathcal{A}, \\ & \sum_{p \in P_{st}} f_p = 1 \quad \forall (s, t) \in V \times V : d_{st} > 0, \\ & w_a = \sum_{\substack{(s,t) \in V \times V : \\ d_{st} > 0}} \sum_{\substack{p \in P_{st} \\ a \in p}} d_{st} f_p \quad \forall a \in \mathcal{A}, \\ & 0 \leq y_a \leq u_a - \ell_a \quad \forall a \in \mathcal{A}, \\ & \pi_i \in \{0, \dots, T - 1\} \quad \forall i \in \mathcal{E}, \\ & f_p \in \{0, 1\} \quad \forall (s, t) \in V \times V : d_{st} > 0, p \in P_{st}. \end{aligned} \tag{*}$$

### 3 Data Sets

All benchmark data sets presented in this paper are available at <https://timpasslib.net> [25]. The file format is described in Appendix A. The website collects the best known solutions and bounds for each data set. A potential new solution can be checked by means of a verification script written in Python, which also computes an optimal routing for a given timetable.

In the remainder of this section, we briefly describe the TimPassLib data sets. All instances are based on a set of lines, where we interpret a line as a sequence of stations without repetitions. Table 1 presents an overview of some key features of the instances. The columns contain the following information:

instance	The name of the instance. More information can be found in the following subsections.
stations	The number of stations.

lines	The number of operated lines. Note that a line can have a frequency higher than one.
OD pairs	The number of OD pairs with $d_{st} > 0$ .
OD total	The total number of passengers.
events	The number of events in the event-activity network.
activities total	The total number of activities in the event-activity network.
activities fixed	The number of activities in the event-activity network with $\ell_a = u_a$ .
activities free	The number of activities in the event-activity network with $u_a - \ell_a = T - 1$ .
activities restricted (restr.)	The number of activities in the event-activity network with $\ell_a < u_a < \ell_a + T - 1$ .
reference objective (ref. obj.)	The objective value w.r.t. (★) of the best timetable computed by the concurrent PESP solver by [7] within a wall time limit of one hour. The computations have been executed on an Intel Xeon CPU E3-1270 v6 running at 3.80 GHz with 32 GB RAM using Gurobi 9.5.2 [29] as MIP solver. The weights for the periodic timetabling problem have been obtained from a passenger routing according to lower bounds $\ell_a$ and change penalty $c$ . The aim of this reference objective is to provide initial solutions obtained with uniform computational power.
lower bound	A lower bound on the objective value of (★) obtained by routing all passengers on shortest paths w.r.t. lower activity bounds $\ell_a$ and change penalty $c$ .
gap	The gap per passenger between reference objective and lower bound: $\frac{\text{reference objective} - \text{lower bound}}{\text{OD total}}$ <p>This gap can be used to estimate the gains of solving the integrated problem.</p>

**Table 1** Overview of key features for the data sets

instance	OD				activities							lower bound	gap
	stations	lines	pairs	total	events	total	fixed	free	restr.	ref. obj.			
Hamburg	68	7	2030	9694166	508	823	449	284	90	141629305	139892927	0.18	
Schweiz_Fernverkehr	140	80	12082	1347686	2234	18467	1563	14787	2117	63273307	60084289	2.37	
toy	8	2	46	2622	32	44	2	16	26	21714	20194	0.58	
toy_2	8	6	46	2622	156	1088	112	868	108	19316	19114	0.08	
regional	34	8	330	325968	412	1520	176	958	386	2068934	1804642	0.81	
grid	25	8	567	2546	392	2382	176	1842	364	50692	47824	1.13	
long-distance	250	42	6106	385868	2800	14958	2716	12242	0	67697819	64906980	7.23	
metro	51	4	2385	63323	964	3240	708	2050	482	12021720	11978129	0.69	
Erding_NDP_S020	51	21	675	558164	1132	5300	320	3944	1036	12291729	12206083	0.15	
Erding_NDP_S021	51	21	675	558164	1132	5396	320	3944	1132	12307765	12206083	0.18	
Stuttgart	560	156	132439	39517136	21328	297094	22856	268606	5632	49009914897	45072189100	99.65	
R1L1	250	55	48842	3147382	3664	6381	642	2827	2912	542908145	522575407	6.46	
R1L2	250	54	48842	3147382	3668	6543	632	2983	2928	542381697	522212362	6.41	
R1L3	250	65	48842	3147382	4184	7025	752	2971	3302	543067240	522199838	6.63	
R1L4	250	71	48842	3147382	4760	8528	830	3910	3788	537879494	520799059	5.43	
R2L1	280	66	61110	3666720	4156	7356	814	3332	3210	681061389	650575045	8.31	
R2L2	280	66	61110	3666720	4204	7561	820	3489	3252	676836085	650293220	7.24	
R2L3	280	80	61110	3666720	5048	8285	970	3397	3918	677126860	649767761	7.46	
R2L4	280	116	61110	3666720	7660	13168	1496	5740	5932	667537183	647184195	5.55	
R3L1	296	73	68284	3878392	4516	9140	794	4770	3576	694086648	665804283	7.29	
R3L2	296	70	68284	3878392	4452	9249	774	4937	3538	694334373	665719574	7.38	
R3L3	296	95	68284	3878392	5724	11165	1038	5631	4496	691688857	665595680	6.73	
R3L4	296	120	68284	3878392	8180	15655	1478	7715	6462	682090300	662251432	5.12	
R4L1	319	86	77878	4183088	4932	10262	996	5502	3764	755138437	723276168	7.62	

**Table 1** (continued)

instance	OD					activities					gap	
	stations	lines	pairs	total	events	total	fixed	free	restr.	ref. obj.		lower bound
R4L2	319	89	77878	4183088	5048	10733	984	5863	3886	754453547	724254447	7.22
R4L3	319	115	77878	4183088	6368	13236	1240	7098	4898	751351849	722434044	6.91
R4L4	319	133	77878	4183088	8384	17753	1572	9635	6546	738792466	720103154	4.47

### 3.1 Hamburg

The instance *Hamburg* models the suburban commuter rail network of S-Bahn Hamburg. The infrastructure is mostly independent from other railways and has only few single-track sections. The network is operated with six lines, where one of the lines has two branches. The period length is 10, as all lines run every 10 min in the rush hour. Bounds for travel and dwell times are derived from the annual timetable for 2023. The passenger demand is based on data from trains with sensors for automatic passenger counts, which is publicly available on the open data portal of Deutsche Bahn AG [30]. We solve a linear program that fits this passenger flow to a gravity model derived from the total number of passengers boarding and alighting at each station.

### 3.2 Schweiz\_Fernverkehr

The instance *Schweiz\_Fernverkehr* is an excerpt of the ICE, TGV, InterCity and InterRegio trains within Switzerland and contains 80 lines. The period length is 120 min. The data is based on GTFS timetable data for 2023 and on station passenger counts, which are both publicly available on the open data platform of the Swiss federal office of transport [31]. We use a gravity model based on geographical distances to sample an OD matrix.

### 3.3 toy and toy\_2

The instances *toy* and *toy\_2* are based on a small artificial data set in the software library LinTim [32]. The line concepts are generated algorithmically and consist of two and six lines, respectively. The period length is 60. Note that the same instances are used in [17].

### 3.4 regional

The instance *regional* is based on the regional train network in Lower Saxony, Germany, and is available as part of the software library LinTim [32]. The line concept is generated algorithmically and consists of 8 lines. The period length is 60. Note that the same instance is used in [17].

### 3.5 grid

The instance *grid* is a benchmark data set, originally introduced in [33] and available as part of the open source data set [34] as well as the software library LinTim [32]. The line concept is generated algorithmically and consists of 8 lines. The period length is 60. Note that the same instance is used in [17].



### 3.6 long-distance

The instance *long-distance* is inspired by the long-distance train network in Germany and part of the developer version of software library LinTim [32]. The line concept is generated algorithmically and consists of 42 lines. The period length is 60. Note that the same instance is used in [17].

### 3.7 metro

The instance *metro* is based on the metro system in Athens, Greece, and is available as part of the software library LinTim [32]. The line concept consists of 4 lines. The period length is 150 and the timetable is planned in increments of 6 s. Note that the same instance is used in [17].

### 3.8 Erding\_NDP\_S020 and Erding\_NDP\_S021

The instances *Erding\_NDP\_S020* and *Erding\_NDP\_S021* are based on the transport supply in Erding, Germany, and are available as part of an open source data set [34]. The corresponding line concepts consist of 21 lines and the period length is 60.

### 3.9 Stuttgart

The instance *Stuttgart* is based on the transport supply in Stuttgart, Germany, and is available as part of an open source data set [34]. The corresponding line concept consists of 156 lines and the period length is 3600, i.e., the timetable is planned in seconds.

### 3.10 RxLy

This group consists of 16 instances, numbered from *R1L1* to *R4L4*. They are an extension of the 16 core instances of the PESPLib [12], which in turn was based on long-distance train network data in Germany from the software library LinTim [32]. The numbers of lines are in the range from 54 up to 133, with a period length of 60.

## 4 Outlook

Integrating passenger routing into periodic timetabling is a natural step to resolve the mutual interdependence of both problems. For more than ten years, the PESPLib library [12] has stimulated a variety of research on periodic timetabling with fixed routing. As of now, recent improvements are only marginal, disregard passenger flow, and therefore do not necessarily improve passenger comfort. With the initiation of TimPassLib, we hope to foster the investigation of periodic timetabling with passenger routing, to extend the scope of the PESPLib library to a solid base of benchmark instances for the integrated problem, and to contribute to the attractiveness of public transport with mathematical optimization methods. We are therefore

very much looking forward to submissions of new solutions or lower bounds as well as new benchmark instances. Furthermore, we think of extending TimPassLib with more optional features in the future, including vehicle capacities and different routing models.

## A. Data Format

In- and output data for the benchmark instances is given in `csv` files that are designed to be easily readable. Each file starts with a header that describes the content of the columns.

### A.1 Input File `Config.csv`

The file `Config.csv` contains additional information needed for the data set.

```
# config_key; value
ptn_name; toy
period_length; 60
ean_change_penalty; 5
```

The file contains a value for each following `config_key`:

<code>ptn_name</code>	name of the data set (string)
<code>period_length</code>	period length $T$ (integer)
<code>ean_change_penalty</code>	change penalty $c$ (integer)

### A.2 Input File `OD.csv`

The file `OD.csv` contains information about the OD pairs.

```
# origin; destination; customers
1; 2; 10
1; 3; 20
1; 4; 10
1; 5; 60
...
```

Here, `origin` gives the id of the start station  $s$  of the OD pair, `destination` the corresponding destination  $t$  and `customers` the number of passengers  $d_{st}$  who wish to travel from  $s$  to  $t$ . Note that all values are integers.

### A.3 Input File `Events.csv`

The file `Events.csv` contains information about the events  $\mathcal{E}$  as events represent arrival and departures of trips at stations.

```
# event_id; type; stop_id; line_id; line_direction; line_freq_repetition
1; "departure"; 2; 2; >; 1
2; "arrival"; 3; 2; >; 1
3; "departure"; 3; 2; >; 1
4; "arrival"; 6; 2; >; 1
...
```

<code>event_id</code>	id of the event (integer)
<code>type</code>	whether the event represents the arrival ("arrival") or departure ("departure") of the corresponding trip
<code>stop_id</code>	id of the corresponding station (integer)
<code>line_id</code>	id of the corresponding line (integer)
<code>line_direction</code>	whether the trip belongs to the line in forward (>) or backward (<) direction
<code>line_freq_repetition</code>	repetition of the line in the planning period to which the trip belongs (integer)

### A.4 Input File `Activities.csv`

The file `Activities.csv` contains information about the activities.

```
# activity_index; type; from_event; to_event; lower_bound; upper_bound
1; "drive"; 1; 2; 3; 4
2; "wait"; 2; 3; 1; 3
3; "drive"; 3; 4; 1; 3
4; "wait"; 4; 5; 1; 3
...
```

<code>activity_index:</code>	id of the activity (integer)
<code>type:</code>	type of the activity. Possible values are:

- "drive" represents a vehicle driving from one stop of a line to the next (can be used for routing passengers)
- "wait" represents a vehicle waiting at a stop (can be used for routing passengers)
- "change" represents a transfer between two lines at the same stop (can be used for routing passengers)

- "sync" used for synchronizing trips of the same line (cannot be used for routing passengers)
- "headway" represents headway constraint between vehicles using the same infrastructure (cannot be used for routing passengers)

from\_event: id of event  $i$  for activity  $(i, j)$  (integer)  
 to\_event: id of event  $j$  for activity  $(i, j)$  (integer)  
 lower\_bound: lower bound  $\ell_a$  of activity  $a$  (integer)  
 upper\_bound: upper bound  $u_a$  of activity  $a$  (integer)

### A.5 Solution File `Timetable.csv`

The solution file `Timetable.csv` should be given in the following format.

```

# event_id; time
1; 0
2; 3
3; 4
4; 5
...

```

event\_id id of the event (integer)  
 time time of the event ( $\{0, \dots, T - 1\}$ )

Since an optimal routing for a given timetable can be computed efficiently by solving a series of shortest path problems, passenger routing data is not required when submitting a solution.

**Acknowledgements** We want to thank Alexander Schiewe for his help in setting up the website <https://timpasslib.net>.

**Author Contributions** Philine Schiewe: Conceptualization, Methodology, Software, Investigation, Validation, Formal Analysis, Data Curation, Writing – Original Draft, Writing – Review & Editing; Marc Goerigk: Conceptualization, Methodology, Investigation, Data Curation, Writing – Original Draft; Niels Lindner: Conceptualization, Methodology, Software, Validation, Investigation, Data Curation, Writing – Review & Editing

**Funding** Open Access funding provided by Aalto University.

**Data Availability** The TimPassLib data set is publicly available at <https://timpasslib.net>.

### Declarations

**Ethical Approval** Not applicable.

**Competing Interests** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. United Nations Development Programme (2015) Sustainable Development Goals. <https://www.undp.org/sustainable-development-goals#sustainable-cities-and-communities>. Accessed 30 Jan 2023
2. Serafini P, Ukovich W (1989) A mathematical model for periodic scheduling problems. *SIAM J Discret Math* 2(4):550–581
3. Liebchen C, Möhring RH (2008) The modeling power of the periodic event scheduling problem: railway timetables—and beyond. In *Computer-aided Systems in Public Transport*, pages 117–150. Springer
4. Borndörfer R, Hoppmann H, Karbstein M, Lindner N (2020) Separation of cycle inequalities in periodic timetabling. *Discret Optim* 35:100552
5. Lindner N, Reisch J (2022) An analysis of the parameterized complexity of periodic timetabling. *J Sched* 25(2):157–176
6. Bortolotto E, Lindner N, Masing B (2022) Tropical neighbourhood search: A new heuristic for periodic timetabling. In: D'Emidio Mattia, Lindner Niels (eds) 22nd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2022). Schloss Dagstuhl - Leibniz-Zentrum für Informatik
7. Borndörfer R, Lindner N, Roth S (2020) A concurrent approach to the periodic event scheduling problem. *J Rail Transp Plan Manag* 15:100175
8. Goerigk M, Liebchen C (2017) An improved algorithm for the periodic timetabling problem. In Gianlorenzo D'Angelo and Twan Dollevoet, editors, 17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017), volume 59 of *OpenAccess Series in Informatics (OASICS)*, pages 12:1–12:14, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz–Zentrum fuer Informatik. ISSN: 2190-6807
9. Goerigk M, Schöbel A (2013) Improving the modulo simplex algorithm for large-scale periodic timetabling. *Comput Oper Res* 40(5):1363–1370
10. Lindner N, Liebchen C (2020) Determining all integer vertices of the PESP polytope by flipping arcs. In Huisman D, Zaroliagis CD, editors, 20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020), volume 85 of *OpenAccess Series in Informatics (OASICS)*, pages 5:1–5:18, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz–Zentrum für Informatik. ISSN: 2190-6807
11. Lindner N, Liebchen C (2022) Timetable merging for the periodic event scheduling problem. *EURO J Transp Logist* 11:100081
12. Goerigk M (2012) PESPLib – a benchmark library for periodic event scheduling. <http://num.math.uni-goettingen.de/~m.goerigk/pesplib/>. Accessed 26 Jan 2023
13. Schöbel A (2017) An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transp Res Part C Emerg Technol* 74:348–365
14. Schiewe P (2020) Integrated optimization in public transport planning, volume 160 of *optimization and its applications*. Springer, Cham. <https://doi.org/10.1007/978-3-030-46270-3>
15. Schiewe P, Schöbel A (2022) Integrated optimization of sequential processes: General analysis and application to public transport. *EURO J Transp Logist* 11:100073
16. Borndörfer R, Hoppmann H, Karbstein M (2017) Passenger routing for periodic timetable optimization. *Public Transport* 9:115–135
17. Schiewe P, Schöbel A (2020) Periodic timetabling with integrated routing: Toward applicable approaches. *Transp Sci* 54(6):1714–1731

18. Siebert M, Goerigk M (2013) An experimental comparison of periodic timetabling models. *Comput Oper Res* 40(10):2251–2259
19. Borndörfer R, Hoppmann H, Karbstein M, Löbel F (2018) The modulo network simplex with integrated passenger routing. In *Operations Research Proceedings 2016: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR)*, Helmut Schmidt University Hamburg, Germany, August 30–September 2, 2016, pages 637–644. Springer
20. Gattermann P, Großmann P, Nachtigall K, Schöbel A (2016) Integrating passengers' routes in periodic timetabling: a SAT approach. In *16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik
21. Löbel F, Lindner N, Borndörfer R (2020) The restricted modulo network simplex method for integrated periodic timetabling and passenger routing. In *Operations Research Proceedings 2019: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR)*, Dresden, Germany, September 4–6, 2019, pages 757–763. Springer
22. Polinder GJ, Cacchiani V, Schmidt M, Huisman D (2022) An iterative heuristic for passenger-centric train timetabling with integrated adaption times. *Comput Oper Res* 142:105740
23. Polinder GJ, Schmidt M, Huisman D (2021) Timetabling for strategic passenger railway planning. *Transp Res B Methodol* 146:111–135
24. Schmidt M, Schöbel A (2015) Timetabling with passenger routing. *OR spectrum* 37(1):75–97
25. Schiewe P, Goerigk M, Lindner N (2023) TimPassLib – A library for integrated periodic timetabling and passenger routing. <https://timpasslib.net>
26. Liebchen C (2006) Periodic timetable optimization in public transport. PhD thesis, Technical University Berlin
27. Bast H, Delling D, Goldberg A, Müller-Hannemann M, Pajor T, Sanders P, Wagner D, Werneck RF (2016) Route planning in transportation networks. In: Kliemann Lasse, Sanders Peter (eds) *Algorithm Engineering: Selected Results and Surveys*, vol 9220. Lecture Notes in Computer Science. Cham, Springer, pp 19–80
28. Lübke J (2009) Passagierrouting und Taktfahrplanoptimierung. Master's thesis, Technische Universität Berlin. Diploma thesis, Technische Universität Berlin
29. Gurobi Optimization, LLC (2023) Gurobi Optimizer Reference Manual
30. Deutsche Bahn AG (2017) Passagierzählung S-Bahn Hamburg. <https://data.deutschebahn.com/dataset/passagierzählung-s-bahn-hamburg.html>. Accessed 23 Jan 2023
31. SBB AG (2023) Open data platform mobility Switzerland. <https://opentransportdata.swiss/en>. Accessed 23 Jan 2023
32. Schiewe A, Albert S, Schiewe P, Schöbel A, Spühler F, Stinzendörfer M (2022) Documentation for LinTim 2022.08. <http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-69236>
33. Friedrich M, Hartl M, Schiewe A, Schöbel A (2017) Angebotsplanung im öffentlichen Verkehr - planerische und algorithmische Lösungen. In: HEUREKA'17: Optimierung in Verkehr und Transport. Heureka'17
34. Collection of open source public transport networks by DFG Research Unit "FOR 2083: Integrated Planning For Public Transportation" (2018). <https://github.com/FOR2083/PublicTransportNetworks>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.