An analysis of the parameterized complexity of periodic timetabling

Niels Lindner[1] · Julian Reisch[2]

Abstract

Public transportation networks are typically operated with a periodic timetable. The periodic event scheduling problem (PESP) is the standard mathematical modeling tool for periodic timetabling. PESP is a computationally very challenging problem: For example, solving the instances of the benchmarking library PESPlib to optimality seems out of reach. Since PESP can be solved in linear time on trees, and the treewidth is a rather small graph parameter in the networks of the PESPlib, it is a natural question to ask whether there are polynomial-time algorithms for input networks of bounded treewidth, or even better, fixed-parameter tractable algorithms. We show that deciding the feasibility of a PESP instance is NP-hard even when the treewidth is 2, the branchwidth is 2, or the carvingwidth is 3. Analogous results hold for the optimization of reduced PESP instances, where the feasibility problem is trivial. Moreover, we show W[1]-hardness of the general feasibility problem with respect to treewidth, which means that we can most likely only accomplish pseudo-polynomial-time algorithms on input networks with bounded tree- or branchwidth. We present two such algorithms based on dynamic programming. We further analyze the parameterized complexity of PESP with bounded cyclomatic number, diameter, or vertex cover number. For event-activity networks with a special—but standard—structure, we give explicit and sharp bounds on the branchwidth in terms of the maximum degree and the carvingwidth of an underlying line network. Finally, we investigate several parameters on the smallest instance of the benchmarking library PESPlib.

Keywords Periodic event scheduling problem · Periodic timetabling · Treewidth · Branchwidth · Carvingwidth

Mathematics Subject Classification 68Q17 · 90B20 · 68Q25 · 90B06 · 90B35 · 90C35 · 90C39

1 Introduction

Creating and optimizing timetables is substantial for planning and operating public transportation networks. Both in local traffic and in long-distance train networks, timetables are often periodic, i.e., the schedule of trips repeats after a certain time period T, e.g., 60 min.

Mathematically, periodic timetabling is captured by the *periodic event scheduling problem* (PESP, Serafini & Ukovich, 1989). The idea behind PESP is to model arrival and departure events of trips in a public transportation network as vertices (*events*) of a directed graph. Dependencies between

✉ Niels Lindner
  lindner@zib.de

  Julian Reisch
  julian.reisch@synoptics.de

[1] Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany

[2] Synoptics GmbH, Chemnitzer Str. 48b, 01187 Dresden, Germany

events, such as driving of a vehicle or changing at a station, are modeled as arcs (*activities*) connecting pairs of events. These activities come with restrictions on their duration, e.g., driving from one station to the next might take at least 7 min. Then, a solution to PESP is an assignment of times in $[0, T)$ to each event (a *periodic timetable*) such that the activity duration restrictions are respected. We refer to Sect. 2 for rigorous formulations.

Deciding whether a periodic timetable exists is an NP-complete problem, even if $T \geq 3$ is not considered as part of the input. This result can be proved by a polynomial-time reduction in T- VERTEX COLORING, where a T-coloring of a graph corresponds to event times in $\{0, 1, \ldots, T - 1\}$ (Odijk, 1994). This suggests a close relationship between PESP and coloring problems. In fact, both PESP and VERTEX COLORING are solvable in linear time on trees, regardless of T. Furthermore, for any T, deciding if a graph admits a T-coloring is fixed-parameter tractable when parameterized by treewidth. More precisely, there is a function f such that for

a given graph G on n vertices with a nice tree decomposition of treewidth $\leq k$ and a natural number T, there is an $O(f(k, T) \cdot n)$ algorithm deciding the T-Vertex Coloring problem on G (Arnborg & Proskurowski, 1989).

This motivates the investigation of parameterized complexity of PESP. In fact, the treewidth turns out to be rather small on the instances of the benchmarking library *PESPlib* (Goerigk, 2012). This library is a collection of 20 periodic timetabling problems, none of which could be solved to optimality in the past years. In Sect. 3, we recall some standard notions from parameterized complexity, list values or give bounds for several typical graph parameters for the smallest PESPlib instance R1L1, and give an overview on the complexity landscape for these parameters. The corresponding hardness results, algorithms, and the computation of the parameters make up the content of the following three sections of the paper.

We show in Sect. 4 that, in contrast to Vertex Coloring, PESP is not fixed-parameter tractable when parameterized by treewidth unless P = NP. Even if the event-activity network has treewidth 2 or branchwidth 2, i.e., every connected component is series–parallel, it is an NP-complete problem to decide whether a feasible periodic timetable exists. When considering *reduced* PESP instances, where the activities carry only lower bounds, but no (non-trivial) upper bounds, we prove that it is NP-complete to decide if there exists a periodic timetable whose weighted periodic slack is below a given threshold. Both proofs work by a reduction in the Subset Sum problem. As a by-product, we also obtain an NP-hardness result on networks of carvingwidth 3. As a consequence, if P $\neq$ NP, then there are only pseudo-polynomial-time algorithms available for both the feasibility and the reduced optimality problem. Moreover, the existence of fixed-parameter tractable algorithms allowing a unary encoding of the period time T is shown to be unlikely, as we show that deciding the feasibility of a PESP instance is W[1]-hard w.r.t. the vertex cover number and hence for the treewidth.

We construct a dynamic program that solves PESP based on branch decompositions in Sect. 5. This provides a pseudo-polynomial algorithm for networks of bounded branchwidth or treewidth. A straightforward tree decomposition analogue is presented in the Appendix. We also prove that the feasibility version of PESP is fixed-parameter tractable when parameterized by the cyclomatic number, i.e., the dimension of the cycle space of the event-activity network. For computing periodic timetables with minimum weighted periodic slack, we give a polynomial-time algorithm when the cyclomatic number is bounded.

As PESP instances arising from public transportation networks typically have a special structure, we show in Sect. 6 that on this type of event-activity networks, the branchwidth can be related to invariants of an underlying line network:

Roughly speaking, the number of lines at a station of a public transport network is a (sharp) lower bound on the branchwidth of the event-activity network. The carvingwidth of the often planar line network provides an upper bound, which is also sharp. Finally, we exploit the relation to line networks in order to compute bounds on the parameters discussed in this paper for the smallest PESPlib instance R1L1.

The presentation finishes with a few concluding remarks in Sect. 7.

2 The periodic event scheduling problem

The *periodic event scheduling problem* (PESP) was introduced in Serafini and Ukovich (1989). PESP instances comprise the following ingredients:

– a directed graph G (often called *event-activity network*) with vertex set $V(G)$ (*events*) and arc set $A(G)$ (*activities*),
– a *period time* $T \in \mathbb{N}$,
– *lower bounds* $\ell \in \mathbb{Z}_{\geq 0}^{A(G)}$ with $\ell_a < T$ for all $a \in A(G)$,
– *upper bounds* $u \in \mathbb{Z}_{\geq 0}^{A(G)}$ with $u_a \geq \ell_a$ for all $a \in A(G)$,
– *weights* $w \in \mathbb{Q}_{\geq 0}^{A(G)}$.

In the application of periodic timetabling in public transport, the events are typically arrivals or departures of a vehicle at a station, and activities model driving between stations, dwelling at a station, passenger transfers, or safety constraint such as minimum distances between vehicles (Liebchen & Möhring, 2007). The weights often reflect the number of passengers using an activity.

Definition 2.1 Given (G, T, ℓ, u) as above, a *periodic timetable* is a vector $\pi \in [0, T)^{V(G)}$ such that there exists a *periodic tension* $x \in \mathbb{R}_{\geq 0}^{A(G)}$ satisfying

$$\forall ij \in A(G) : \ell_{ij} \leq x_{ij} \leq u_{ij} \text{ and } \pi_j - \pi_i \equiv x_{ij} \bmod T.$$
(1)

Intuitively, π gives the cyclic order of the events, and x corresponds to the duration of the activities. If there exists a periodic timetable π, then a periodic tension can be computed by

$$x_{ij} := [\pi_j - \pi_i - \ell_{ij}]_T + \ell_{ij}, \quad ij \in A(G),$$
(2)

where $[\cdot]_T$ denotes the modulo-T-operator taking values in $[0, T)$. Conversely, from a vector $x \in \mathbb{R}_{\geq 0}^{A(G)}$ with $\ell \leq x \leq u$, one can construct a periodic timetable with tension x by a graph traversal; see also Lemma 2.7.

In terms of the incidence matrix B of G and its transpose B^t, condition (1) can be rewritten as

$$\ell \leq x \leq u \quad \text{and} \quad B^t \pi \equiv x \mod T.$$

Since B and hence are B^t are totally unimodular (Schrijver, 1986, Example 19.2) and the bounds ℓ, u are integer, it follows that if a periodic timetable exists, then there is also an integer timetable with an integer periodic tension. However, in general, it is not at all clear that a periodic timetable exists:

Definition 2.2 (T- PESP- FEASIBILITY) Given a tuple (G, T, ℓ, u) as above, decide if there exists a periodic timetable π.

Theorem 2.3 (Odijk, 1994) T- PESP- FEASIBILITY *is* NP-*complete for fixed* $T \geq 3$.

Proof It is clear that T- PESP- FEASIBILITY is in NP, as a feasible timetable π with periodic tension x serves as certificate. We recall the proof in order to emphasize the natural relationship between PESP and VERTEX COLORING. Recall that the T-VERTEX COLORING problem is to decide for an undirected graph H whether it admits a T-coloring, i.e., a map $f : V(H) \to \{0, \ldots, T-1\}$ such that $f(i) \neq f(j)$ for every pair (i, j) of adjacent vertices in H.

Fix $T \geq 3$. Given an undirected graph H, construct a T- PESP- FEASIBILITY instance (G, T, ℓ, u) as follows: G is obtained from H by arbitrarily directing the edges, and set $\ell := 1$, $u := T - 1$. Then, H admits a T-coloring if and only if (G, T, ℓ, u) has a periodic timetable π. Namely, if $f : V(H) \to \{0, \ldots, T-1\}$ is a T-coloring, then setting $\pi_i := f(i)$ for all $i \in V(G)$ is a periodic timetable, as for every arc $ij \in A(G)$ then holds $x_{ij} = [\pi_j - \pi_i]_T = [f(j) - f(i)]_T \in [1, T-1]$. Vice versa, if the PESP instance is feasible, then there is an integer timetable, giving rise to a T-coloring. □

So far, we have neglected the weight vector $w \in \mathbb{Q}_{\geq 0}^{A(G)}$. The weights come into play in the optimization variant of PESP, which we state as its corresponding decision version. If x is a periodic tension, we call $y := x - \ell \geq 0$ the *periodic slack*.

Definition 2.4 (T- PESP- OPTIMALITY) Given a tuple (G, T, ℓ, u, w) as above and a number M, find a periodic timetable π with periodic slack y such that $w^t y \leq M$.

Minimizing the weighted periodic slack, or equivalently, the weighted periodic tension, can be interpreted as minimizing the total travel time of all passengers in a public transportation network. Clearly, T- PESP- OPTIMALITY is an NP-hard optimization problem. However, in many non-railway public transport networks, minimum distances are neglected for planning, and the driving and dwelling times of

vehicles have a rather small span, so that they can be assumed as fixed. Contracting the corresponding activities yields a graph where only transfer activities remain, and these have typically no restrictions on their durations in the sense that lower and upper bounds differ by at least $T - 1$. This motivates the following specialization of PESP, called *reduced PESP* in Pätzold and Schöbel (2016):

Definition 2.5 (T- RPESP- OPTIMALITY) Given (G, T, ℓ, u, w) as above with $u \geq \ell + T - 1$ and a number M, find a periodic timetable π with periodic slack y such that $w^t y \leq M$.

Note that the feasibility problem is trivial to solve: In fact, *any* integral vector $\pi \in [0, T)^{V(G)}$ is a periodic timetable, because for any activity $ij \in A(G)$,

$$\ell_{ij} \leq x_{ij} = [\pi_j - \pi_i - \ell_{ij}]_T + \ell_{ij} \leq T - 1 + \ell_{ij} \leq u_{ij}.$$

Theorem 2.6 (Nachtigall, 1993) *For any fixed* $T \geq 3$, T-RPESP- OPTIMALITY *is* NP-*hard.*

Proof We adapt the proof of Nachtigall to our notions and notations. Fix some period time $T \geq 3$. We reduce T- PESP-FEASIBILITY to T- RPESP- OPTIMALITY.
Let (G, T, ℓ, u) be a T- PESP- FEASIBILITY instance. Without loss of generality, assume that $u - \ell < T$, because if the instance is feasible, then there is a periodic tension x satisfying $x < \ell + T$ by (2). Add to each arc $a \in A(G)$ from i to j a reverse copy $\bar{a}$ with $\ell_{\bar{a}} := [-u_a]_T$. Set all weights w to 1. For this T- RPESP- OPTIMALITY instance, let π be a periodic timetable with tension x as defined in (2). Then, for any original arc a holds $\ell_a \leq x_a < \ell_a + T$ and $x_a \equiv -x_{\bar{a}}$ mod T. For the slacks y_a and $y_{\bar{a}}$, we obtain

$$y_a + y_{\bar{a}} = [x_a - \ell_a]_T + [u_a - x_a]_T.$$

Since $0 \leq x_a - \ell_a \leq u_a - \ell_a < T$ and $-T < \ell_a - x_a \leq u_a - x_a \leq u_a - \ell_a < T$,

$$y_a + y_{\bar{a}} = \begin{cases} u_a - \ell_a & \text{if } u_a - x_a \geq 0, \\ u_a - \ell_a + T & \text{if } u_a - x_a < 0. \end{cases}$$

In particular, for the weighted slack of the T- RPESP- OPTIMALITY instance holds

$$\sum_{a \in A(G)} (y_a + y_{\bar{a}}) \geq \sum_{a \in A(G)} (u_a - \ell_a),$$

and equality holds if and only if $x_a \leq u_a$ for all $a \in A(G)$. This means that the T- PESP- FEASIBILITY instance is feasible if and only if the described T- RPESP- OPTIMALITY instance has weighted periodic slack at most $M := \sum_{a \in A(G)} (u_a - \ell_a)$. □

We turn now to simple algorithms for T- PESP- OPTI-MALITY. Consider at first instances where undirecting the event-activity network G results in a tree (shortly, G is a tree):

Lemma 2.7 *Suppose that G is a tree on n vertices. Then, T- PESP- OPTIMALITY on (G, T, ℓ, u, w) can be solved in $O(n)$ time. Moreover, ℓ is an optimal periodic tension, and the minimum weighted periodic slack is 0.*

Proof If undirecting G results in a tree on n vertices, then the transpose B^t of the incidence matrix B of G is a $(n-1) \times n$ matrix of full rank $n-1$. In particular, $B^t \pi = \ell$ has a solution over $\mathbb{Z}$, and reducing modulo T gives a feasible periodic timetable π^* with periodic tension ℓ and hence weighted periodic slack 0.

Avoiding linear algebra, π^* can as well be obtained by traversing the tree, starting with $\pi_v^* = 0$ at an initial vertex v and setting $\pi_j^* := [\pi_i^* + \ell_{ij}]_T$ when traversing $ij \in A(G)$, and $\pi_j^* := [\pi_i^* - \ell_{ij}]_T$ if $ji \in A(G)$ is traversed. A depth-first traversal takes $O(|V(G)| + |A(G)|) = O(n)$ time. □

On general networks, there are several ways to give naive exponential-time algorithms for T- PESP- OPTIMALITY:

Lemma 2.8 *On instances (G, T, ℓ, u, w) with n events and m activities, T- PESP- OPTIMALITY can be solved in*

1. $O^*(T^{n-1})$, or
2. $O^*(2^{n-1}n^{n-2})$, or
3. $O^*(3^m)$ time,

where $O^*(\cdot)$ means $O(\cdot)$ ignoring polynomial factors.

Proof

1. Enumerate all T^n integral vectors in $[0, T)^{V(G)}$, compute x by (2), and check the bounds. This is an $O(mT^n)$ algorithm. If π is a periodic timetable, then for any $d \in \mathbb{R}$, π' defined by $\pi_i' := [\pi_i + d]_T$ for all $i \in V(G)$ is a periodic timetable with the same periodic tension. In particular, only T^{n-1} vectors have to be enumerated.
2. By Nachtigall (1998), if G is weakly connected and the instance is feasible, there is an optimal periodic tension x^* and a spanning tree F of G such that $x_a^* \in \{\ell_a, u_a\}$ for all $a \in A(F)$. Enumerate all $O(n^{n-2})$ spanning trees F of G. For each such F, enumerate all 2^{n-1} vectors $x \in \prod_{a \in A(F)}\{\ell_a, u_a\}$. Interpreting x as a periodic tension defines a periodic timetable $\pi \in [0, T)^{V(G)}$ which can be computed by an $O(n + m)$ depth-first traversal as in the proof of Lemma 2.7 (replacing ℓ by x). Use (2) to compute the periodic tension x_a of all $O(m)$ remaining co-tree arcs $a \notin A(F)$ and check if the bounds are satisfied. If G is not connected, T- PESP- OPTIMALITY can be solved on each component individually.

3. Let B denote the incidence matrix of G. Then, the modulo constraint $B^t \pi \equiv x \mod T$ is satisfied if and only if there is a vector $p \in \mathbb{Z}^{A(G)}$ such that $B^t \pi = x - Tp$. Since any entry of $B^t \pi$ lies in the interval $(-T, T)$, and any tension computed by (2) satisfies $x \in [0, 2T)$, it suffices to consider $p \in \{0, 1, 2\}^{A(G)}$, cf. Liebchen (2006, Lemma 9.2). The algorithm is now to solve the problem

Minimize $w^t y$
s.t. $B^t \pi = x - Tp$,
                             $\ell \le x \le u$

for each fixed $p \in \{0, 1, 2\}^{A(G)}$. This is a series of 3^m linear programs. □

Somewhat unsurprisingly, Lemma 2.8 implies that all three presented PESP variants are solvable in polynomial time when the number of events or the number of activities is fixed. In the remainder of the paper, we investigate several graph parameters and their effects on the parameterized complexity of PESP.

3 Parameters

3.1 Parameterized Complexity

Parameterized complexity is an active topic in the context of scheduling; see, for example, Mnich & van Bevern, 2018. Before introducing the parameters discussed in this paper, we recall several standard notions from the field of parameterized complexity and refer to the books (Downey & Fellows, 1999) and (Cygan et al., 2015) for further details.

A *parameterized problem* over an alphabet Σ is a language $L \subseteq \Sigma^* \times \mathbb{N}$. A parameterized problem L is called *fixed-parameter tractable* (FPT) if there are an algorithm A, a computable function $f : \mathbb{N} \to \mathbb{N}$, and a constant $c \in \mathbb{N}$ such that for each $(x, k) \in \Sigma^* \times \mathbb{N}$, A decides whether $(x, k) \in L$ within $f(k) \cdot \text{size}(x, k)^c$ steps. In this case, we will call A an FPT algorithm. The class of FPT problems forms a complexity class, which will also be denoted by FPT.

A parameterized problem L belongs to the complexity class XP if there are an algorithm A and functions $f, g : \mathbb{N} \to \mathbb{N}$ such that A decides whether $(x, k) \in L$ within $f(k) \cdot \text{size}(x, k)^{g(k)}$ steps. Clearly, FPT $\subseteq$ XP.

Generalizing FPT algorithms to non-deterministic algorithms, we obtain the complexity class para-NP. We have FPT = para-NP if and only if P = NP, and no para-NP-hard problem can be in XP.

For parameterized problems $L \subseteq \Sigma_L^* \times \mathbb{N}$ and $M \subseteq \Sigma_M^* \times \mathbb{N}$, a *parameterized reduction* of L to M is an algorithm A

Table 1 Parameters of R1L1

Parameter	Value
No. of vertices	3664
No. of arcs	6385
Cyclomatic number	2722
Vertex cover number	1832
Diameter	88
Maximum degree	26
Branchwidth	$\in [58, 70]$
Treewidth	$\in [57, 97]$
Carvingwidth	$\in [29, 1820]$

that transforms an instance $(x, k) \in \Sigma_L^* \times \mathbb{N}$ to an instance $(x', k') \in \Sigma_M^* \times \mathbb{N}$ such that

1. $(x, k) \in L$ if and only if $(x', k') \in M$,
2. $k' \le f(k)$,
3. A requires $f(k) \cdot \text{size}(x)^c$ time,

where $f : \mathbb{N} \to \mathbb{N}$ is computable and $c \in \mathbb{N}$ is a constant, and both f and c are independent of (x, k).

Finally, we define the complexity class W[1] as the class of all parameterized problems that admit a parameterized reduction to the k- CLIQUE problem. It is known that FPT $\subseteq$ W[1] $\subseteq$ XP $\cap$ para-NP, and it is believed that FPT $\ne$ W[1].

3.2 Parameter search

We now return to PESP. The benchmarking library PES-Plib (Goerigk, 2012) contains 20 difficult T- PESP- OPTI-MALITY instances, created with the LinTim toolbox with data of the German railway network (Goerigk et al., 2013). Despite several efforts, none of the instances is currently solved to proven optimality.

For the event-activity network of the smallest PESPlib instance R1L1, the values of several typical graph parameters in parameterized complexity are summarized in Table 1. Most of the parameters are easy to obtain; we will elaborate in Sect. 6 on the computation of the others. The cyclomatic number, also known as feedback edge set number, is a common measure for the difficulty of PESP instances, as it counts the number of integral variables used in a cycle-based mixed integer programming formulation (Borndörfer et al., 2019). However, parameters such as treewidth and branchwidth are much smaller. As T- PESP- OPTIMALITY is solvable in linear time on trees by Lemma 2.7, it seems therefore reasonable to investigate parameterized algorithms in terms of treewidth. In particular, we could ask whether there is a polynomial-time algorithm for PESP on event-activity networks with bounded treewidth, i.e., T- PESP- OPTIMALITY belongs to XP. Even
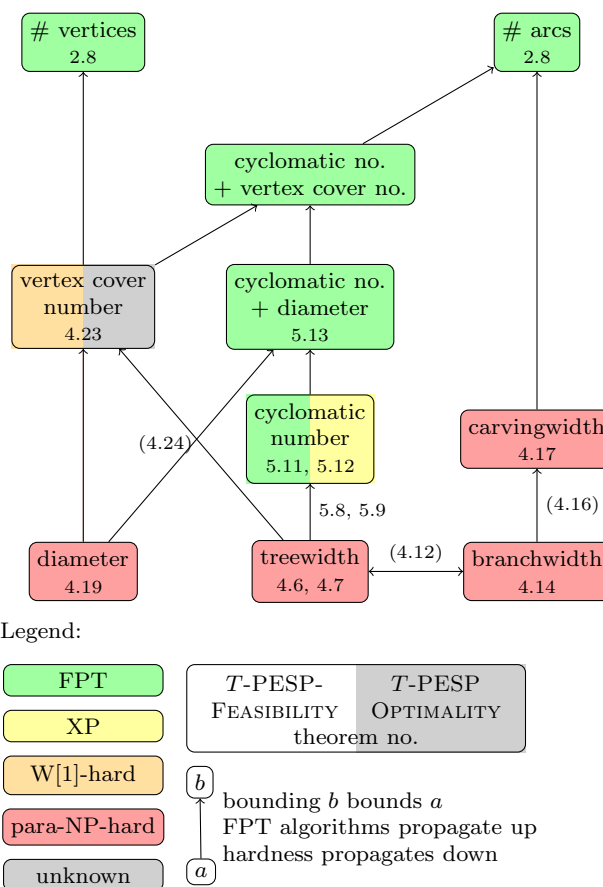

Fig. 1 Parameter hierarchy and complexity results for T- PESP- FEA-SIBILITY and T- PESP- OPTIMALITY

better, we wonder if there is an FPT algorithm. Unfortunately, the answer is negative unless P = NP.

The complexity landscape of the parameters discussed in this paper is manifold. An overview of the results and the relations between the parameters is depicted in Fig. 1. We will introduce the parameters in the next two sections, presenting hardness results in Sect. 4 and algorithms in Sect. 5.

4 Hardness results

4.1 Para-NP-Hardness

4.1.1 Reducing SUBSET SUM

Definition 4.1 *(Garey & Johnson, 1979, SP13)* The SUBSET SUM problem is the following: Given $r \in \mathbb{N}$, $c \in \mathbb{Z}_{\ge 0}^r$ and $C \in \mathbb{Z}_{\ge 0}$ with $C \le \sum_{i=1}^r c_i$, is there a $z \in \{0, 1\}^r$ such that $c^t z = C$?

The SUBSET SUM problem is weakly NP-complete (Karp, 1972). We will at first construct a polynomial-time reduction in SUBSET SUM to T- PESP- FEASIBILITY.
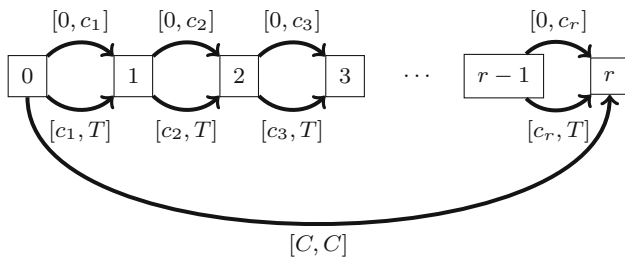
Fig. 2 Instance $I(r, c, C)$: arcs a are labeled with $[\ell_a, u_a]$, $T := \sum_{i=1}^{r} c_i + 1$

Definition 4.2 For a SUBSET SUM instance (r, c, C) as above, define $I(r, c, C)$ as the instance (G, T, ℓ, u) for T-PESP-FEASIBILITY as depicted in Fig. 2 with $T := \sum_{i=1}^{r} c_i + 1$.

Lemma 4.3 *The* SUBSET SUM *instance* (r, c, C) *has a solution if and only if* T-PESP-FEASIBILITY *has a solution on the instance* $I(r, c, C)$.

Proof Let π be a periodic timetable for $I(r, c, C)$. Then, $[\pi_i - \pi_{i-1}]_T \in \{0, c_i\}$ holds for all $i \in \{1, \ldots, r\}$. Set

$$z_i := \begin{cases} 1 & \text{if } [\pi_i - \pi_{i-1}]_T = c_i, \\ 0 & \text{otherwise,} \end{cases} \quad i = 1, \ldots, r.$$

For the arc $(0, r)$, we then obtain

$$\begin{aligned} C = [C]_T &= [\pi_r - \pi_0]_T \\ &= \left[\sum_{i=1}^{r} (\pi_i - \pi_{i-1}) \right]_T \\ &= \left[\sum_{i=1}^{r} z_i c_i \right]_T = \sum_{i=1}^{r} z_i c_i, \end{aligned}$$

and found a positive answer to the SUBSET SUM problem on (r, c, C). Conversely, any vector $z \in \{0, 1\}^r$ such that $c^t z = C$ yields a feasible periodic timetable π by setting

$$\pi_0 := 0 \quad \text{and}$$
$$\pi_i := \pi_{i-1} + z_i c_i, \quad i = 1, \ldots, r.$$

$\square$

4.1.2 Treewidth

Definition 4.4 *(e.g., Robertson & Seymour, 1984)* Given a graph G, a *tree decomposition* of G is a pair $(\mathcal{T}, \mathcal{X})$ consisting of a tree $\mathcal{T}$ and a family of *bags* $\mathcal{X} = (X_t)_{t \in V(\mathcal{T})}$ with $X_t \subseteq V(G)$ for each $t \in V(\mathcal{T})$ such that

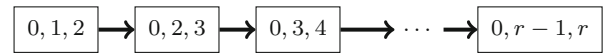1. $\bigcup_{t \in V(\mathcal{T})} X_t = V(G)$,

Fig. 3 An optimal tree decomposition of width 2 for the $I(r, c, C)$ network

2. for each $a \in A(G)$, there is a bag X_t containing both endpoints of a,
3. for each $v \in V(G)$, the subforest of $\mathcal{T}$ induced by $\{t \in V(\mathcal{T}) \mid v \in X_t\}$ is connected.

The *width* of a tree decomposition is $\max_{t \in V(\mathcal{T})} |X_t| - 1$, and the *treewidth* of G is defined as the minimum possible width of a tree decomposition, i.e.,

$$\text{tw}(G) := \min \left\{ \max_{t \in V(\mathcal{T})} |X_t| \,\middle|\, \begin{array}{l} (\mathcal{T}, \mathcal{X}) \text{ is a tree} \\ \text{decomposition of } G \end{array} \right\} - 1.$$

This definition applies to both undirected and directed graphs, and as well to multigraphs. According to Definition 4.4, the treewidth of a graph with multiple edges equals the treewidth of the graph where all multiple edges between two vertices are replaced by a single edge. The simple connected graphs of treewidth 1 are precisely the trees.

Lemma 4.5 *For any* SUBSET SUM *instance* (r, c, C) *with* $r \geq 2$, *the event-activity network* G *of the instance* $I(r, c, C)$ *has treewidth 2.*

Proof The path of length r depicted in Fig. 3, where each node is labeled with its bag, is a tree decomposition of G. Checking the properties of a tree decomposition is straightforward. The maximum bag size is 3, and hence $\text{tw}(G) \leq 2$. Removing one of the two arcs from i to $i+1$, $i = 0, \ldots, r-1$, does not change the treewidth in the sense of Definition 4.4. As $r \geq 2$, this results in a simple graph containing a cycle, so that $\text{tw}(G) \geq 2$. $\square$

An alternative way to see that the network G of $I(r, c, C)$ has treewidth at most 2 is to observe that G is series–parallel (see, for example, Bodlaender & van Antwerpen-de Fluiter, 2001, Lemma 3.4). With Lemmas 4.3 and 4.5, we obtain:

Theorem 4.6 T-PESP-FEASIBILITY *is* NP-*complete on networks of treewidth 2.*

Since the transformation in the proof of Theorem 2.6 reduces T-PESP-FEASIBILITY to the T-RPESP-OPTIMALITY problem by adding antiparallel arcs, which does not alter the treewidth, we have moreover:

Theorem 4.7 T-RPESP-OPTIMALITY *is* NP-*complete on networks of treewidth 2.*

Remark 4.8 If simple graphs are desired, one can dispose of the parallel arcs in $I(r, c, C)$ by subdividing any arc with

bounds $[0, c_i]$ into two arcs with bounds $[0, c_i]$ and $[0, 0]$ without affecting the feasibility of the PESP instance. Moreover, one checks that this does not increase the treewidth of $I(r, c, C)$.

Remark 4.9 In the proof of Lemma 4.3, the period time T is chosen very large. The above NP-completeness theorems hence do not hold when T is fixed. We will give a pseudo-polynomial algorithm for T- PESP- OPTIMALITY with bounded branchwidth in Sect. 5, showing that fixing both T and the treewidth results in a polynomial-time algorithm.

Remark 4.10 We want to remark that there has already been a paper (van Heuven van Staereling, 2018) titled *Tree Decomposition Methods for the periodic event scheduling problem*. However, the title is misleading, as the algorithm there is about finding trees *in* the network rather than considering tree decompositions in the usual sense.

4.1.3 Branchwidth

Definition 4.11 (*Robertson & Seymour, 1991*) Given a graph G, a *branch decomposition* of G is a pair $(\mathcal{B}, \varphi)$, where $\mathcal{B}$ is a tree such that every non-leaf node has degree 3, and φ is a bijection from the leaves of $\mathcal{B}$ to $A(G)$. Deleting an edge e of $\mathcal{B}$ disconnects $\mathcal{B}$ into two subtrees and hence partitions the leaves of $\mathcal{B}$ into two sets. Applying φ, this yields a partition $A = A_e^1 \cup A_e^2$. This defines in turn a *vertex separator* $S_e \subseteq V$ as the set of vertices that are incident both to an edge in A_e^1 and to an edge in A_e^2.

The *width* of a branch decomposition $(\mathcal{B}, \varphi)$ is defined as $\max_{e \in E(\mathcal{B})} |S_e|$. The *branchwidth* of G is then the minimum possible width of a branch decomposition, i.e.,

$$\mathrm{bw}(G) := \min \left\{ \max_{e \in E(\mathcal{B})} |S_e| \; \middle| \; \begin{array}{l} (\mathcal{B}, \varphi) \text{ is a branch} \\ \text{decomposition of } G \end{array} \right\}.$$

Treewidth and branchwidth are related as follows:

Theorem 4.12 (Robertson & Seymour, 1991, 5.1) *If* $\mathrm{bw}(G) \geq 2$, *then*

$$\mathrm{bw}(G) \leq \mathrm{tw}(G) + 1 \leq \left\lfloor \frac{3}{2} \mathrm{bw}(G) \right\rfloor.$$

It follows immediately that both T- PESP- FEASIBILITY and T- RPESP- OPTIMALITY are NP-complete on networks with branchwidth 3. We prove below that the NP-completeness is already given for branchwidth 2.

Lemma 4.13 *For any* SUBSET SUM *instance* (r, c, C), *the event-activity network G of the instance $I(r, c, C)$ has branchwidth 2.*
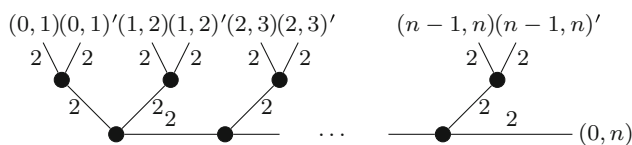

Fig. 4 An optimal branch decomposition of width 2 for the $I(r, c, C)$ network

Proof Figure 4 shows a branch decomposition of G, where the leaves are labeled with the corresponding arc and the edges are labeled with the cardinality of the corresponding vertex separator. This is clearly a branch decomposition. Checking the cardinalities of the vertex separators is again straightforward, and hence $\mathrm{bw}(G) \leq 2$. The network G cannot have branchwidth 1, as in any branch decomposition, the edge incident to the leaf representing $(0, r)$ always induces the vertex separator $\{0, r\}$ of size 2. $\square$

Theorem 4.14 *The problems T- PESP- FEASIBILITY and T- RPESP- OPTIMALITY are* NP-*complete on networks with branchwidth at most 2.*

Proof For T- PESP- FEASIBILITY, this follows by combining Lemma 4.3 with Lemma 4.13. Introducing anti-parallel arcs in the network of $I(r, c, C)$ does not increase the branchwidth of 2: In the branch decomposition of the proof of Lemma 4.13, replace a leaf with a vertex of degree 3 adjacent to two new leaves corresponding to the two anti-parallel arcs. The new edges have vertex separators of size 2. Therefore, the transformation in the proof of Theorem 2.6 does not alter the branchwidth, and T- RPESP- OPTIMALITY is NP-complete on networks with branchwidth at most 2. $\square$

Another approach to prove Lemma 4.13 and Theorem 4.14 is to exploit that a graph of treewidth at most 2 has branchwidth at most 2 (combine, for example, Bodlaender & van Antwerpen-de Fluiter, 2001, Lemma 3.5 with Robertson & Seymour, 1991, 4.2).

4.1.4 Carvingwidth

Carvingwidth is defined analogously to branchwidth, by labeling the leaves of an unrooted binary tree with the vertices of the original graph instead of the edges.

Definition 4.15 (*Seymour & Thomas, 1994*) Given a graph G, a *carving decomposition* of G is a pair $(\mathcal{C}, \psi)$, where $\mathcal{C}$ is a tree such that every non-leaf node has degree 3, and ψ is a bijection from the leaves of $\mathcal{C}$ to $V(G)$. Removing an edge e of $\mathcal{C}$ induces a partition of the leaves of $\mathcal{C}$, and hence via ψ also a partition $V(G) = V_e^1 \,\dot\cup\, V_e^2$. Let $\delta(V_e^1) \,(= \delta(V_e^2))$ denote the set of cut edges.
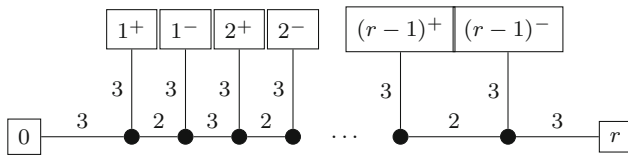
Fig. 5 An optimal carving decomposition of width 3 of the modified $I(r, c, C)$ network

The maximum cardinality of $\delta(V_e^1)$ taken over all $e \in E(\mathcal{C})$ is the *width* of $(\mathcal{C}, \psi)$. The *carvingwidth* of G is defined as

$$\text{cw}(G) := \min \left\{ \max_{e \in E(\mathcal{C})} |\delta(V_e^1)| \, \middle| \, \begin{array}{l} (\mathcal{C}, \psi) \text{ is a carving} \\ \text{decomposition of } G \end{array} \right\}.$$

The definition of carvingwidth applies as well to multigraphs, but in contrast to branch- and treewidth, it is sensitive to multiple edges: The carvingwidth is at least the maximum vertex degree $\Delta(G)$, as for each $v \in V(G)$, $\text{cw}(G) \geq \deg(v) = |\delta(v)|$. Carvingwidth is related to branchwidth as follows:

Theorem 4.16 (Nestoridis & Thilikos, 2014; Eppstein, 2018) *Let G be a graph with maximum vertex degree $\Delta(G)$. Then,*

$$\max \left(\Delta(G), \left\lceil \frac{1}{2} \text{bw}(G) \right\rceil \right) \leq \text{cw}(G) \leq \Delta(G) \cdot \text{bw}(G).$$

Theorem 4.17 *T-PESP-FEASIBILITY is NP-complete on networks of carvingwidth at most 3, and T-RPESP-OPTIMALITY is NP-complete on networks of carvingwidth at most 6.*

Proof Let $r \geq 2$ and consider again an instance of the form $I(r, c, C)$ with event-activity network G. Then, $\Delta(G) = 4$, so that $\text{cw}(G) \geq 4$. For all $i \in \{1, \ldots, r-1\}$, split vertex i into two new vertices i^+ and i^-, connected by a single directed arc (i^+, i^-) with bounds $\ell_{i^+,i^-} = u_{i^+,i^-} = 0$. The splitting is done in such a way that the arcs entering i are now entering i^+, and the arcs leaving i are leaving i^-. Any periodic timetable has the same value at i^+ and i^-, so that the proof of Lemma 4.3 carries over. However, the modified graph has maximum degree 3. The carving decomposition in Fig. 5, where the edges labeled with the number of cut edges, has width 3. Hence, T-PESP-FEASIBILITY is NP-complete on networks of carvingwidth 3. As a consequence, keeping in mind the arc duplication occurring in the proof of Theorem 2.6, T-RPESP-OPTIMALITY is NP-complete for carvingwidth 6. □

We also obtain from the proof of Theorem 4.17 that T-PESP-FEASIBILITY is in fact NP-complete on networks with maximum degree 3.

Remark 4.18 T-PESP-OPTIMALITY is trivial to solve on graphs G with $\text{cw}(G) = 1$, as then $\Delta(G) \leq 1$ by Theorem 4.16. If $\text{cw}(G) = 2$, then $\Delta(G) \leq 2$, so that any weakly connected component of G, seen as an undirected graph, is either a path or a cycle. T-PESP-OPTIMALITY is solvable in linear time on paths (Lemma 2.7). We will see later in Theorem 5.12 that T-PESP-OPTIMALITY admits a polynomial-time algorithm when the cyclomatic number is bounded by a constant, and in particular on a single cycle.

4.1.5 Diameter

The *diameter* of a graph is the maximum length of an undirected shortest path between two vertices.

Lemma 4.19 *T-PESP-FEASIBILITY is NP-hard on networks with diameter 1.*

Proof For the first statement, we turn the instances $I(r, c, C)$ used in Lemma 4.3 into complete graphs by adding activities a with $\ell_a = 0$ and $u_a = T - 1$ and observe that this does not alter the feasibility. □

In Sect. 5, we will give an FPT algorithm for T-PESP-OPTIMALITY when both diameter and cyclomatic number are bounded.

4.2 W[1]-Hardness

LIST COLORING dates back to Vizing (1976) and Erdős et al. (1980). It is known to be W[1]-hard when parameterized by treewidth (Fellows et al., 2011), or by the vertex cover number (Fiala et al., 2011, Theorem 1). We will now describe a parameterized reduction in the LIST COLORING problem to T-PESP-FEASIBILITY, which will be useful for two purposes.

At first, we can deduce the W[1]-hardness of T-PESP-FEASIBILITY w.r.t. the vertex cover number.

Secondly, note that reducing the SUBSET SUM problem does still allow for FPT algorithms for T-PESP-FEASIBILITY w.r.t. treewidth or branchwidth when the period time T is encoded in unary. We show that in this situation, T-PESP-FEASIBILITY is W[1]-hard, i.e., there is most likely no such "pseudo"-FPT algorithm. In particular, we can only hope for "pseudo"-XP algorithms, where the exponent of T is a nonconstant function of the parameter. We will indeed present such an algorithm in Sect. 5.

4.2.1 Reducing LIST COLORING

Definition 4.20 The LIST COLORING problem is the following: Given a graph H together with a finite list $L(v) \subseteq \mathbb{Z}_{\geq 0}$ for each $v \in V(H)$, decide whether there is a vertex coloring of H such that each vertex v is colored with a color in $L(v)$.

Given a LIST COLORING instance (H, L), we construct a T-PESP-FEASIBILITY instance (G, T, ℓ, u) as follows: Define $T := \max_{v \in V(H)} L(v) + 1$. Let G be any orientation of H. For each edge of H, we define for the corresponding activity a in $A(G)$ the bounds $\ell_a := 1$ and $u_a := T - 1$. Further add a new vertex v_0 to G. Let $v \in V(H)$ with $L(v) = \{c_1, \ldots, c_r\}, c_1 < \cdots < c_r$. Add r parallel activities $a_1, \ldots, a_r$ from v_0 to v, with bounds

$$\ell_{a_i} := c_i, \quad u_{a_i} := c_{i-1} + T, \quad i = 1, \ldots, r,$$

where we set $c_0 := c_r - T$. This way, we model the disjunctive constraints of choosing a color in $L(v)$ as a PESP instance (Liebchen & Möhring, 2007, §3.3).

We claim that (H, L) has a feasible list coloring if and only if (G, T, ℓ, u) admits a feasible periodic timetable. Thus, let $\pi \in [0, T)^{V(H)}$ be a list coloring for (H, L). If $ij \in A(H)$, then $\pi_j \neq \pi_i$, so that $[\pi_j - \pi_i - 1]_T \leq T - 2$ is a feasible periodic slack. Extend π to a timetable on G by setting $\pi_{v_0} := 0$. Let $v \in V(H)$, and assume that v is colored with the jth color from its list, i.e., $\pi_v = c_j \in L(v)$. For $i \in \{1, \ldots, r\}$, for the ith activity a_i from v_0 to v, the periodic tension would be

$$[\pi_v - \pi_{v_0} - \ell_{a_i}]_T + \ell_{a_i} = [c_j - c_i]_T + c_i$$
$$= \begin{cases} c_j & \text{if } i \leq j, \\ c_j + T & \text{if } i > j. \end{cases}$$

In the first case $c_j \leq T \leq c_{i-1} + T$, and in the second $c_i > c_j$ implies $c_j + T \leq c_{i-1} + T$. We conclude that π is a feasible periodic timetable.

Conversely, consider a feasible periodic timetable $\pi \in [0, T)^{V(G)}$. By a shift replacing π by $[\pi - \pi_{v_0}]_T$, we can assume that $\pi_{v_0} = 0$. By restriction, using that $\pi_j \neq \pi_i$ for all $ij \in A(H)$, π yields a coloring of H. It remains to check that this is a feasible list coloring. The periodic tension on an activity a_i from v_0 to v must satisfy

$$[\pi_v - \pi_{v_0} - \ell_{a_i}]_T + \ell_{a_i} = [\pi_v - c_i]_T + c_i \leq c_{i-1} + T$$

for all $i \in \{1, \ldots, r\}$. Suppose that there is an index j such that $c_j \leq \pi_v < c_{j+1}$. Then, the above inequality for $i = j+1$ means

$$\pi_v + T = [\pi_v - c_{j+1}]_T + c_{j+1} \leq c_j + T,$$

hence $\pi_v = c_j$. If $\pi_v \geq c_r$, then $\pi_v \leq c_0 + T = c_r$. Finally, if $\pi_v < c_1$, then $\pi_v \leq c_r - T < 0$, contradicting $\pi_v \geq 0$. This shows that $\pi_v \in \{c_1, \ldots, c_r\}$, so that π is indeed a feasible list coloring.

We have hence proved the following:

Lemma 4.21 *A* LIST COLORING *instance (H, L) has a solution if and only if (G, T, ℓ, u) is feasible.*

4.2.2 Vertex cover number

Definition 4.22 For a graph G, its *vertex cover number* $\mathrm{vc}(G)$ is defined as the minimum cardinality of a vertex cover of G.

Theorem 4.23 T-PESP-FEASIBILITY *is* W[1]-*hard when parameterized by the vertex cover number.*

Proof LIST COLORING is W[1]-hard when parameterized by the vertex cover number (Fiala et al., 2011, Theorem 1). Construct a T-PESP-FEASIBILITY instance (G, T, ℓ, u) from a LIST COLORING instance (H, L) as in Lemma 4.21. Since all arcs in G not present in H are connected to v_0, we obtain $\mathrm{vc}(G) \leq \mathrm{vc}(H) + 1$. $\qquad\square$

It remains unclear whether T-PESP-FEASIBILITY or T-PESP-OPTIMALITY admit polynomial-time algorithms for fixed vertex cover number, i.e., if these problems belong to XP. In Sect. 5.3, we show that fixing both vertex cover number and cyclomatic number enables an FPT algorithm for T-PESP-OPTIMALITY.

4.2.3 Treewidth and diameter revisited

The treewidth of a graph is known to be bounded by its vertex cover number:

Lemma 4.24 (Fiala et al., 2011, §2) *Let G be a graph. Then, $\mathrm{tw}(G) \leq \mathrm{vc}(G)$.*

Moreover, the vertex cover number bounds the diameter, as any path of length d contains at least $d/2$ vertices of any vertex cover. As a direct consequence of Theorem 4.23, we hence obtain:

Corollary 4.25 T-PESP-FEASIBILITY *is* W[1]-*hard when parameterized by treewidth, branchwidth, or diameter.*

5 Parameterized algorithms

5.1 A dynamic program for bounded branchwidth

5.1.1 PESP and vertex separators

The SUBSET SUM problem is weakly NP-complete and can be solved by pseudo-polynomial-time algorithms. In the following, we present a dynamic program for T-PESP-OPTIMALITY running in pseudo-polynomial time for event-activity networks of bounded branchwidth. Since T-PESP-OPTIMALITY comprises both T-PESP-FEASIBILITY

and T- RPESP- OPTIMALITY, this implicitly gives pseudo-polynomial time algorithms for these problems as well.

The key insight for our dynamic programming approach is the following decomposition property: Let $I = (G, T, \ell, u, w)$ be a T- PESP- OPTIMALITY instance. For any partition $A(G) = A^1 \,\dot\cup\, A^2$, we can partition I into two subinstances I^1 resp. I^2 restricted to the activities in A^1 resp. A^2. If y is a feasible periodic slack on I, then the restrictions y^1 resp. y^2 to I^1 resp. I^2 yield feasible periodic slacks with the property

$$\sum_{a \in A(G)} w_a y_a = \sum_{a \in A^1} w_a y_a^1 + \sum_{a \in A^2} w_a y_a^2. \tag{3}$$

On the level of timetables, we obtain from a timetable π on I two timetables π^1 resp. π^2 on I^1 resp. I^2 such that π, π^1, and π^2 all coincide when restricted to the events of the vertex separator S associated with the partition of $A(G)$ as in Definition 4.11.

Conversely, we can glue two periodic timetables π^1 and π^2 together to a timetable π on I if the restrictions to a vertex separator S satisfy $\pi^1|_S = \pi^2|_S$. The corresponding periodic slacks then hold Eq. (3).

Definition 5.1 Let $I = (G, T, \ell, u, w)$ be a T- PESP- OPTIMALITY instance, and let $S \subseteq V(G)$. For a vector $\rho \in [0, T)^S$, define $\mathrm{OPT}(I, S, \rho)$ as the minimum weighted slack of a periodic timetable π on I when additionally $\pi|_S = \rho$ is required.

For minimum weighted slacks, the above discussion shows the following:

Lemma 5.2 *Let $I = (G, T, \ell, u, w)$ be a feasible T- PESP-OPTIMALITY instance. Let $A(G) = A^1 \,\dot\cup\, A^2$ be a partition with vertex separator S giving subinstances I^1 and I^2 of I. Then,*

$$\mathrm{OPT}(I) = \min\{\, \mathrm{OPT}(I^1, S, \pi|_S) + \mathrm{OPT}(I^2, S, \pi|_S) \mid$$
$$\pi \text{ is a feasible per. timetable on } I\}.$$

More generally, if additionally the timetable is fixed on $W \subseteq V(G)$ to $\rho \in [0, T)^W$, then

$$\mathrm{OPT}(I, W, \rho) = \min\{\, \mathrm{OPT}(I^1, S \cup W^1, \pi|_{S \cup W^1})$$
$$+ \,\mathrm{OPT}(I^2, S \cup W^2, \pi|_{S \cup W^2}) \mid$$
$$\pi \text{ is a feasible periodic timetable}$$
$$\text{on } I \text{ with } \pi|_W = \rho\},$$

where $W^i = W \cap V^i$ denotes the intersection of W with the set of events V^i of I^i, $i = 1, 2$.

5.1.2 A branch decomposition approach

Since branch decompositions naturally encode vertex separators, we describe at first a branch-decomposition-based dynamic program for T- PESP- OPTIMALITY. Let $I = (G, T, \ell, u, w)$ be a T- PESP- OPTIMALITY instance. We assume that G is 2-edge-connected when seen as undirected graph. Let $(\mathcal{B}, \varphi)$ be a branch decomposition of G with node set $V(\mathcal{B})$ and edge set $E(\mathcal{B})$. Subdivide an arbitrary edge of $E(\mathcal{B})$ and call the new node τ the *root*. Recall from Definition 4.11 that every edge $e \in E(\mathcal{B})$ corresponds to a partition $A = A_e^1 \,\dot\cup\, A_e^2$ with vertex separator S_e. We assume that A_e^1 is the subset of activities coming from the component of $\mathcal{B} \setminus \{e\}$ not containing the root τ.

Algorithm 5.3 For each edge $e \in E(\mathcal{B})$, we compute an $|S_e|$-dimensional table F_e having an entry for each $\pi \in \{0, \ldots, T-1\}^{S_e}$. The table F_e is filled by a dynamic program starting from the edges $e \in E(\mathcal{B})$ incident to leaves and with decreasing distance to the root:

1. If $e \in E(\mathcal{B})$ is incident to a leaf corresponding via φ to an activity $ij \in A(G)$, then set

$$F_e(\pi) := \begin{cases} w_{ij} y_{ij} & \text{if } y_{ij} \leq u_{ij} - \ell_{ij}, \\ \infty & \text{otherwise,} \end{cases}$$

where $y_{ij} := [\pi_j - \pi_i - \ell_{ij}]_T$.

2. If e is incident to two edges e_1, e_2 with larger distance from the root, then set

$$F_e(\pi) := \min\{F_{e_1}(\pi'|_{S_{e_1}}) + F_{e_2}(\pi'|_{S_{e_2}}) \mid$$
$$\pi' \in \{0, \ldots, T-1\}^{S_{e_1} \cup S_{e_2}}, \pi'|_{S_e} = \pi\}.$$

3. If the tables of the two edges e_1, e_2 incident to τ have been computed, return

$$\min\{F_{e_1}(\pi) + F_{e_2}(\pi) \mid \pi \in \{0, \ldots, T-1\}^{S_{e_1}}\}.$$

Lemma 5.4 *Let $e \in E(\mathcal{B})$, $\pi \in \{0, \ldots, T-1\}^{S_e}$. Denote by I_e the subinstance of I containing precisely the activities in A_e^1.*

1. *If $F_e(\pi) < \infty$, then $F_e(\pi) = \mathrm{OPT}(I_e, S_e, \pi)$.*
2. *If $F_e(\pi) = \infty$, then I_e is infeasible.*

Proof Recall from Sect. 2 that it suffices to consider timetables with values in the discrete set $\{0, \ldots, T-1\}$, as ℓ and u are integer.

If e is incident to a leaf associated with an activity $ij \in A(G)$, then $A_e^1 = \{ij\}$ and $S_e = \{i, j\}$, as G is 2-edge-connected. Hence, $\mathrm{OPT}(I_e, S_e, \pi)$ is the minimum weighted slack of the activity ij when the timetable at i resp. j is fixed

to π_i resp. π_j. Therefore, we set $F_e(\pi)$ to $w_{ij}[\pi_j - \pi_i - \ell_{ij}]_T$ if the slack $[\pi_j - \pi_i - \ell_{ij}]_T$ is feasible and otherwise to ∞.

Otherwise, let e be adjacent to $e_1, e_2 \in E$, with e_1, e_2 having larger distance from the root than e. Then, $A_e^1 = A_{e_1}^1 \dot\cup A_{e_2}^1$ and hence $S_e \subseteq S_{e_1} \cup S_{e_2}$. Moreover, $(S_{e_1} \cup S_{e_2}) \setminus S_e$ is the vertex separator of the partition of $A_e^1 = A_{e_1}^1 \dot\cup A_{e_2}^1$. Applying Lemma 5.2 for $W = S_e$ and $S = (S_{e_1} \cup S_{e_2}) \setminus S_e$ yields the formula in Algorithm 5.3. $\qquad\square$

Lemma 5.5 *If $k = \max_{e \in E(\mathcal{B})} |S_e|$ and $m = |A(G)|$, then Algorithm 5.3 computes $\mathrm{OPT}(I)$ or decides that I is infeasible in $O(mT^{\lfloor 3k/2 \rfloor})$ time.*

Proof We first consider correctness. At the root τ with incident edges e_1, e_2, Algorithm 5.3 computes the tables F_{e_1}, F_{e_2} such that $F_{e_i}(\pi) = \mathrm{OPT}(I_{e_i}, S_{e_i}, \pi)$ for all π and $i = 1, 2$ by Lemma 5.4. Observe that $A_{e_1}^1 = A_{e_2}^2$, $A_{e_1}^2 = A_{e_2}^1$, and $S_{e_1} = S_{e_2}$, so that by the first equation in Lemma 5.2,

$$\mathrm{OPT}(I) = \min\{\mathrm{OPT}(I_{e_1}, S_{e_1}, \pi) + \mathrm{OPT}(I_{e_2}, S_{e_2}, \pi) \mid$$
$$\pi \in \{0, \ldots, T-1\}^V \text{ feasible timetable}\}.$$

This is precisely reflected in the third step of Algorithm 5.3, treating infeasible subinstances with infinite objective value.

Concerning running time, Step 1 can be done in $O(T^2)$ time and is called m times. Step 3 takes $O(T^k)$ time since $|S_{e_1}| \leq k$. As any node in the rooted branch decomposition has either 0 or 2 children and there are m leaves, there are $m - 1$ edges for which Step 2 is called. Each of the $|S_e|$ table entries requires to take a minimum over $|(S_{e_1} \cup S_{e_2}) \setminus S_e|$ previously computed table entries.

We claim that

$$2|S_{e_1} \cup S_{e_2}| \leq |S_{e_1}| + |S_{e_2}| + |S_e|.$$

If $i \in S_{e_1} \setminus S_{e_2}$, then i is adjacent to an activity $a \in A_{e_1}^1$ and $a' \notin A_{e_1}^1$. Since $A_{e_1}^1$ and $A_{e_2}^1$ are disjoint, $a \notin A_{e_2}^1$. As $i \notin S_{e_2}$, then also $a' \notin A_{e_2}^1$, and consequently $a' \notin A_e^1 = A_{e_1}^1 \cup A_{e_2}^1$. It follows that $i \in S_e$, and any such i appears hence twice in the right-hand side of the above inequality. By symmetry, the same holds for all $i \in S_{e_2} \setminus S_{e_1}$. Clearly, any $i \in S_{e_1} \cap S_{e_2}$ is counted both in $|S_{e_1}|$ and in $|S_{e_2}|$. This proves the claim. The relation between the separators is also depicted in Fig. 6.

Since the size of any of the three vertex separators is bounded by k, we obtain

$$|S_e| + |(S_{e_1} \cup S_{e_2}) \setminus S_e| = |S_{e_1} \cup S_{e_2}| \leq \left\lfloor \frac{3k}{2} \right\rfloor.$$

Thus, Step 2 accounts in total for a running time of $O(mT^{\lfloor 3k/2 \rfloor})$, and this dominates the other steps, as $k \geq 2$ since $|S_e| = 2$ for every edge e incident to a leaf of $\mathcal{B}$. $\qquad\square$
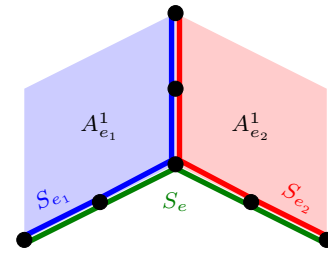

Fig. 6 Relation between S_e, S_{e_1}, and S_{e_2}

Having presented the core dynamic program, we now turn to the surrounding problems:

Finding an optimal branch decomposition If $\mathrm{bw}(G) \leq k$, then there is a linear-time algorithm computing a branch decomposition of width $\leq k$ (Bodlaender & Thilikos, 1997).

Computing an optimal timetable By additional bookkeeping, we can not only compute the minimum weighted slack, but also a periodic timetable realizing this slack.

2-edge-connectedness It is clear from the description of T-PESP-OPTIMALITY that the problem can be solved on each weakly connected component of G individually. Moreover, if one of these components is not 2-edge-connected, then the optimal periodic slack of any bridge will be zero (Borndörfer et al., 2019, §3.2). Hence, one can safely assume w.l.o.g. that G is 2-edge-connected. Note that this assumption implies $\mathrm{bw}(G) \geq 2$.

Fixing If π is a feasible periodic timetable and $d \in \mathbb{R}$, then the timetable π' defined by $\pi_i' := [\pi_i + d]_T$ for all $i \in V(G)$ is feasible as well and produces the same periodic slack. In particular, in Algorithm 5.3, for all $e \in E$, one can choose an event $i \in S_e$ and then fix $\pi_i = 0$. Thus, Algorithm 5.3 can be adapted to run in $O(mT^{\lfloor 3k/2 \rfloor - 1})$ time.

As a consequence, in conjunction with Lemma 2.7 and Theorem 4.14, we obtain

Theorem 5.6 *For $k \in \mathbb{N}$, there is an $O(mT^{\lfloor 3k/2 \rfloor - 1})$ algorithm solving T-PESP-OPTIMALITY on networks G with m activities and $\mathrm{bw}(G) \leq k$. In particular, if $k \geq 2$ is fixed, then the problems T-PESP-OPTIMALITY, T-PESP-FEASIBILITY and T-RPESP-OPTIMALITY are all weakly NP-complete.*

Bounding treewidth and carvingwidth

It is also possible to construct a tree-decomposition-based dynamic program for bounded treewidth. This tree decomposition version is expected to be asymptotically faster, but requires potentially more space. We refer to the Appendix for details.

We omit a carving-decomposition-based algorithm. Bounding the carvingwidth by k means that the branchwidth is bounded by $2k$ according to Theorem 4.16, and we can invoke Algorithm 5.3.

5.2 Cyclomatic number

We have already seen in Lemma 2.8 that fixing the number of events or the number of activities leads to FPT algorithms for T-PESP-OPTIMALITY. In this subsection, we discuss fixed-parameter tractability by cyclomatic number. The cyclomatic number is a common measure for the difficulty of PESP instances, as it counts the number of integral variables used in a cycle-based mixed integer programming formulation (Borndörfer et al., 2019).

Definition 5.7 Let G be a graph on n vertices, m edges and c weakly connected components. The *cyclomatic number* of G is defined as $\mu(G) := m - n + c$.

The following seems to be folklore:

Lemma 5.8 *Let G be a graph. Then,* $\mathrm{tw}(G) \leq \mu(G) + 1$.

Proof We give a proof by induction on $\mu(G)$. If $\mu(G) = 0$, then G is a forest and therefore $\mathrm{tw}(G) = 1$.

Now, let G be a graph with $\mu(G) > 0$. Then, G contains a cycle and hence an edge e such that $G' := G \setminus \{e\}$ has the same number of connected components as G, and $\mu(G') = \mu(G) - 1$. By induction hypothesis, $\mathrm{tw}(G') \leq \mu(G') + 1 = \mu(G)$, so that we find a tree decomposition $(\mathcal{T}, \mathcal{X})$ of G' with maximum bag size at most $\mu(G) + 1$. If there is a bag containing both endpoints of e, then $(\mathcal{T}, \mathcal{X})$ is also a valid tree decomposition for G, and so $\mathrm{tw}(G) \leq \mu(G)$. Otherwise, let i be an endpoint of e and add i to each bag of $(\mathcal{T}, \mathcal{X})$. This is a tree decomposition for G of width $\mu(G) + 1$, so that $\mathrm{tw}(G) \leq \mu(G) + 1$. □

While it is true that treewidth and branchwidth can be bounded in terms of each other (see Theorem 4.12), this does not hold for treewidth and cyclomatic number:

Lemma 5.9 *For $k \geq 2$, there is a class $\mathcal{C}_k$ of simple connected graphs such that $\mathrm{tw}(G) \leq k$ holds for all $G \in \mathcal{C}_k$, but for any $N \in \mathbb{N}$, there is a graph $G \in \mathcal{C}$ with $\mu(G) \geq N$.*

Proof Let $\mathcal{C}_k$ be the class of graphs G built from a finite disjoint union of cliques of size k with vertex sets $V_1, \ldots, V_r$ together with one additional vertex v joined to each vertex from each clique. Let $\mathcal{T}$ be a path on the vertices $\{1, \ldots, r\}$, and set $X_i := V_i \cup \{v\}$, $i = 1, \ldots, r$. Then, $(\mathcal{T}, \mathcal{X})$ is a tree decomposition of G and, as $|X_i| = k + 1$, we have $\mathrm{tw}(G) \leq k$. The cyclomatic number of G is given by

$$\mu(G) = \left(r \cdot \frac{k(k-1)}{2} + rk \right) - (rk + 1) + 1 = r \cdot \frac{k(k-1)}{2},$$

and for $k \geq 2$, this goes to infinity as $r \to \infty$. □

Lemma 5.10 *On networks where no vertex has degree 2, T-PESP-OPTIMALITY is FPT when parameterized by the cyclomatic number.*

Proof Let (G, T, ℓ, u, w) be a T-PESP-OPTIMALITY instance. We can safely remove all $i \in V(G)$ with $\deg(i) = 1$, as in any optimal solution, the incident activity must have periodic slack 0. Hence, we can assume that G has minimum degree 3. By the Handshaking lemma,

$$2m = \sum_{i \in V(G)} \deg(i) \geq 3n,$$

and hence

$$\mu = m - n + c \geq \frac{n}{2} + 1,$$

so that $n \leq 2\mu - 2$. This means that fixing μ provides a fixed bound on the number n of events, and we conclude by Lemma 2.8. □

Contracting vertices of degree 2 is a preprocessing technique that is applied to large PESP instances, for example, in Goerigk and Liebchen (2017) and Borndörfer et al. (2019).

Theorem 5.11 *T-PESP-FEASIBILITY is FPT when parameterized by the cyclomatic number.*

Proof Let (G, T, ℓ, u) be a T-PESP-FEASIBILITY instance. Remove all events of degree 1 from G, as this does neither affect feasibility nor alter the cyclomatic number. Now, all degree 2 vertices of G are arranged on (undirected) paths between two vertices of degree ≥ 3. Consider such a path from s to t with $\deg(s), \deg(t) \geq 3$, forward activities $a_1, \ldots, a_r$ and backward activities $b_1, \ldots, b_s$. Delete all intermediate vertices between s and t and insert a single activity a from s to t with

$$\ell_a := \sum_{i=1}^{r} \ell_{a_i} - \sum_{j=1}^{s} u_{b_j} \quad \text{and} \quad u_a := \sum_{i=1}^{r} u_{a_i} - \sum_{j=1}^{s} \ell_{b_j}.$$

Clearly, if x is a feasible tension with $\ell_{a_i} \leq x \leq u_{a_i}$ and $\ell_{b_j} \leq x_{b_j} \leq u_{b_j}$ for all i and j, then also $\ell_a \leq x_a \leq u_a$ with $x_a := \sum_{i=1}^{r} x_{a_i} - \sum_{j=1}^{s} x_{b_j}$. Conversely, any x_a with $\ell_a \leq x_a \leq u_a$ can be split into feasible tensions on all a_i and b_j. Thus, this transformation preserves feasibility. Moreover, contracting vertices of degree 2 does not change the cyclomatic number, so that we can assume that G has minimum degree 3. Invoke Lemma 5.10. □

Theorem 5.12 *T-PESP-OPTIMALITY is in XP when parameterized by the cyclomatic number.*

Proof Let F be a spanning forest of G and let $\gamma_1, \ldots, \gamma_\mu$ be its fundamental cycles, seen as incidence vectors $\{-1, 0, 1\}^{A(G)}$. Then (e.g., Nachtigall, 1998), $x \in \mathbb{R}^{A(G)}$ is a feasible periodic tension if and only if

$$\ell \leq x \leq u \quad \text{and} \quad \forall i \in \{1, \ldots, \mu\} : \gamma_i^t x \equiv 0 \mod T.$$

Decomposing $\gamma_i = \gamma_{i,+} - \gamma_{i,-}$ into positive resp. negative part $\gamma_{i,+}, \gamma_{i,-} \in \{0, 1\}^{A(G)}$, the modulo constraints are equivalent to

$$\forall i \in \{1, \ldots, \mu\} :$$
$$\gamma_i^t x = T z_i,$$
$$\left\lceil \frac{\gamma_{i,+}^t \ell - \gamma_{i,-}^t u}{T} \right\rceil \leq z_i \leq \left\lfloor \frac{\gamma_{i,+}^t u - \gamma_{i,-}^t \ell}{T} \right\rfloor,$$
$$z_i \in \mathbb{Z}.$$

These are the so-called *cycle inequalities* (Odijk, 1994). In particular, for each i, one has to check at most

$$\left\lfloor \frac{\gamma_{i,+}^t u - \gamma_{i,-}^t \ell}{T} \right\rfloor - \left\lceil \frac{\gamma_{i,+}^t \ell - \gamma_{i,-}^t u}{T} \right\rceil + 1$$
$$\leq \frac{(\gamma_{i,+} + \gamma_{i,-})^t (u - \ell)}{T} + 1$$

values for z_i. Since, as in the proof of Theorem 2.6, we can assume w.l.o.g. that $u - \ell < T$, we have the estimate

$$z_i \leq |\{a \in A(G) : \gamma_{i,a} \neq 0\}| + 1 \leq n + 1,$$

as the γ_i are simple cycles and hence contain at most n vertices.

The description of the polynomial-time algorithm is as follows: Enumerate all $O((n+1)^\mu)$ integral vectors $(z_1, \ldots, z_\mu)$ satisfying the cycle inequalities and solve the problem

$$\text{Minimize} \qquad w^t x$$
$$\text{subject to} \qquad \ell \leq x \leq u,$$
$$\gamma_i^t x = T z_i, \qquad \text{for all } i \in \{1, \ldots, \mu\}.$$

This is a minimum cost network tension problem and can be solved in polynomial time by network flow approaches (Hadjiat & Maurras, 1997; Nachtigall & Opitz, 2008). Alternatively, the above minimization problem can be solved by linear programming. □

It remains open whether T- PESP- OPTIMALITY can be solved with an FPT w.r.t. the cyclomatic number.

5.3 Cyclomatic number and diameter

The main obstacle for a FPT for T- PESP- OPTIMALITY is that the cyclomatic number does not bound the number of vertices. However, if, for example, one additionally fixes the diameter, then also T- PESP- OPTIMALITY becomes fixed-parameter tractable:

Corollary 5.13 T- PESP- OPTIMALITY *is* FPT *when parameterized by cyclomatic number and diameter.*

Proof We adapt the proof of Theorem 5.12. In our final estimate of z_i, the number of activities contained in γ_i can be bounded from above by $2d$ if d denotes the diameter of the graph. □

A by-product of Corollary 5.13 is that T- PESP- OPTIMALITY is also fixed-parameter tractable when parameterized by cyclomatic number and vertex cover number.

Finally, we want to remark that fixing both the cyclomatic number and the diameter does *not* bound the number of vertices:

Lemma 5.14 *For any $k \in \mathbb{N}$, there is an infinite class of simple connected graphs of diameter at most 2 and cyclomatic number at most k.*

Proof For $r \geq k$, let G be a star graph on r leaves. Connect k distinct pairs of leaves by an edge. Then, $\mu(G) = (k + r) - (r + 1) + 1 = k$ and G has diameter 2. □

6 Structure of realistic event-activity networks

In this section, we discuss the size of the so far discussed graph parameters on realistic periodic timetabling instances. We consider networks with a special structure based on line networks. This structure is the direct outcome of a typical modeling process (Nachtigall, 1998; Liebchen & Möhring, 2007; Schöbel, 2017; Pätzold et al., 2017). For example, the railway networks in the benchmarking library *PESPlib* (Goerigk, 2012) are found as subgraphs of networks with this structure. For this type of networks, we give lower and upper bounds on the branchwidth in terms of the underlying line network. We use this theoretical result to compute bounds on the branchwidth of the smallest PESPlib instance R1L1.

6.1 Line-based event-activity networks

Public transportation systems of cities, but also railway services, are typically organized in *lines*.

Definition 6.1 A *line network* $(N, \mathcal{L})$ is a directed multigraph N, together with a set $\mathcal{L}$ of directed walks on G such

that the arc set $A(N)$ is the disjoint union of $A(\ell)$ over all lines $\ell \in \mathcal{L}$.

Line networks reflect the maps which public transport companies offer for passenger information, displaying stations and lines. Depending on the precise application, lines may also constitute non-simple paths or contain cycles (e.g., London's Circle Line or Berlin's Ringbahn). In the context of line planning, we interpret a line network as a frequency-expanded line plan, i.e., some lines might have the same vertex sequence. Given a line network $(N, \mathcal{L})$, construct an event-activity network G as follows:

1. For each line $\ell \in \mathcal{L}$ and each arc $ij \in A(\ell)$, create a *departure event* (i, ℓ, dep) and an *arrival event* (j, ℓ, arr), and connect these by a *driving activity* $((i, \ell, \text{dep}), (j, \ell, \text{arr}))$.
2. For each vertex $i \in V(N)$ and each line $\ell \in \mathcal{L}$, add a *dwelling activity* $((i, \ell, \text{arr}), (i, \ell, \text{dep}))$ if both events exist.
3. For each vertex $i \in V(N)$ and each pair (ℓ_1, ℓ_2) of distinct lines, add a *transfer activity*
$((i, \ell_1, \text{arr}), (i, \ell_2, \text{dep}))$ if both events exist.

Definition 6.2 An event-activity network G is *line-based* if it arises from a line network $(N, \mathcal{L})$ by the above construction. Shortly, G is *based* on $(N, \mathcal{L})$.

Denote by $\deg^+(i)$ resp. $\deg^-(i)$ the number of outgoing resp. ingoing arcs at i. We summarize some straightforward structural properties of line-based networks in the following lemma:

Lemma 6.3 *Let G be based on $(N, \mathcal{L})$.*

1. *G is bipartite, the parts being the departure and arrival events, respectively.*
2. *Every departure event has a unique outgoing activity, and every arrival event has a unique ingoing activity. In both cases, these are driving activities.*
3. *The driving activities in G form a perfect matching in G.*
4. *Deleting the driving activities from G and undirecting the arcs results in the disjoint union of complete bipartite graphs $K_{\deg^+(i),\deg^-(i)}$ over $i \in V(N)$.*

Remark 6.4 Our definition of line-based networks disregards *headway activities*, which are typically employed to model different line frequencies or safety distances. However, the railway instances found in the PESPlib do not feature this particular kind of activities. (See §6.3 for the analysis of R1L1.)

6.2 Branchwidth of line-based networks

To give bounds on the branchwidth of line-based event-activity networks, we start with a well-known result on the branchwidth of minors:

Theorem 6.5 (Robertson & Seymour, 1991, 4.1) *If G is a graph and H is a minor of G, then $\mathrm{bw}(H) \le \mathrm{bw}(G)$.*

By Lemma 6.3, this implies that if G is based on $(N, \mathcal{L})$, then

$$\mathrm{bw}(G) \ge \max_{i \in V(N)} \mathrm{bw}(K_{\deg^+(i),\deg^-(i)}).$$

As we did not manage to find a reference in the literature for the branchwidth of complete bipartite graphs, we give a proof here:

Lemma 6.6 *The complete bipartite graph $K_{p,q}$ has branchwidth $\min(p, q)$.*

Proof Assume $p \le q$. Let P and Q denote the two parts, $|P| = p$, $|Q| = q$. The vertex separator associated with any neighborhood $\delta(w)$ for $w \in Q$ is given by P. Moreover, if $E \subsetneq \delta(w)$ is a proper subset, then the cardinality of the corresponding vertex separator is $|E| + 1 \le p$. Take any ternary tree with q leaves labeled with the vertices in Q. Then, replace each leaf w by any ternary tree with p leaves, labeled by the edges in $\delta(w)$. The result is a branch decomposition of $K_{p,q}$ of width p. This shows $\mathrm{bw}(K_{p,q}) \le p$.

To show that $\mathrm{bw}(K_{p,q}) \ge p$, we make use of *tangles*. That is, if we can find a collection $\mathcal{T}$ of subsets of $E(K_{p,q})$ such that

1. for each $A \in \mathcal{T}$, the size of the corresponding vertex separator is at most $p - 1$,
2. for each $A \subseteq E(K_{p,q})$ inducing a separator of size $\le p - 1$, either A or its complement is in $\mathcal{T}$,
3. for any three sets $A_1, A_2, A_3 \in \mathcal{T}$, their union is not $E(K_{p,q})$,
4. for each $A \in \mathcal{T}$, the subgraph induced by A does not contain all vertices of $K_{p,q}$,

then $\mathrm{bw}(K_{p,q}) \ge p$ or $p \le 2$ holds (Robertson & Seymour, 1991, 4.3). Clearly, $\mathrm{bw}(K_{1,q}) = 1$, as these are star graphs, and $\mathrm{bw}(K_{2,q}) = 2$, as these are series–parallel and contain cycles. Hence, suppose $p \ge 3$ and define

$$\mathcal{T} := \{A \subseteq E(K_{p,q}) \mid \mathrm{sep}(A) \le p - 1,$$
$$|V(K_{p,q}[A]) \cap P| \le p - 1,$$
$$|V(K_{p,q}[A]) \cap Q| \le p - 1\},$$

where $\mathrm{sep}(A)$ denotes the cardinality of the vertex separator associated with A, and $K_{p,q}[A]$ is the subgraph induced of $K_{p,q}$ by A. We check the above properties:

1. This is clear.
2. Let $A \subseteq E(K_{p,q})$ with $\text{sep}(A) \leq p - 1$ and suppose that $K_{p,q}[A]$ contains all vertices of P. Then, there must be a vertex $v \in P$ incident to some edge of A, but not contained in the separator. Hence, A must contain $\delta(v)$, and every vertex $w \in Q$ is incident to the edge $vw \in A$. Similarly, if A contains at least p vertices from Q, then it contains $\delta(w)$ for at least $|V(K_{p,q}[A]) \cap Q| - (p - 1)$ vertices $w \in Q$.

   This means if $\text{sep}(A) \leq p - 1$ and $A \notin \mathcal{T}$, then A contains $\delta(v)$ for some $v \in P$ and $\delta(w)$ for at least $q - (p - 1)$ vertices $w \in Q$. The subgraph induced by the complement of A hence does not contain v, and it also does not contain at least $q - (p - 1)$ vertices of Q. Therefore, the complement of A is in $\mathcal{T}$.
3. It follows by the argument in 2. that for each $A \in \mathcal{T}$, the vertex separator is given by $V(K_{p,q}[A])$. Now $K_{p,q}[A]$ is a simple bipartite graph on at most $p - 1$ vertices. It follows that $|A| \leq (p - 1)^2/4$. Now, if $A_1, A_2, A_3 \in \mathcal{T}$, the cardinality of their union is at most $3(p - 1)^2/4 < p^2 \leq pq = |E(K_{p,q})|$.
4. This follows as $|V(K_{p,q}[A])| \leq 2p - 2 < 2p \leq p + q = |V(K_{p,q})|$.

Hence, we conclude $\text{bw}(K_{p,q}) = p$. $\qquad\square$

Theorem 6.7 *Let G be a line-based event activity network, based on the line network $(N, \mathcal{L})$. Then,*

$$\max_{i \in V(N)} \min(\deg^+(i), \deg^-(i)) \leq \text{bw}(G) \leq \text{cw}(N),$$

and both bounds are sharp.

Recall from Sect. 4.1.4 that $\text{cw}(N)$ denotes the carvingwidth of N, and that

$$\text{cw}(N) \geq \max_{i \in V(N)} (\deg^+(i) + \deg^-(i)).$$

Speaking more intuitively, the carvingwidth is hence bounded by the maximum number of lines departing and arriving at a stop of the line network. In practice, line networks are often planar, and the carvingwidth of planar graphs can be computed in polynomial time by the ratcatcher algorithm (Seymour & Thomas, 1994).

Proof of Theorem 6.7 The lower bound follows from Lemma 6.3, Theorem 6.5, and Lemma 6.6. For $r \in \mathbb{N}$, let N_r be a graph on the vertex set $\{0, 1, \ldots, 2r\}$, and arcs $(i, 0)$ for $i \in \{1, \ldots, r\}$ and $(0, i)$ for $i \in \{r+1, \ldots, 2r\}$. The lines are given by $\mathcal{L} := \{(i, 0, i + r) \mid i \in \{1, \ldots, r\}\}$. The resulting line-based event-activity network G_r has the structure of a complete bipartite graph $K_{r,r}$ plus $2r$ activities connecting the $K_{r,r}$ with events of degree 1 each. It follows that

$$\text{bw}(G_r) = r = \max_{i \in V(N_r)} \min(\deg^+(i), \deg^-(i)).$$

As N_r is a star graph on $2r$ rays, its carvingwidth is easily determined to be $2r$. Hence, we found a family of graphs for which the lower bound is sharp, and the upper bound is larger than the lower bound.

Concerning the upper bound, we first partition the activities of G: For each $i \in V(N)$, let A_i denote the set of all activities incident to some departure event at i. Then, $\{A_i \mid i \in V(N)\}$ partitions $A(G)$ because of the bipartite structure of G. For $i \in V(N)$, let $(\mathcal{B}_i, \varphi_i)$ be a branch decomposition of the subgraph of G induced by A_i. Add a root b_i to each $\mathcal{B}_i$ by subdividing an arbitrary edge. Let $(\mathcal{C}, \psi)$ be an optimal carving decomposition of N. Attach to every leaf v of $\mathcal{C}$ the tree $B_{\psi(v)}$, identifying v with $b_{\psi(v)}$. This results in a branch decomposition $(\mathcal{B}, \varphi)$ of G.

Now, let $e \in E(\mathcal{B})$ and let $A(G) = A_e^1 \dot\cup A_e^2$ be the induced partition. If $e \in E(\mathcal{B}_i)$, then one of A_e^1, A_e^2 is contained in A_i, so that the vertex separator S_e has size at most $\deg^+(i) + \deg^-(i)$: S_e contains at most all $\deg^-(i)$ arrival events at i, and every other vertex in S_e must be either a departure event or the unique arrival event following a departure event. Observe that $\deg^+(i) + \deg^-(i) \leq \text{cw}(N)$ due to Theorem 4.16. In the other case that $e \in E(\mathcal{C})$, there is a subset $W_e \subseteq V(G)$ such that $A_e^1 = \bigcup_{i \in W_e} A_i$. Each vertex of the vertex separator S_e is hence an arrival event at some $i \in W_e$. The set S_e is in bijection to $\delta(W_e)$ by mapping (j, ℓ, arr) to $ij \in A(\ell) \subseteq A(N)$, where (i, ℓ, dep) is the unique driving activity entering (j, ℓ, arr). Hence, as $\mathcal{C}$ was chosen to be optimal, $|S_e| = |\delta(W_e)| \leq \text{cw}(N)$.

Finally, we show that the upper bound is sharp: For $r \in \mathbb{N}$, let N_r' be a directed simple cycle on r vertices. Then, $\text{cw}(N_r') = 2$. The event-activity network G_r' based on N_r' is then a directed simple cycle on $2r$ vertices and has branchwidth 2. On the other hand, the lower is not 2, as $\max_{i \in V(N_r')} \min(\deg^+(i), \deg^-(i)) = 1$. $\qquad\square$

6.3 Parameters of R1L1

6.3.1 An upper bound on branchwidth

The network G in its original shape does not satisfy the properties of Lemma 6.3. However, with small modifications, we can find a line network N such that G is a subgraph of an event-activity network based on N.

Transfer activities $a \in A(G)$ are in practice typically recognized by a large span $u_a - \ell_a$. As the period time is $T = 60$, we let $A_t := \{ij \in A(G) \mid u_a - \ell_a \geq 59\}$. We call any vertex i with $ij \in A_t$ for some j an arrival event, and analogously any vertex j with $ij \in A_t$ for some i is called a departure
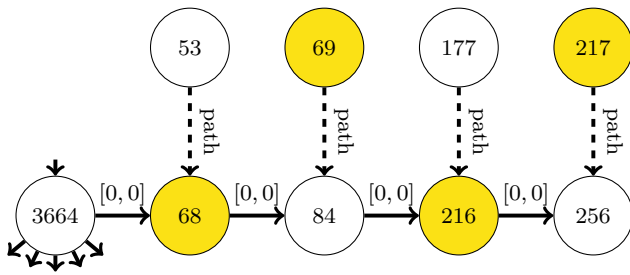
Fig. 7 Removed part of R1L1, events recognized as departures are marked yellow

event. This is well defined and extends to a bipartition of G into arrival resp. departure events.

Now, we interpret any activity from a departure to an arrival is called a driving activity. The set of driving activities is not quite a perfect matching in G: There are two arrival events (84 and 256) with two ingoing driving activities, and two arrival events (53 and 177) with no ingoing driving activity. This is due to four mysterious activities with lower and upper bound 0 breaking the structure at this particular spot; see Fig. 7. However, the vertices 53 to 84 and 177 to 256 can be removed from G by sequentially deleting vertices of degree 1. Since the remaining network has branchwidth at least two, removing vertices of degree 1 has no effect on branchwidth:

Lemma 6.8 *Let G be a connected graph, and let $v \in V(G)$ be a vertex of degree* 1. *If* $\mathrm{bw}(G \setminus \{v\}) \geq 2$, *then* $\mathrm{bw}(G) = \mathrm{bw}(G \setminus \{v\})$.

Proof By Theorem 6.5, $\mathrm{bw}(G \setminus \{v\}) \leq \mathrm{bw}(G)$. For the reverse inequality, consider a branch decomposition $(\mathcal{B}, \varphi)$ of $G \setminus \{v\}$. Since G is connected, v is adjacent to some vertex w of degree ≥ 2. Choose an edge $e \neq \{v, w\}$ incident with w, and replace the leaf of $\mathcal{B}$ corresponding to e by a node with the two children e and $\{v, w\}$. This is a branch decomposition of G. The size of the vertex separator corresponding to $\{v, w\}$ is 1, the size of the separator w.r.t. e is at most 2, and all other vertex separators remain unchanged. This shows $\mathrm{bw}(G) \leq \mathrm{bw}(G \setminus \{v\})$. □

With this adjustment, G has 3552 vertices and 6273 arcs. G satisfies now properties 1–3 of Lemma 6.3. Deleting the driving activities yields a disjoint union of bipartite graphs G_i, but they are not all complete. We define N now as the network obtained from G by contracting all these bipartite graphs G_i to a single vertex i. Then, G is a subgraph of an event-activity network based on N, choosing, for example, $\mathcal{L}$ as the set of all single-arc walks. By Theorems 6.5 and 6.7, $\mathrm{bw}(G) \leq \mathrm{cw}(N)$.

The network N obtained in this way is unfortunately not planar. The maximum degree in N is 62, so that $\mathrm{cw}(N) \geq 62$. To compute an upper bound, we first preprocess N by removing vertices of degree 2, as this does not alter carvingwidth

(Belmonte et al., 2013, Lemma 6). We use then the Kuratowski subgraph detection algorithm implemented in the Python package *networkx* (Hagberg et al., 2008) to recursively remove (multi-)arcs from N until the graph becomes planar. We prefer arcs of low multiplicity, and a sequence of 20 removals of simple arcs finally yields a planar graph N'. We implemented the ratcatcher method of Robertson and Seymour to compute $\mathrm{cw}(N') = 62$ and an optimal carving decomposition. As $V(N') = V(N)$, this is also a carving decomposition of N, but the width increases to 70. We hence conclude $\mathrm{cw}(N) \in [62, 70]$ and $\mathrm{bw}(G) \leq 70$.

6.3.2 A lower bound on branchwidth

Since G is only realized as a subgraph of an event-activity network based on N, we cannot invoke Theorem 6.7 directly. Of course, it remains true that $\mathrm{bw}(G)$ is at least the branchwidth of the (disconnected) subgraph G' obtained by deleting the driving activities. G' is reasonably small, but there seems to be no freely available software for exact branchwidth computations. However, there are treewidth codes, and we use the algorithm by Tamaki (2019), which has been implemented for the PACE 2017 challenge on exact treewidth computations (Dell et al., 2018). It turns out that $\mathrm{tw}(G') = 20$, hence $\mathrm{bw}(G) \geq 14$.

The largest of the components of G' is a bipartite graph with maximum part size 31. If this component were complete, then Theorem 6.7 would have predicted $\mathrm{bw}(G) \geq 31$.

To obtain a better bound on $\mathrm{bw}(G)$, we use balanced vertex separators:

Lemma 6.9 (Robertson & Seymour, 1995, 3.1) *Let G be a graph. Then, there is a vertex separator S with $|S| \leq \mathrm{bw}(G)$ and*

$$\max(|V^1|, |V^2|) \leq \frac{2}{3}|V(G)| - \frac{1}{2}|S|,$$

where $V(G) = V^1 \dot\cup V^2 \dot\cup S$, and no vertex in V^1 is adjacent to a vertex in V^2 and vice versa.

We now compute a minimum cardinality vertex separator subject to the balance constraint of Lemma 6.9 by plugging in a straightforward integer program into the CPLEX[1] 12.10 solver. We do not use the full network G as input, but take a smaller network that is obtained after standard preprocessing for T- PESP- Optimality instances (Borndörfer et al., 2019, §3.2). This network is a minor of G, so that we obtain a valid bound on the branchwidth. CPLEX finds a vertex separator of cardinality 58 and is able to solve the instance to optimality. We conclude $\mathrm{bw}(G) \geq 58$.

---

[1] https://www.ibm.com/analytics/cplex-optimizer.

6.3.3 Treewidth and carvingwidth

Since bw$(G) \in [58, 70]$, we obtain by Theorem 4.12 that tw$(G) \in [58, 104]$. As the maximum degree in R1L1 is 26, cw$(G) \in [29, 1820]$ by Theorem 4.16. Determining the exact treewidth of G turns out to be computationally infeasible. We instead use *TCS-Meiji* (Tamaki, 2019) and *FlowCutter* (Hamann & Strasser, 2018), the best two submissions of the PACE 2017 challenge on heuristic treewidth computations (Dell et al., 2018), with different random seeds to obtain a better upper bound on tw(G). The best bound we could find was tw$(G) \leq 97$.

6.3.4 Practical implications

The instance R1L1 has a period time of $T = 60$. It becomes clear from Table 1 that none of the algorithms presented in Sect. 5 can be applied for solving R1L1 in practice. For example, storing $T^{\text{bw}(G)-1} \geq 60^{57}$ table entries for the branch-decomposition-based algorithm 5.3 as 32-bit integers would require roughly $9 \cdot 10^{101}$ bytes of space.

7 Conclusion

The results of this paper underline that PESP is a notoriously hard problem. Although there are several primal heuristics available, the promising global approaches fail to compute provably optimal solutions: For example, mixed-integer programming formulations suffer from weak linear programming relaxations and transformations to Boolean satisfiability problems scale badly. It fits into this picture that exploiting structural parameters such as treewidth does not lead to a polynomial-time algorithm unless $P \neq NP$, and to no pseudo-fixed-parameter algorithm unless FPT $\neq$ W[1]. Moreover, the dynamic programs of Sect. 5 are only of theoretical interest. It is even unclear for tentatively large parameters as cyclomatic number and vertex cover number if T-PESP-OPTIMALITY becomes fixed-parameter tractable.

On the positive side, it has been demonstrated in Lindner and Liebchen (2019) that balanced edge separators lead to benefits when computing lower bounds of T-PESP-OPTIMALITY instances. We think that this should also carry over to vertex separators, and that good heuristic tree or branch decompositions may be useful as a source for separators in order to tackle PESP by a divide-and-conquer approach.

Appendix A Dynamic program from tree decompositions

In this appendix, we develop a tree decomposition analogue of Algorithm 5.3. Let $I = (G, T, \ell, u, w)$ be a T-PESP-OPTIMALITY instance. Let $(\mathcal{T}, \mathcal{X})$ be a tree decomposition. We assume that $(\mathcal{T}, \mathcal{X})$ is *rooted*, i.e., we pick an arbitrary leaf node $\tau \in V(\mathcal{T})$ and turn $\mathcal{T}$ into an arborescence where all edges point away from τ. Further suppose that the tree decomposition is *nice* (Kloks, 1994, Definition 13.1.15), i.e., each node $t \in V(\mathcal{T})$ with bag X_t fits into precisely one of the following categories:

– *Root*: $t = \tau$,
– *Leaf*: $t \neq \tau$, t has no children, and $|X_t| = 1$,
– *Introduce*: t has exactly one child u and one parent, $X_u \subseteq X_t$, and $|X_t| = |X_u| + 1$,
– *Forget*: t has exactly one child u and one parent, $X_t \subseteq X_u$, and $|X_t| = |X_u| - 1$,
– *Join*: t has exactly two children u_1, u_2 and one parent, and $X_t = X_{u_1} = X_{u_2}$.

Algorithm A.1 For each node $t \in V(\mathcal{T})$, we compute a $|X_t|$-dimensional table D_t having an entry for each $\pi \in \{0, \dots, T-1\}^{X_t}$. The table D_t is filled by the following dynamic program from the leaves to the root, depending on the role of the node t in the nice tree decomposition:

– If $t \neq \tau$ is a *leaf*, then $X_t = \{i\}$ for a single event $i \in V$. For all $\pi \in \{0, \dots, T-1\}$, set $D_t(\pi) := 0$.
– If t is an *introduce* vertex with child u, then $X_t = X_u \cup \{i\}$ for some event i. Put

$$D_t(\pi) := D_u(\pi|_{X_u}) + \sum_{ij \in A(G): \, j \in X_u} w_{ij} y_{ij} + \sum_{ji \in A(G): \, j \in X_u} w_{ji} y_{ji},$$

where, for $ij \in A(G)$,

$$
y_{ij} := \begin{cases}
[\pi_j - \pi_i - \ell_{ij}]_T \\
\quad \text{if } [\pi_j - \pi_i - \ell_{ij}]_T \leq u_{ij} - \ell_{ij}, \\
\infty \\
\quad \text{otherwise.}
\end{cases}
$$

– If t is a *forget* vertex with child u, then $X_t = X_u \setminus \{i\}$ for some event i. Set

$$
D_t(\pi) := \min\{D_u(\pi') \mid
$$
$$
\pi' \in \{0, \ldots, T-1\}^{X_u}, \pi'|_{X_t} = \pi\}.
$$

– If t is a *join* vertex with children u_1 and u_2, then $X_t = X_{u_1} = X_{u_2}$. If $D_{u_1}(\pi) < \infty$ and $D_{u_2}(\pi) < \infty$, define

$$
D_t(\pi) := D_{u_1}(\pi) + D_{u_2}(\pi)
$$
$$
- \sum_{ij \in A:\, i \in X_t,\, j \in X_t} w_{ij}[\pi_j - \pi_i - \ell_{ij}]_T,
$$

otherwise set $D_t(\pi) := \infty$.

– If $t = \tau$ is the *root*, then compute D_τ treating τ as a forget node. Return

$$
\min\{D_\tau(\pi) \mid \pi \in \{0, \ldots, T-1\}^{X_\tau}\}.
$$

For a node $t \in V(\mathcal{T})$, define G_t as the subnetwork of G induced by the events in the bag of X_t and the bags of all descendants of t in $\mathcal{T}$. Denote by I_t the corresponding T-PESP-Optimality subinstance of I.

Lemma A.2 *Let $t \in V(\mathcal{T})$ be a node of the tree decomposition and $\pi \in \{0, \ldots, T-1\}^{X_t}$.*

1. *If $S_t(\pi) < \infty$, then $S_t(\pi) = \mathrm{OPT}(I_t, X_t, \pi)$.*
2. *If $S_t(\pi) = \infty$, then there is no feasible periodic timetable on I_t coinciding with π on X_t.*

Proof Let $G_t = (V_t, A_t)$. If $t \neq \tau$ is a leaf, then $A_t = \emptyset$, and both statements are trivial.

Introducing an event i at t with child u means that

$$
V_t = V_u \,\dot\cup\, \{i\},
$$
$$
A_t = A_u \,\dot\cup\, (\{ij \in A : j \in X_u\} \,\dot\cup\, \{ji \in A : j \in X_u\}).
$$

The latter is a partition of the activities of G_t whose vertex separator is contained in X_u. Due to Lemma 5.2, $D_t(\pi)$ must hence equal $D_u(\pi|_{X_u})$ plus the minimum weighted slack of the subinstance associated with $A_t \setminus A_u$ fixing the timetable at X_t, and the latter is precisely given by the formula in Algorithm A.1.

When forgetting an event i at t with child u, then $G_t = G_u$. However, the timetable has to be fixed at X_t which contained one vertex less than X_u, so that we minimize over all table entries where the timetable restricted to X_u is the same.

If t is a join vertex with u_1 and u_2 as children, then $G_t = G_{u_1} \cup G_{u_2}$. We apply Lemma 5.2 to $A_t = A_{u_1} \,\dot\cup\, (A_{u_2} \setminus A_{u_1})$ and to $A_{u_2} = (A_{u_2} \setminus A_{u_1}) \,\dot\cup\, (A_{u_1} \cap A_{u_2})$. Note that by the connectedness property of the tree decomposition, any activity $A_{u_1} \cap A_{u_2}$ has both endpoints in $X_t = X_{u_1} = X_{u_2}$, so that X_t contains the vertex separators of both partitions. Now, the minimum weighted slack on G_t is the sum of the minimum weighted slacks on G_{u_1} and G_{u_2}, subtracting the minimum weighted slack of the activities that have been counted twice, i.e., $A_{u_1} \cap A_{u_2}$. Hence, the minimum weighted slack on G_t can be computed as described in Algorithm A.1. If fixing the timetable π on X_t yields an infeasible timetable, then either $D_{u_1}(\pi)$ or $D_{u_2}(\pi)$ must have been infinite, and vice versa.

Finally, if $t = \tau$ is the root, then τ has degree one and hence has a unique child u, so that we can treat it as a forget node.   □

Lemma A.3 *If the instance I is feasible, then Algorithm A.1 returns $\mathrm{OPT}(I)$, and otherwise ∞. Moreover, Algorithm A.1 runs in $O(|V(\mathcal{T})|T^{k+1})$ time, where $k := \max_{t \in V(\mathcal{T})} |X_t| - 1$.*

Proof By Lemma A.2, at the root τ, the table entry $D_\tau(\pi)$ is the minimum weighted slack on the subnetwork $G_\tau = G$ fixing the timetable at X_τ to π (or ∞). Minimizing over all entries in D_τ hence gives the minimum weighted slack of the T-PESP-Optimality instance (or detects infeasibility). The running time estimate is straightforward, as we need to fill each of the $|V(\mathcal{T})|$ tables with at most T^{k+1} entries and only employ summation and minimization.   □

Theorem A.4 *For $k \in \mathbb{N}$, there is an $O(nT^k)$ algorithm solving T-PESP-Optimality on event-activity networks G with n events and $\mathrm{tw}(G) \leq k$.*

Proof By Bodlaender (1996), if $\mathrm{tw}(G) \leq k$, then a tree decomposition with $O(n)$ nodes realizing width $\mathrm{tw}(G)$ can be found in $O(f(k) \cdot n)$ time. This can be transformed into a nice tree decomposition on $O(n)$ nodes within $O(n)$ time (Kloks, 1994, Lemma 13.1.3). Applying Lemma A.3 provides an $O(nT^{k+1})$ algorithm. Using the same fixing strategy as for Theorem 5.6, we obtain an $O(nT^k)$ algorithm.   □

Remark A.5 The branch-decomposition-based algorithm of Theorem 5.6 has an asymptotic running time of $O(mT^{\lfloor 3\,\mathrm{bw}(G)/2 \rfloor - 1})$, and the tree-decomposition-based one runs in $O(nT^{\mathrm{tw}(G)})$ time; see Theorem A.4. By Theorem 4.12, if G happens to be 2-edge-connected, then

$$
nT^{\mathrm{tw}(G)} \leq mT^{\lfloor 3\,\mathrm{bw}(G)/2 \rfloor - 1},
$$

so that the tree-decomposition-based algorithm can be expected to be asymptotically superior.

Remark A.6 In terms of memory, Algorithm 5.3 with the fixing strategy needs to store at most $T^{\mathrm{bw}(G)-1}$ table entries per edge, whereas Algorithm A.1 (with fixing) stores at most $T^{\mathrm{tw}(G)-1}$ entries per node. In both cases, at most 3 tables need to be stored at the same time, as any node in one of the decompositions has at most 2 children. Since $\mathrm{bw}(G) - 1 \leq \mathrm{tw}(G)$ if $\mathrm{bw}(G) \geq 2$, the branch-decomposition-based method potentially requires less space.

References

Arnborg, S., & Proskurowski, A. (1989). Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics, 23*(1), 11–24.

Belmonte, R., & van't Hof, P. (2013). Characterizing graphs of small carving-width. *Discrete Applied Mathematics, 161*(13), 1888–1893.

Bodlaender, H. L. (1996). A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing, 25*(6), 1305–1317.

Bodlaender, H. L., & Thilikos, D. M. (1997). Constructive linear time algorithms for branchwidth. In P. Degano, R. Gorrieri, & A. Marchetti-Spaccamela (Eds.), *Automata, languages and programming* (pp. 627–637). Berlin: Springer.

Bodlaender, H. L., & van Antwerpen-de Fluiter, B. (2001). Parallel algorithms for series parallel graphs and graphs with treewidth two. *Algorithmica, 29*, 534–559.

Borndörfer, R., Lindner, N., & Roth, S. (2019). A concurrent approach to the periodic event scheduling problem. *Journal of Rail Transport Planning & Management*, 100–175.

Cygan, M., Fomin, F. V., Kowalik, Ł, Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., & Saurabh, S. (2015). *Parameterized algorithms*. Cham: Springer.

Dell, H., Komusiewicz, C., Talmon, N., & Weller, M. (2018). The PACE 2017 parameterized algorithms and computational experiments challenge: The second iteration. In D. Lokshtanov & N. Nishimura (Eds.), *12th international symposium on parameterized and exact computation (IPEC 2017), volume 89 of Leibniz international proceedings in informatics (LIPIcs), Dagstuhl* (pp. 30:1–30:12). Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Downey, R. G., & Fellows, M. R. (1999). *Parameterized complexity. Monographs in computer science*. New York: Springer.

Eppstein, D. (2018). The effect of planarization on width. *Journal of Graph Algorithms and Applications, 22*(3), 461–481.

Erdős, P., Rubin, A. L., & Taylor, H. (1980). Choosability in graphs. In *Combinatorics, graph theory and computing, proceedings of the west coast conference, Arcata/California, 1979* (pp. 125–157).

Fellows, M. R., Fomin, F. V., Lokshtanov, D., Rosamond, F., Saurabh, S., Szeider, S., & Thomassen, C. (2011). On the complexity of some colorful problems parameterized by treewidth. *Information and Computation, 209*(2), 143–153.

Fiala, J., Golovach, P. A., & Kratochvíl, J. (2011). Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science, 412*(23), 2513–2523. Theory and Applications of Models of Computation (TAMC 2009).

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability*. San Francisco: W. H. Freeman and Company.

Goerigk, M. (2012). PESPlib—A benchmark library for periodic event scheduling. http://num.math.uni-goettingen.de/m.goerigk/pesplib/.

Goerigk, M. & Liebchen, C. (2017). An improved algorithm for the periodic timetabling problem. In *ATMOS, volume 59 of OASICS* (pp. 12:1–12:14). Schloss Dagstuhl—Leibniz-Zentrum für Informatik.

Goerigk, M., Schachtebeck, M., & Schöbel, A. (2013). Evaluating line concepts using travel times and robustness. *Public Transport, 5*, 267–284.

Hadjiat, M., & Maurras, J. F. (1997). A strongly polynomial algorithm for the minimum cost tension problem. *Discrete Mathematics, 165–166*, 377–394. Graphs and Combinatorics.

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. In Varoquaux, G., Vaught, T., & Millman, J. (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15), Pasadena, CA, USA.

Hamann, M., & Strasser, B. (2018). Graph bisection with pareto optimization. *Journal of Experimental Algorithmics* **23**

Karp, R. M. (1972). *Reducibility among combinatorial problems* (pp. 85–103). Boston: Springer.

Kloks, T. (1994). *Treewidth, computations and approximations. Lecture Notes in Computer Science* (Vol. 842). New York: Springer.

Liebchen, C. (2006). *Periodic timetable optimization in public transport*. PhD thesis, Technische Universität Berlin.

Liebchen, C. & Möhring, R. H. (2007). The modeling power of the periodic event scheduling problem: Railway timetables—And beyond. In *Algorithmic methods for railway optimization* (pp. 3–40). Springer.

Lindner, N., & Liebchen, C. (2019). New perspectives on PESP: T-partitions and separators. In V. Cacchiani & A. Marchetti-Spaccamela (Eds.), *19th Symposium on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS 2019), volume 75 of OpenAccess series in informatics (OASIcs)* (pp. 2:1–2:18). Dagstuhl, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Mnich, M., & van Bevern, R. (2018). Parameterized complexity of machine scheduling: 15 open problems. *Computers & Operations Research, 100*, 254–261.

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

Nachtigall, K. (1998). Periodic network optimization and fixed interval timetables. Habilitation thesis, Universität Hildesheim.

Nachtigall, K., & Opitz, J. (2008). Solving periodic timetable optimisation problems by modulo simplex calculations. In M. Fischetti & P. Widmayer (Eds.), *8th workshop on algorithmic approaches for transportation modeling, optimization, and systems (ATMOS'08), volume 9 of OpenAccess Series in Informatics (OASIcs)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Dagstuhl, Germany.

Nestoridis, N. V., & Thilikos, D. M. (2014). Square roots of minor closed graph classes. *Discrete Applied Mathematics, 168*, 34–39. Fifth Workshop on Graph Classes, Optimization, and Width Parameters, Daejeon, Korea, October 2011.

Odijk, M. A. (1994). Construction of periodic timetables, part 1: A cutting plane algorithm. Technical Report 94-61, TU Delft.

Pätzold, J., Schiewe, A., Schiewe, P., & Schöbel, A. (2017). Look-ahead approaches for integrated planning in public transportation. In *ATMOS, volume 59 of OASICS* (pp. 17:1–17:16). Schloss Dagstuhl—Leibniz-Zentrum für Informatik.

Pätzold, J., & Schöbel, A. (2016). A matching approach for periodic timetabling. of OpenAccess Series in Informatics (OASICs) (pp. 1:1–1:15). Dagstuhl, GermanyIn M. Goerigk & R. Werneck (Eds.), *16th workshop on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS 2016)* (Vol. 54, pp. 2190–6807). ISSN: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Robertson, N., & Seymour, P. (1984). Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B, 36*(1), 49–64.

Robertson, N., & Seymour, P. (1991). Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B, 52*(2), 153–190.

Robertson, N., & Seymour, P. (1995). Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B, 63*(1), 65–110.

Schrijver, A. (1986). *Theory of linear and integer programming*. New York: Wiley.

Schöbel, A. (2017). An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research Part C: Emerging Technologies, 74*, 348–365.

Serafini, P., & Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics, 2*(4), 550–581.

Seymour, P., & Thomas, R. (1994). Call routing and the ratcatcher. *Combinatorica, 14*(2), 217–241.

Tamaki, H. (2019). Positive-instance driven dynamic programming for treewidth. *Journal of Combinatorial Optimization, 37*(4), 1283–1311.

van Heuven van Staereling, I. (2018). Tree decomposition methods for the periodic event scheduling problem. In R. Borndörfer & S. Storandt (Eds.), *18th workshop on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS 2018), volume 65 of OpenAccess Series in Informatics (OASIcs)* (pp. 6:1–6:13). Dagstuhl. Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Vizing, V. G. (1976). Vertex colorings with given colors. *Diskretnyi Analiz, 29*, 3–10.