Periodic Timetabling with Cyclic Order Constraints

Enrico Bortoletto
Zuse Institute Berlin, Germany

Niels Lindner
Freie Universität Berlin, Germany

Berenike Masing
Zuse Institute Berlin, Germany

Abstract

Periodic timetabling for highly utilized railway networks is a demanding challenge. We formulate an infrastructure-aware extension of the Periodic Event Scheduling Problem (PESP) by requiring that not only events, but also activities using the same infrastructure must be separated by a minimum headway time. This extended problem can be modeled as a mixed-integer program by adding constraints on the sum of periodic tensions along certain cycles, so that it shares some structural properties with standard PESP. We further refine this problem by fixing cyclic orders at each infrastructure element. Although the computational complexity remains unchanged, the mixed-integer programming model then becomes much smaller. Furthermore, we also discuss how to find a minimal subset of infrastructure elements whose cyclic order already prescribes the order for the remaining parts of the network, and how cyclic order information can be modeled in a mixed-integer programming context. In practice, we evaluate the impact of cyclic orders on a real-world instance on the S-Bahn Berlin network, which turns out to be computationally fruitful.

2012 ACM Subject Classification Applied computing → Transportation; Mathematics of computing → Permutations and combinations; Mathematics of computing → Combinatorial optimization

Keywords and phrases Periodic Timetabling, Railway Timetabling, Periodic Event Scheduling Problem, Cyclic Orders, Mixed-Integer Programming

Digital Object Identifier 10.4230/OASIcs.ATMOS.2023.7

Funding *Enrico Bortoletto*: Funded within the Research Campus MODAL, funded by the German Federal Ministry of Education and Research (BMBF) (fund number 05M20ZBM).

Acknowledgements We thank DB Netz AG for providing real-world data and timetabling parameters for the S-Bahn Berlin network.

1 Introduction

Any public transportation network revolves around its timetable. A timetable is not only central for passengers to arrange their journeys, but also in the typical planning process of public transport (see, e.g., [9]), the timetable serves as a base for cost-sensitive planning steps such as vehicle and crew scheduling. It is therefore indispensable for the success of a public transportation system to operate a carefully designed timetable.

Timetabling for railway networks is a particularly demanding task, since an operationally feasible timetable must guarantee a high level of safety: Two trains must always be separated by a sufficient spatial and temporal distance. In the classical railway safety logic, the railway infrastructure is divided into block sections, and at any point in time, each block section can be occupied by at most one train. In recent years, the demand for trains has been increasing, and it is likely to grow further, given the major role that railway transport is supposed to attain in the future. However, infrastructure capacities do not grow as fast. For example, from 1995 to 2022, the number of freight trains in Germany has almost doubled, the number of passenger trains has increased by roughly a third, whereas the size of the network shrank by 12% [16]. This boosts the importance of modeling safety constraints with high precision in order to not waste optimization potential.

There have been several successful approaches in mathematical optimization of railway timetables [9], but these models are typically aperiodic. A large quantity of railway networks, especially suburban networks, are however operated with a periodic timetable, where trips repeat with a certain period time T. Mathematically, periodic timetable optimization can be expressed in terms of the Periodic Event Scheduling Problem (PESP) [17]. There is a decent amount of literature on periodic timetabling using PESP (e.g., [14, 12, 15, 6, 7]), but the safety considerations typically remain on a very coarse level. For example, *headway activities* can separate two events, e.g., two departures of two trains on the same track, by at least a certain minimum headway time [7]. This approach is however only workable when dwelling and turnaround times of trains are extremely small, or neglected entirely. In fact, classical headway activities alone cannot resolve what is called the *track occupation problem* in the recent paper [10]. For example, when a train occupies a track from minute 0 to 10 for turnaround, a second train might arrive at the same track at minute 5 and leave at minute 15. All events are separated by at least 5 minutes of headway time, so that this timetable would be feasible in the standard PESP model, although it is in fact operationally infeasible. To our knowledge, there is only little literature where periodic timetabling is combined with a proper infrastructure-derived modeling of safety constraints (e.g., [3]).

We try to close this gap by introducing *Infrastructure-Aware* PESP: In addition to a PESP instance on an event-activity network G, we are given a set of infrastructure elements that we can think of as block sections, and each activity in G is associated to at most one such infrastructure element. We demand that any pair of distinct activities associated to the same infrastructure element e must be separated by a minimum headway time $h_e \geq 0$. We then formulate a mixed-integer programming model for Infrastructure-Aware PESP using constraints described in [10] that resolve the track occupation problem.

Not unexpectedly, solving *Infrastructure-Aware* PESP is challenging: PESP alone is an NP-hard optimization problem [17], and even medium-sized instances have withstood attempts to solve them to optimality. For example, none of the instances of the benchmark library PESPlib [4] have been solved to optimality, even though a variety of algorithms is available [14, 13, 5, 1, 2]. It is in the nature of safety constraints that they affect pairs of events or activities, so that they contribute a major part of the problem size. However, in highly utilized networks, we have the following intuition: Fixing the timetable on parts that are operating close to capacity limits should have far-reaching consequences on the less crowded parts of the network. We will however not fix a specific timetable, but rather a *cyclic order* of activities associated to a common infrastructure element. For example, the S-Bahn Berlin network has several block sections that are used by as much as 7 trains within the period time of 20 minutes, while a minimum headway time of 2.5 minutes between two succeeding trains is desired. In particular, fixing the order of the trains on that block section leaves only little degrees of freedom for a timetable. Since we are considering periodic timetables, we do not consider total orders, but cyclic orders, i.e., we consider the orders (a_0, a_1, a_2), (a_1, a_2, a_0), (a_2, a_0, a_1) of three activities a_0, a_1, a_2 as equivalent, but different to (a_0, a_2, a_1). We then define *Infrastructure-Aware Fixed-Cycle-Order* PESP, where we prescribe a specific local cyclic order of the activities on each infrastructure element. On a realistic instance, it is probable that cyclic orders of close-by infrastructure elements are related or even must necessarily be the same, so that we also investigate algorithmic methods to capture the mutual compatibility of those local cyclic orders.

As a practical use case for our theoretical machinery, we evaluate Infrastructure-Aware PESP and the impact of orders on a real-world instance comprising the full S-Bahn Berlin network. It turns out that fixing cyclic orders has significant positive impact on performance in practice, although our additions maintain the computational complexity of PESP. We furthermore evaluate various methods to enhance Infrastructure-Aware PESP by information on local cyclic orders and their compatibility with each other.

In Section 2 we recall the basics of PESP. We introduce Infrastructure-Aware PESP and investigate a few theoretical properties in Section 3.1. Cyclic orders enter the picture in Section 3.2, and we describe how to work with them algorithmically in Section 3.3. Moving forward, we dedicate Section 4.1 to detailing the practical characteristics of the S-Bahn Berlin scenario that we use for testing. After analyzing cyclic orders on this instance in Section 4.2, we finally present in Section 4.3 the results and interpretations of our computational experiments. Section 5 ends the paper with our ideas for further research.

2 The Periodic Event Scheduling Problem

The *Periodic Event Scheduling Problem* (PESP) [17] is the usual mathematical model for optimizing periodic timetables in public transport. It has been discussed in numerous works, and we here very briefly recapitulate its main contents and formulations. An instance of the problem is given as a tuple (G, T, ℓ, u, w), comprising a directed graph G with $|V(G)| = n$ and $|A(G)| = m$, whose nodes are *events* and arcs are *activities*, a *period time* $T \in \mathbb{N}$, vectors $\ell \in \mathbb{R}^{A(G)}$ and $u \in \mathbb{R}^{A(G)}$ of lower and upper bounds on the arcs, respectively, and an arc-weight vector $w \in \mathbb{R}^{A(G)}_{\geq 0}$.

▶ **Definition 1** ([17]). *Given an instance (G, T, ℓ, u, w) as above, the* Periodic Event Scheduling Problem (PESP) *is to find a* periodic timetable $\pi \in \mathbb{R}^{V(G)}$ *and a* periodic tension $x \in \mathbb{R}^{A(G)}$ *such that*

a) $\pi_j - \pi_i \equiv x_a \bmod T$ *for all* $a = (i, j) \in A(G)$,

b) $\ell \leq x \leq u$,

c) $w^\top x$ *is minimum,*

or to decide that no such π and x exist.

If π is a periodic timetable, then a corresponding periodic tension is given by setting $x_a := [\pi_j - \pi_i - \ell_a]_T + \ell_a$ for all $a = (i, j) \in A(G)$, where $[\cdot]_T$ denotes the modulo T operator with values in $[0, T)$. Conversely, a periodic timetable can be recovered from a periodic tension by a graph traversal (see, e.g., [6, Theorem 9.8]).

We assume that ℓ and u are integral, so that by [14] the feasibility of a PESP instance implies the existence of an integral optimal solution. Moreover, we require that G contains no loops and that $0 \leq \ell \leq T - 1$ and $0 \leq u - \ell \leq T - 1$; this can always be achieved by preprocessing [6].

In the context of railway timetabling, events typically model arrivals or departures of trains at stations. Activities represent, e.g., driving between two adjacent stations, dwelling or turning at a station, or passenger transfers. Moreover, headway activities can be used to guarantee minimum distances between two events; we will discuss the modeling of safety constraints in more detail in Section 3.1. The weights w often estimate the number of passengers using a specific activity, so that $w^\top x$ can be interpreted as the total travel time of all passengers. Alternatively, the weights can be used to minimize the number of vehicles. We refer to [7] for further modeling aspects of PESP.

Several mixed-integer programming formulations for PESP are known [6]. We focus on the cycle-based formulation, which relies on the cycles of an integral cycle basis B of G [8]:

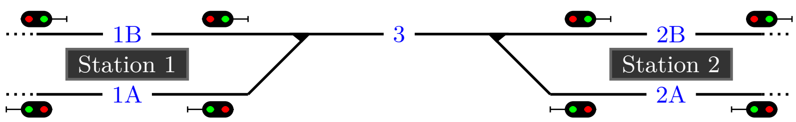
$$\begin{array}{lrll} \text{Minimize} & \displaystyle\sum_{a\in A(G)} w_a x_a & & \\ \hline \text{s.t.} & \displaystyle\sum_{a\in A(G)} \gamma_a x_a = T z_\gamma & \gamma \in B & \\ & \ell_a \leq x_a \leq u_a & a \in A(G) & \\ & z_\gamma \in \mathbb{Z} & \gamma \in B & \end{array} \tag{1}$$

3 The Infrastructure-Aware Periodic Event Scheduling Problem

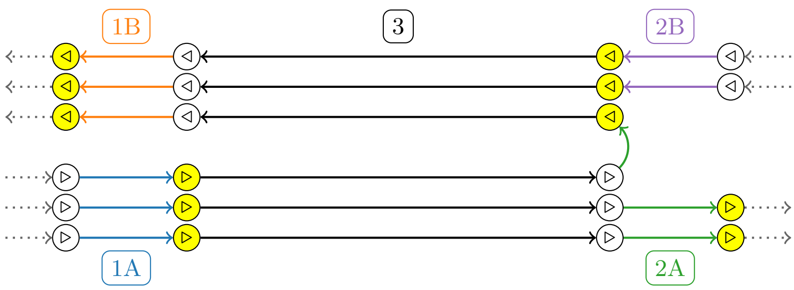
3.1 Infrastructure Awareness

Having railway timetabling in mind, we will be working with a special version of PESP that is *"infrastructure-aware"*. Along with a PESP instance (G, T, ℓ, u, w) we also have an *infrastructure map* $\eta\colon \mathcal{A} \to E$, where $\mathcal{A} \subseteq A(G)$, and E is a set of *infrastructure elements*. For each $e \in E$, we define $A_e := \eta^{-1}(e)$, i.e., the set of arcs that share the same infrastructure element e, and thus $\mathcal{A} = \bigcup_{e\in E} A_e$.

In railway terms, we think of the infrastructure elements as block sections, so that no two trains can occupy the same block section at the same time. The set A_e consists of those driving, dwelling or turnaround activities that share the common infrastructure element e. Of course, G might contain, e.g., passenger-related activities such as transfers, that do not need to be associated to an infrastructure element, and this is why $\mathcal{A}$ is only required to be a subset of $A(G)$. An exemplary railway infrastructure and event-activity network, illustrating the sets E and A_e, is depicted in Figure 1.



(a) A sample railway infrastructure. Station 1 and 2 have one platform with two tracks each, the section between Station 1 and 2 is single-track. As set E of infrastructure elements, we consider five block sections labeled with corresponding tracks: $E = \{1A, 1B, 2A, 2B, 3\}$.



(b) A mesoscopic event-activity network G for three lines operating on the infrastructure depicted in Figure 1a. Yellow vertices are departure events, white vertices are arrival events, and the arrows indicate the direction. Two lines pass through Station 1 and 2 in both directions, while a third line is turning on track 2A. We associate distinct colors to the infrastructure elements $e \in E$, and the activities in the set A_e are all colored with the color of e. For a periodic timetable to be operationally feasible, it is necessary that activities of the same color do not overlap in time.

Figure 1 An interpretation of Infrastructure-Aware PESP in the context of railway timetabling.

The goal is to find a solution to a given PESP instance such that two distinct activities $a_1 = (i_1, j_1)$ and $a_2 = (i_2, j_2)$ mapping to same infrastructure element $\eta(a_1) = \eta(a_2) = e$ are never scheduled to temporally overlap, but instead are separated by a minimum headway time $h_e \geq 0$ in the following sense (see also Figure 2):

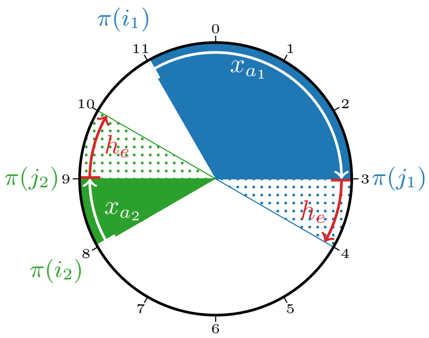


Figure 2 A visualization of two scheduled activities $a_1 = (i_1, j_1), a_2 = (i_2, j_2) \in A_e$ for some $e \in E$ on a clock, $T = 12$. Definition 2 requires that the distance $[\pi(i_2) - \pi(i_1)]_T$ must be at least x_{a_1} (filled blue sector) $+ h_e$ (dotted blue sector).

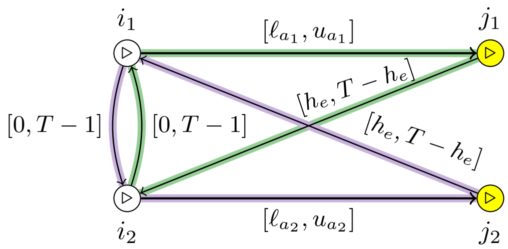


Figure 3 The Q3 formulation for the pairs (a_1, a_2) and (a_2, a_1), where $a_1 = (i_1, j_1), a_2 = (i_2, j_2) \in A_e$, $a_1 \neq a_2$, introduces two directed 3-cycles $q(a_1, a_2)$ (green) and $q(a_2, a_1)$ (purple). The Q3 constraints state that the periodic tension along each of these cycles sums up to T. As shown in [10], the Q3 constraints are equivalent to the activity separation constraints (2).

▶ **Definition 2.** *Let (G, T, ℓ, u, w) be a* PESP *instance, let $\eta\colon \mathcal{A} \to E$ be an infrastructure map, and let $h \in \mathbb{R}^E_{\geq 0}$. The* Infrastructure-Aware PESP *is to find a periodic timetable π with a corresponding tension x that optimally solve* PESP *on (G, T, ℓ, u, w), subject to the activity separation constraints*

$$[\pi_{i_2} - \pi_{i_1}]_T \geq x_{a_1} + h_e \tag{2}$$

for all $e \in E$ and all $a_1 = (i_1, j_1), a_2 = (i_2, j_2) \in A_e := \eta^{-1}(e)$ with $a_1 \neq a_2$, or to decide that no such solution exists.

▶ **Remark 3.** *To avoid the degeneracy that arises when $x_{a_1} = h_e = 0$ in* (2)*, we will from now on work with a* positivity assumption*: We require that for each $e \in E$ that $h_e > 0$ or that $\ell_a > 0$ holds for all $a \in A_e$.*

In words, the constraints (2) state that the activity a_2 cannot start before a_1 has finished and an additional time of h_e has passed. The constraints hence do not only separate *events* as standard headway activities do, but they also separate *activities*, which is necessary, e.g., as soon as trains have comparatively long dwelling times on a track [10]. For reasons that will become apparent later, we do not model the activity separation constraints (2) directly, but we choose to use the equivalent "Q3" formulation introduced in [10]. To do so (see also Figure 3), for any pair of distinct arcs $a_1 = (i_1, j_1)$ and $a_2 = (i_2, j_2)$ in the same set A_e we add a headway arc $a^{\mathrm{I}} = (j_1, i_2)$ with bounds $[h_e, T - h_e]$, as well as a headway arc $a^{\mathrm{II}} = (i_2, i_1)$ with bounds $[0, T-1]$, thereby creating a directed 3-cycle $q(a_1, a_2)$ on the arcs $a_1, a^{\mathrm{I}}, a^{\mathrm{II}}$. All such auxiliary arcs have weight 0. Let us denote as G^h the digraph of the original instance augmented with all necessary headway arcs, and let B^h be an integral cycle basis of G^h.

Then the following is a mixed-integer programming model for Infrastructure-Aware PESP:

$$\begin{array}{llr}
\text{Minimize} & \displaystyle\sum_{a \in A(G^h)} w_a x_a & \\
\hline
\text{s.t.} & \displaystyle\sum_{a \in A(G^h)} \gamma_a x_a = T z_\gamma & \gamma \in B^h \\
& \ell_a \leq x_a \leq u_a & a \in A(G^h) \\
& z_\gamma \in \mathbb{Z} & \gamma \in B^h \\
\hline
\text{(Q3 constraints)} & \displaystyle\sum_{a \in q(a_1,a_2)} x_a = T & e \in E, a_1, a_2 \in A_e, a_1 \neq a_2
\end{array} \tag{3}$$

Note that we express the Q3 constraints in terms of periodic tensions rather than of periodic offset variables as was done in [10].

▶ **Remark 4.** *The number of Q3 constraints in* (3) *is* $\sum_{e \in E} |A_e|(|A_e| - 1)$*. In particular, standard* PESP *arises when* $|A_e| \leq 1$ *for all* $e \in E$*. This also implies that* Infrastructure-Aware PESP *is NP-complete, because it belongs to NP, and for any* PESP *instance, setting* $E := A(G)$ *and* $\eta(a) := a$ *for all* $a \in A(G)$ *yields an equivalent* Infrastructure-Aware PESP *instance with* $|A_e| = 1$ *for all* $e \in E$*.*

The following polyhedral property is inherited from PESP:

▶ **Lemma 5.** *Consider a feasible instance for* Infrastructure-Aware PESP*. Then there is an optimal solution* (x, z) *to* (3) *and a spanning forest* F *of* G^h *such that* $x_a = \ell_a$ *or* $x_a = u_a$ *for all* $a \in A(F)$*.*

Proof. Let (x^*, z^*) be an optimal solution to (3). Then x^* is also optimal for the linear program that arises when fixing z to z^*. We can therefore assume that x^* is a vertex of the polytope

$$P := \{x \in \mathbb{R}^{A(G^h)} \mid \Gamma x = T z^*, Qx = T, \ell \leq x \leq u\}, \tag{4}$$

where Γ is the matrix with the vectors in B^h as rows, and Q is the matrix that has the incidence vectors of all Q3 constraint cycles $q(a_1, a_2)$ as rows. Since B^h is a cycle basis, Γ spans the cycle space of G^h, so that the row span of Q is contained in the row span of Γ. We therefore conclude that for the vertex x^*, the set of arcs $a \in A(G)$ for which one of the inequalities $\ell_a \leq x_a^*$ or $x_a^* \geq u_a$ is tight, must induce a maximal cycle-free subgraph of G^h, i.e., a spanning forest. ◀

We quickly note that the Q3 constraints and the positivity assumption (Remark 3) have implications on upper bounds.

▶ **Lemma 6.** *Let* (x, z) *be a feasible solution to* (3)*, and let* π *be a corresponding periodic timetable. For all* $e \in E$ *with* $|A_e| \geq 2$*, we have* $0 \leq x_a < T$ *for all* $a = (i, j) \in A_e$ *where* $x_a = [\pi_j - \pi_i]_T$*.*

Proof. Let $e \in E$ and $|A_e| \geq 2$, and let $a_1 = (i_1, j_1) \in A_e$.

We first suppose $h_e > 0$. Then a_1 is part of a Q3 constraint for a cycle $q(a_1, a_2)$ using a headway arc a^{I} in (3), and hence $0 \leq \ell_a \leq x_a \leq T - x_{a^{\mathrm{I}}} \leq T - h_e < T$. Since $[\pi_{j_1} - \pi_{i_1}]_T$ and x_{a_1} congruent modulo T and are both contained in $[0, T)$, they must be equal.

Now suppose that $h_e = 0$ and $\ell_{a_1} > 0$. Using (2), we find $x_{a_1} \leq [\pi_{i_2} - \pi_{i_1}]_T < T$, so that again $x_{a_1} = [\pi_{j_1} - \pi_{i_1}]_T$. ◀

With that, we can derive the following degree bounds.

▶ **Lemma 7.** *Let $(G, T, \ell, u, w, \eta, h)$ be a feasible instance for* Infrastructure-Aware PESP, *let $E' \subseteq E$ be any subset of infrastructure elements, and define $\mathcal{A}' := \bigcup_{e \in E'} A_e$.*

a) *If $h_e > 0$ for every $e \in E'$, then*

$$\forall i \in V(G): \quad \deg_{\mathcal{A}'}(i) \leq |E'|, \tag{5}$$

where $\deg_{\mathcal{A}'}(i)$ is the total degree of v in the subgraph of G with arc set $\mathcal{A}'$.

b) *Instead, if $h_e = 0$ for every $e \in E'$ and $\ell_a > 0$ for every $a \in \mathcal{A}'$, then*

$$\forall i \in V(G): \quad \max\left\{\delta^+_{\mathcal{A}'}(i), \delta^-_{\mathcal{A}'}(i)\right\} \leq |E'|, \tag{6}$$

where $\delta^+_{\mathcal{A}'}(i)$ and $\delta^-_{\mathcal{A}'}(i)$ are, respectively, out-degree and in-degree of i in the subgraph of G with arc set $\mathcal{A}'$.

Proof.

a) Suppose $h_e > 0$ for some $e \in E'$, and that there is a node i such that two arcs that are both in A_e are incident with i. If i is the tail of both arcs, then they are of the form $a_1 = (i, j)$ and $a_2 = (i, k)$. Using $h_e > 0$, $x_{a_1} \geq \ell_{a_1} \geq 0$ and (2), we have

$$0 < x_{a_1} + h_e \leq [\pi_i - \pi_i]_T = 0, \tag{7}$$

which cannot be. If i instead is the head of both arcs, then they are of the form $a_1 = (j, i)$ and $a_2 = (k, i)$, and by (2),

$$[\pi_k - \pi_j]_T \geq x_{a_1} + h_e \quad \text{and} \quad [\pi_j - \pi_k]_T \geq x_{a_2} + h_e. \tag{8}$$

Without loss of generality, we can assume $x_{a_1} \geq x_{a_2}$, and hence have k scheduled between i and j, but then, using Lemma 6,

$$[\pi_k - \pi_j]_T \geq x_{ji} + h_e = [\pi_i - \pi_j]_T + h_e = [\pi_i - \pi_k]_T + [\pi_k - \pi_j]_T + h_e > [\pi_k - \pi_j]_T, \tag{9}$$

which cannot be either. Finally, they could be of the form $a_1 = (j, i)$ and $a_2 = (i, k)$, and again by (2) and noting that $x_{a_1} = [\pi_i - \pi_j]_T$ due to Lemma 6,

$$x_{a_1} = [\pi_i - \pi_j]_T \geq x_{a_1} + h_e > x_{a_1} \tag{10}$$

which is also impossible. We conclude that if $h_e > 0$, then i can be incident with at most one arc of A_e. Consequently, if $h_e > 0$ for every $e \in E'$, then i is incident with at most $|E'|$ arcs that are contained in $\mathcal{A}'$.

b) Suppose instead that $h_e = 0$ for some $e \in E'$, as well as $\ell_a > 0$ for every $a \in \mathcal{A}'$. We observe that the contradiction (7) is still valid due to $x_{a_1} \geq \ell_{a_1} > 0$. Moreover, (9) holds because $[\pi_i - \pi_k]_T = x_{a_2} \geq \ell_{a_2} > 0$ by Lemma 6. We therefore conclude that at most one arc of A_e can enter i, and at most one arc of A_e can leave i. This implies b). ◀

These bounds have strong consequences on the structure and connectivity of G, if the instance is to be feasible at all. We consider, for example, the case when $\mathcal{A} = A(G)$, and $|E| = 1$.

▶ **Theorem 8.** *Consider an instance of* Infrastructure-Aware PESP *with infrastructure map $\eta: \mathcal{A} = A(G) \to \{e\}$ such that G is weakly connected, $|A(G)| \geq 1$. If the instance is feasible, then exactly one of the following holds:*

a) *$h_e > 0$ and G consists of a single arc.*

b) *$h_e = 0$ and G is a directed path.*

c) *$h_e = 0$ and G is a simple directed cycle.*

Proof. This is immediate from Lemma 7. ◀

▶ **Corollary 9.** Infrastructure-Aware PESP *is solvable in polynomial-time on instances with* $|E| = 1$ *and* $\mathcal{A} = |A(G)|$.

Proof. If there is only a single infrastructure element e, and $A_e = A(G)$, it is necessary for any feasible solution (x, z) of (3) to satisfy $\sum_{a \in A(G)} x_a \leq T$ in order to separate all arcs from each other. By Theorem 8, each weakly connected component of G is a path or a cycle. For each cycle γ, we then must have $\sum_{a \in \gamma} x_a = T$, because periodic tensions along a cycle sum up to an integer multiple of the period time, this multiple is at most T due to arc separation, but it is also larger than 0 because of the positivity assumption. We deduce that G is either a *single* cycle or a disjoint union of paths. In the latter case, solving Infrastructure-Aware PESP is trivial: Either $x^* = \ell$ is an optimal solution, or the instance is infeasible. In the single cycle case, Infrastructure-Aware PESP is solved by the simple linear program

$$\min\{w^\top x \mid \gamma^\top x = T, \ell \leq x \leq u\}, \tag{11}$$

observing that the condition $\gamma^\top x = T$ is both necessary and sufficient to guarantee non-overlapping of the activities along the cycle. ◀

3.2 Cyclic Orders

We have seen in Corollary 9 that directed cycles play a special role within Infrastructure-Aware PESP: In the trivial case that $|E| = 1$ and that G is a directed cycle, we could boil down the Q3 constraints to a single constraint, namely that the periodic tensions along the cycle sum up to T. This is due to the fact that the directed cycle fixes a cyclic ordering of its activities. Our aim is now to mimic this for an arbitrary number of infrastructure elements. To this end, we will fix for each $e \in E$ a cyclic order of the activities in A_e.

▶ **Definition 10** ([11]). *Let S be a finite set. Two total orders $(a_0, \dots, a_{n-1})$ and $(b_0, \dots, b_{n-1})$ on S are* cyclically equivalent *if there is an integer k such that for all $i \in \{0, \dots, n-1\}$ holds $a_i = b_{[i+k]_n}$. A* cyclic order *on S is an equivalence class Δ of total orders on S with respect to cyclic equivalence.*

We will denote both a total order and the cyclic order given by its equivalence class by $(a_0, \dots, a_{n-1})$, and apply this concept directly to PESP:

▶ **Definition 11.** *Let (G, T, ℓ, u, w) be a* PESP *instance with periodic timetable π. Suppose that $\Delta = (a_0, \dots, a_{n-1})$ is a cyclic order of a subset $S = \{a_0, \dots, a_{n-1}\} \subseteq A(G)$, where $a_k = (i_k, j_k)$ for all $k \in \{0, \dots, n-1\}$. We say that π* respects the cyclic order Δ *on S if $(\pi_{i_0}, \pi_{j_0}, \pi_{i_1}, \pi_{j_1}, \dots, \pi_{i_n}, \pi_{j_n})$ defines a cyclic order in the equivalence class of $\leq$.*

We return to Infrastructure-Aware PESP. Since in any feasible solution, for each infrastructure element $e \in E$, the activities do not overlap, any such solution gives rise to a cyclic order on A_e.

▶ **Theorem 12.** *Let $(G, T, \ell, u, w, \eta, h)$ be an* Infrastructure-Aware PESP *instance, let x be a feasible solution to* (3) *with corresponding periodic timetable π. Let $e \in E$ be an arbitrary infrastructure element, and write $A_e = \{a_0, \dots, a_{n-1}\}$ with $a_k = (i_k, j_k)$ for $k \in \{0, \dots, n-1\}$.*

a) *The timetable π respects some cyclic order on A_e.*

b) *The timetable π respects $\Delta_e = (a_0, \dots, a_{n-1})$ on A_e if and only if*

$$\sum_{a \in A_e} x_a + \sum_{k=0}^{n-1} [\pi_{i_{[k+1]_n}} - \pi_{j_k}]_T = T. \tag{12}$$

c) *The following constraint implies that π respects Δ_e and all Q3 constraints associated to e in* (3)*:*

$$\sum_{a \in Q(\Delta_e)} x_a = T \tag{13}$$

where $Q(\Delta_e)$ is the directed cycle in G^h consisting of the arcs in A_e and the headway arcs $a^{\mathrm{I}}_{a_k, a_{[k+1]_n}}$ between j_k and $i_{[k+1]_n}$ with bounds $[h_e, T - h_e]$ that have been added for the Q3 formulation in the cycle $q(a_k, a_{[k+1]_n})$, $k \in \{0, \dots, n-1\}$.

Proof. **a)** Any pair of activities is separated by h in the sense of Definition 2.

b) If π respects Δ_e, then there is a cyclic shift of $(\pi_{i_0}, \pi_{j_0}, \pi_{i_1}, \pi_{j_1}, \dots, \pi_{i_n}, \pi_{j_n})$ which is a total order with respect to $\leq$. This is equivalent to

$$[\pi_{j_0} - \pi_{i_0}]_T + [\pi_{i_1} - \pi_{j_0}]_T + \dots + [\pi_{j_{n-1}} - \pi_{i_{n-1}}]_T + [\pi_{i_0} - \pi_{j_{n-1}}]_T = T, \tag{14}$$

because the left-hand side is congruent to 0 modulo T, and $[\cdot]_T$ can in fact be omitted except at exactly one summand. Due to the positivity assumption, $[\pi_{j_k} - \pi_{i_k}]_T = x_{a_k}$ for all k, so that (12) is equivalent to (14).

c) We first note $x_{a^{\mathrm{I}}_{a_k, a_{[k+1]_n}}} = [\pi_{j_{[k+1]_n}} - \pi_{i_k}]_T$. Hence, if (13) holds, then (12) holds, and thus π respects Δ_e. Consider for $k \neq l$ a cycle $q(a_k, a_l)$ defining a Q3 constraint at $e \in E$. Since $(\pi_{i_k}, \pi_{j_k}, \pi_{i_l})$ is a subsequence of $(\pi_{i_0}, \pi_{j_0}, \pi_{i_1}, \pi_{j_1}, \dots, \pi_{i_n}, \pi_{j_n})$, which is a cyclic shift of a total order with respect to $\leq$,

$$\sum_{a \in q(a_k, a_l)} x_a = [\pi_{j_k} - \pi_{i_k}]_T + [\pi_{i_l} - \pi_{j_k}]_T + [\pi_{i_k} - \pi_{i_l}]_T = T. \tag{15}$$

◀

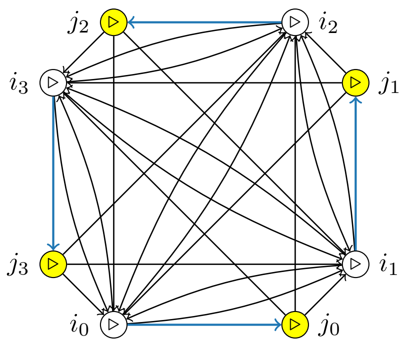
We now define a version of Infrastructure-Aware PESP, where cyclic orders at each infrastructure element are fixed.

▶ **Definition 13.** *Let $(G, T, \ell, u, w, \eta, h)$ be an instance of* Infrastructure-Aware PESP*, and let Δ_e be a set of cyclic order on A_e for each $e \in E$. The* Infrastructure-Aware Fixed-Cycle-Order PESP *is to find a solution to* Infrastructure-Aware PESP *that additionally respects Δ_e on A_e for all $e \in E$, or to decide that no such solution exists.*

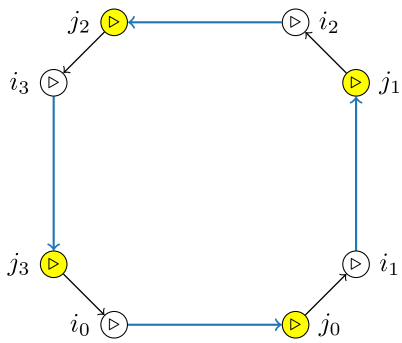
The Infrastructure-Aware Fixed-Cycle-Order PESP has to be treated with caution, because fixing cyclic orders beforehand will in general have severe impacts on feasibility and optimization potential. However, there are practical situations, where such information is known or can be propagated (see also Section 3.3).

Theorem 12 allows to formulate Infrastructure-Aware Fixed-Cycle-Order PESP as a mixed-integer linear program: We can prescribe a specific cyclic order at each infrastructure element $e \in E$ by adding the constraints (13) to (3). A very elegant consequence of Theorem 12 is that the $\sum_{e \in E} |A_e|(|A_e| - 1)$ Q3 constraints can then be discarded. Since then also the headway arcs of the form a^{II} used in the Q3 constraints lose their significance, they can be deleted as well, so that the model size drops considerably. Figure 4 visualizes this effect.

▶ **Remark 14.** *Several results carry over to the setting of fixed cyclic orders:* Infrastructure-Aware Fixed-Cycle-Order PESP *is NP-complete with the same reasoning as in Remark 4. If there is only one infrastructure element to which all arcs are associated, then* Infrastructure-Aware Fixed-Cycle-Order PESP *is polynomial-time-solvable. Furthermore, the spanning forest property Lemma 5 carries over to* Infrastructure-Aware Fixed-Cycle-Order PESP*.*



(a) When no cyclic order on A_e is fixed, then the Q3 constraints state that the periodic tension along each directed 3-cycle $q(a_k, a_l)$ for $k \neq l$ must sum up to T.



(b) When a cyclic order Δ_e on A_e is fixed, it suffices to require that the periodic tension along a single cycle, namely the directed Hamiltonian cycle that is induced by Δ_e, adds up to T. Here, $\Delta_e = (a_0, a_1, a_2, a_3)$.

Figure 4 Arcs in the Q3 formulation for Infrastructure-Aware PESP vs. (13) for Infrastructure-Aware Fixed-Cycle-Order PESP for $A_e = \{a_0, a_1, a_2, a_3\}$, $a_k = (i_k, j_k)$, $k \in \{0, 1, 2, 3\}$. Choosing a cyclic order Δ_e on A_e corresponds to choosing a directed Hamiltonian cycle Figure 4b in the digraph Figure 4a built by the union of the cycles $q(a_k, a_l)$ for $a_k, a_l \in A_e$, $k \neq l$.

3.3 Propagating Cyclic Orders and Chronological Constraints

The Infrastructure-Aware Fixed-Cycle-Order PESP requires formally to fix a cyclic order at each infrastructure element. This might be a tedious task not only due to the number of infrastructure elements, but also since the cyclic orders need to be compatible between "related" infrastructure elements. Such an information is often present in real-world scenarios, and we suggest two strategies to exploit this computationally.

3.3.1 Identifying Maximal Infrastructure Elements

Let $\mathcal{T}$ denote a set of trips and let $\tau : \mathcal{A} \to \mathcal{T}$ be a map whose restriction to each A_e is injective, i.e., no two arcs in the set A_e for a given infrastructure element e can be associated with the same trip. We call $\tau(A_e)$ the set of trips on e. We introduce a binary relation $\preceq$ on E by defining $e \preceq e'$ if and only if $\tau(A_e) \subseteq \tau(A_{e'})$ and *all* trips on e must necessarily appear in the same cyclic order on e'. That is, we want that Δ_e is a subsequence of $\Delta_{e'}$, when identifying arcs with their trips.

For example, if two branches of a railway network join, and there is no possibility of overtaking, then the order $\Delta_{e'}$ of the arcs in $A_{e'}$, i.e., of the trips $\tau(A_{e'})$, on the first common infrastructure element e' is already fixing the order Δ_e of the trips $\tau(A_e)$ on the last infrastructure element e on each branch before joining.

The relation $\preceq$ is a preorder on E. To prescribe a cycle ordering at each $e \in E$, it is hence enough to fix a cycle ordering at each maximal element of $\preceq$.

Algorithmically, we can construct a directed infrastructure graph H such that $(e, e') \in A(H)$ if and only if $e \preceq e'$. We then contract directed cycles in H, so that H becomes acyclic and $\preceq$ becomes a partial order. In practical terms, elements belonging to a directed cycle are associated with the same set of trips, and those must appear in the same cyclic order. The maximal elements of the partial order can then be identified with the sinks of H, i.e., the vertices with out-degree 0.

A real-world example of the resulting directed acyclic graph is given in Figure 5.

3.3.2 Chronological Constraints

It might be beneficial not to fix cyclic orders everywhere, but only at some infrastructure elements. Moreover, not all cyclic orders are equally good, for example, when regular patterns of trains are desired. We are therefore seeking to add the enforcing and compatibility of cyclic orders to the mixed-integer programming formulation of Infrastructure-Aware PESP.

To this end, at each $e \in E$, we introduce a binary variable $\sigma^e_\Delta\{0,1\}$ for each cyclic order Δ on A_e, and enforce Δ or not via the big-M constraints

$$\sum_{a \in Q(\Delta)} x_a \leq T\sigma^e_\Delta + T|A_e|(1 - \sigma^e_\Delta) \qquad e \in E, \Delta \text{ cyclic order on } A_e \tag{16}$$

$$\sum_{a \in Q(\Delta)} x_a \geq T\sigma^e_\Delta + 2T(1 - \sigma^e_\Delta) \qquad e \in E, \Delta \text{ cyclic order on } A_e \tag{17}$$

$$\sum_{\Delta \text{ cyclic order on } A_e} \sigma^e_\Delta = 1 \qquad e \in E \tag{18}$$

which are derived from (13). If $\sigma^e_\Delta = 1$, then (16) and (17) enforce Δ on A_e. Otherwise, if the order Δ is not respected, then $\sum_{a \in Q(\Delta)} x_a \neq T$, and due to the positivity assumption and the fact that $\sum_{a \in Q(\Delta)} x_a$ is an integral multiple of T, we must have $\sum_{a \in Q(\Delta)} x_a \geq 2T$. Note that (17) is redundant for feasible integer solutions, but it strengthens the linear programming relaxation. Note that the cycle $Q(\Delta)$ is composed of $|A_e|$ pairs (a, a^I) of arcs that are part of a q-cycle, so that $x_a + x_{a^I} \leq T$ by virtue of the Q3 constraints (3). In particular, we always have $\sum_{a \in Q(\Delta)} x_a \leq T|A_e|$.

The compatibility of orders among elements $e \preceq e'$ can be modeled by

$$\sigma^e_\Delta \leq \sum_{\substack{\Delta' \text{ cyclic order on } A_{e'} \\ \text{restricting to } \Delta \text{ on } A_e}} \sigma^{e'}_{\Delta'} \qquad e \in E, \Delta \text{ cyclic order on } A_e \tag{19}$$

$$\sum_{\substack{\Delta \text{ cyclic order on } A_e \\ \text{induced by } \Delta' \text{ on } A_{e'}}} \sigma^e_\Delta \leq \sigma^{e'}_{\Delta'} \qquad e \in E, \Delta' \text{ cyclic order on } A_e \tag{20}$$

Note that using the sum in (19) and (20) is justified by (18).

4 Computational Results

4.1 Instances

We evaluate the use of cyclic orders in a case study of two detailed real-world instances of Infrastructure-Aware PESP. Both instances comprise the full S-Bahn Berlin network, a suburban commuter rail network consisting of 16 lines, which is operated periodically with a period time of 20 minutes. Since the timetable is planned with a resolution of 0.1 minutes on a mesoscopic scale and we want to stick to integral bounds, we therefore consider $T = 200$. The (lower) bounds for driving, dwelling and turnaround activities are derived from the 2022 annual timetable. We further assume that driving activities are fixed, i.e., lower and upper bound coincide. The infrastructure information and the minimum headway times h_e are set according to the planning parameters at DB Netz AG, which is responsible for the S-Bahn Berlin timetable. The network contains several stretches where 6 or 7 trains ride per direction and 20 minutes, so that planning a conflict-free timetable is demanding. On the other hand, fixing a cyclic order on an infrastructure element with high usage is expected to largely limit the degree of freedom for timetabling the remaining parts of the network.

Our first instance `i1` does not consider transfer activities, because our data does not contain any information about passenger flows. The arc weights are simple: They are 2 for all arcs that are relevant for passengers, and 1 otherwise, e.g., for turnarounds. The rationale is that feasibility is a major issue, but there is still an incentive to minimize dwelling and turnaround times, with a priority on dwelling. Moreover, this approach is also suitable to minimize the required number of vehicles.

To make the case study a little more meaningful, we created a second instance `i2` with an artificial passenger flow. For each station, we counted the number of public transport trips, including subway, buses and trams, departing at that station within a typical peak hour, and use that number as a demand per station. We then simulate 100,000 passengers that pop up on a station, distributed according to the demand, and use the shortest route according to the annual timetable to their destination, which is sampled with the help of a gravity model. The second instance hence contains transfer activities, and the weights are chosen according to the number of passengers using the activity in question.

Some characteristics of both instances are summarized in Table 1.

Table 1 Characteristics of our two instances.

Instance type	# nodes	# total arcs	# headway arcs	# transfer arcs
`i1` (without transfers)	2412	8439	6027	0
`i2` (with transfers)	2412	9405	6027	966

4.2 Maximal Infrastructure Elements

A consequence of fixing the driving activities is that in most cases, it will be superfluous to add cyclic orders for driving activities, as they are implied by the ones for dwelling inequalities. However, there are exceptions, e.g., single-track sections.

It turns out that $\preceq$ as defined in Section 3.3.1 has 22 maximal elements out of 192 infrastructure elements, so that fixing a cyclic order at only 22 infrastructure elements suffices to prescribe a cyclic order at each infrastructure element. The poset induced by $\preceq$ is visualized in Figure 5.

4.3 Experiments

All our experiments were conducted on an Intel i7-9700K CPU with 32 GB RAM, using Gurobi 10. Preliminary runs used the standard MIP formulation of PESP presented in [17], which proved to be unreasonably slower than the cycle formulation even at solving trivial instances, and so all tests presented here use the cycle formulation, as in (1). A quite influential choice when using the cycle formulation is that of which cycle basis to use, and in this paper we used two options. For some tests, we used a strictly fundamental cycle basis arising from a `bfs`-tree, which we will denote as $B_{\texttt{bfs}}$. For other tests, we instead used a strictly fundamental cycle basis arising from a minimum span spanning tree, which we will denote as $B_{\texttt{span}}$.

First and foremost, we tested `i1`, modeled as seen in (3). Using $B_{\texttt{span}}$, a primal solution is found after 1 minute and 12 seconds, with a 17% gap, and the optimal solution is found after 20 minutes and 26 seconds, when also proven optimality is achieved. The optimal value is 3058. Instead, using $B_{\texttt{bfs}}$, no primal solution was found within 3 hours, with almost no dual bound to speak of either. Nonetheless $B_{\texttt{bfs}}$ proved to be quite good when many orders are fixed.

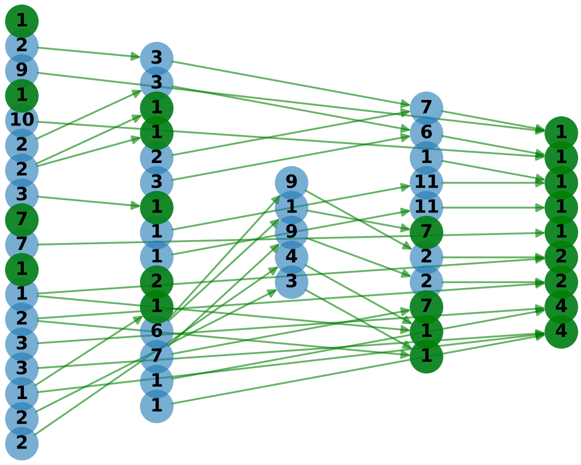


Figure 5 The directed acyclic graph induced by $\preceq$ for both instances has 22 sinks (green), which identify the maximal infrastructure elements as in Section 3.3.1. The vertex labels indicate the number of infrastructure elements that are equivalent w.r.t. $\preceq$. The five columns show from left to right the infrastructure elements used by 3 to 7 trains within 20 minutes, we omitted the ones with 2 trains or less.

Next, we tested if and how much solving speed would improve by fixing order information. To do so, we took the annual timetable of the S-Bahn and derived feasible cyclic orders to impose onto the activities A_e per each stationary infrastructure $e \in E$. Note that the objective value of the annual timetable is 5128 in `i1`, and 673759 in `i2`.

We then conducted tests with different levels of fixing and using both bases $B_{\texttt{bfs}}$ and $B_{\texttt{span}}$. For the transfer-less instance, results of these tests can be seen in Table 2. The same tests were conducted on `i2`, whose results can be found in Table 3. With the added transfers the instance is particularly harder to solve. In fact, without fixing any orders, a primal solution is found only after 51 minutes and 22 seconds, and at the mark of the hour the gap to dual bound is 64.2%, with objective value 483716.

Note that the improvements in objective value for `i1` and `i2` compared to the annual timetable have to be taken with a grain of salt: The minimum turnaround times in our model are quite low and can only be achieved with a second driver, which is practically feasible, but only in exceptional cases. Moreover, our gravity model might not reflect the actual passenger distribution.

Finally, now only using the cycle basis $B_{\texttt{span}}$ and `i1`, in Table 4 we show various test results that include the σ variables introduced in Section 3.3. The sets displayed in the "Test configuration" column indicate a list of the sizes of A_e's for which we added corresponding σ variables to the model. For each such σ variable we always include equations as seen in (16) and (18). Test configurations marked by the letter b, also include equations as seen in (17), bounding below. Test configurations marked by the letter l, also include equations as seen in (19) and (20), linking orders for compatibility. Finally, test configurations marked by the letter r are ones were we enforced regularity on the σ's there included, meaning any σ_Δ is there set to 0 if $\Delta \geq 4$ and Δ consecutively orders two activities of the same line. This applies only when a line is operated with higher frequency than once per 20 minutes, because in this case, it is not desirable that two trips of the same line are directly succeeding. It is important to note that, except for enforced regularity, all σ-constraints are merely descriptive of timetable behaviour, as they are all redundant with respect to the base model of Infrastructure-aware PESP. For that reason, it is entirely possible to use a continuous relaxation of the σ variables instead of proper binary variables, since it is entirely unnecessary to find perfectly integral values for all σ's. All tests showed that this is always very beneficial, and so all tests use continuous σ's.

4.4 Interpretation of Results

As expected, fixing order information greatly improves solving time, as seen in Table 2, although the cycle basis choice remains quite influential. In general it can be observed that the more orders are fixed, the more $B_{\texttt{bfs}}$ is faster than $B_{\texttt{span}}$, and vice versa. The former basis, by nature of the spanning tree from which it arises, is characterized by particularly short cycles (average of ∼4 arcs per cycle). The latter, instead, also by nature of the spanning tree from which it arises, is characterized by particularly long cycles (average of ∼93 arcs per cycle). Generally, we do not know enough about the performance of cycle bases in solving PESP, but preliminary tests, also using other cycle bases of intermediate average cycle length, seemed to confirm this inverse relationship, namely "short" bases being better with lots of order-fixing, and "long" bases being better in less constrained settings. Using a more meaningful objective value, that of `i2`, also the tests shown in Table 3 confirm the same pattern shown in the previous table. In fact, faster times in Table 2 are almost invariably matching to smaller optimality gaps in Table 3.

It is worth to note that the optimal value of each test varies, as fixing orders at different infrastructure elements constrains the problem differently. In that sense, it is then interesting to observe and compare how much closer to the global optimum (3058) some test configurations end up, sometimes with relatively little time increase, such as test $[i \neq 7]$.

As per Table 4, the main takeaway is that, indeed, including descriptive σ variables and constraints is of significant aid to solving time, as long as the size of the model does not excessively increase. This size increase is always driven by the presence of unreasonably many σ variables for all possible cyclic orders of large A_e's. In greater detail, we can say that constraints of the form (17), marked by b in the tests, seem to only hinder the solver, whereas it is harder to pass judgement on linking constraints, marked by l. Although detrimental when infrastructure with larger A_e's is involved, linking constraints seem to be of use when applied to infrastructure with small A_e's. This might be because of an amplification of the issues already created by the increasing size of the model. Another reason for that could be akin to what makes continuous σ's perform better than binary σ's, i.e., letting such descriptive constraints be less precise may allow better agility. Finally, we note that enforced regularity, marked by r, is powerful when infrastructure with larger A_e's is involved, which is of no surprise, since many cyclic orders of larger sets are irregular, and therefore many σ constraints would then be greatly simplified by forcing the indicator variable to 0.

5 Future Work

Given the high variance in performance with respect to the choice of cycle basis, it is tempting to investigate further bases, e.g., cycle bases that combine "long" cycles that correspond to activities used by lines, and "short" cycles such as the q-cycles in the Q3 formulation. Moreover, since the number of possible cyclic orders explodes in larger instances, it is natural to think about dynamic generation of σ-variables. Finally, given that fixing a cycle order boosts running times, we imagine that a heuristic, that optimizes first for a given cycle order and then modifies that order by local k-opt moves and optimizes again, could be beneficial to solve realistic and also larger Infrastructure-Aware PESP instances.

References

1 R. Borndörfer, N. Lindner, and S. Roth. A concurrent approach to the periodic event scheduling problem. *Journal of Rail Transport Planning & Management*, 15:100175, 2020. Best Papers of RailNorrköping 2019. doi:10.1016/j.jrtpm.2019.100175.

2 E. Bortoletto, N. Lindner, and B. Masing. Tropical Neighbourhood Search: A New Heuristic for Periodic Timetabling. In M. D'Emidio and N. Lindner, editors, *22nd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2022)*, volume 106 of *Open Access Series in Informatics (OASIcs)*, pages 3:1–3:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/OASIcs.ATMOS.2022.3.

3 F. Fuchs, A. Trivella, and F. Corman. Enhancing the interaction of railway timetabling and line planning with infrastructure awareness. *Transportation Research Part C: Emerging Technologies*, 142:103805, September 2022. doi:10.1016/j.trc.2022.103805.

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

5 P. Großmann, S. Hölldobler, N. Manthey, K. Nachtigall, J. Opitz, and P. Steinke. Solving Periodic Event Scheduling Problems with SAT. In H. Jiang, W. Ding, M. Ali, and X. Wu, editors, *Advanced Research in Applied Artificial Intelligence*, Lecture Notes in Computer Science, pages 166–175, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-31087-4_18.

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

7 C. Liebchen and R. H. Möhring. The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables — and Beyond. In F. Geraets, L. Kroon, A. Schoebel, D. Wagner, and C. D. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, Lecture Notes in Computer Science, pages 3–40, Berlin, Heidelberg, 2007. Springer. doi:10.1007/978-3-540-74247-0_1.

8 C. Liebchen and L. Peeters. Integral cycle bases for cyclic timetabling. *Discrete Optimization*, 6(1):98–109, February 2009. doi:10.1016/j.disopt.2008.09.003.

9 R. M. Lusby, J. Larsen, M. Ehrgott, and D. Ryan. Railway track allocation: models and methods. *OR Spectrum*, 33(4):843–883, October 2011. doi:10.1007/s00291-009-0189-0.

10 B. Masing, N. Lindner, and C. Liebchen. Periodic Timetabling with Integrated Track Choice for Railway Construction Sites. Technical Report 22-26, Zuse Institute Berlin, 2022. URL: https://nbn-resolving.org/urn:nbn:de:0297-zib-88626.

11 N. Megiddo. Partial and complete cyclic orders. *Bulletin of the American Mathematical Society*, 82(2):274–276, 1976.

12 K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables.* Habilitation Thesis, Universität Hildesheim, 1998.

13 K. Nachtigall and J. Opitz. Solving Periodic Timetable Optimisation Problems by Modulo Simplex Calculations. In M. Fischetti and P. Widmayer, editors, *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08)*, volume 9 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany, 2008. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/OASIcs.ATMOS.2008.1588.

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

15 L. Peeters. *Cyclic Railway Timetable Optimization.* PhD thesis, Erasmus Universiteit Rotterdam, January 2003.

16 Allianz pro Schiene e. V. Das Schienennetz in Deutschland, 2023. Retrieved on 08/07/2023. URL: https://www.allianz-pro-schiene.de/themen/infrastruktur/schienennetz/.

17 P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM J. Discret. Math.*, 2:550–581, 1989.

A Appendix – Tables

Table 2 Fixed order test on i1, with cycle basis $B_{\texttt{bfs}}$ in the white rows, and cycle basis $B_{\texttt{span}}$ in the gray rows. The "Test configuration" column indicates a list of the sizes of A_e's for which the order was fixed. For example, in test $[i \neq 5] = \{3, 4, 6, 7\}$ we fixed the cyclic orders for each and every $e \in E$ with $|A_e| \neq 5$, meaning all those of size in $\{3, 4, 6, 7\}$, and similarly for other rows. The time limit of each test was 15 minutes.

Test configuration	Time to primal (s)	Time to optimal (s)	Optimal value
$[i \geq 3] = \{3, 4, 5, 6, 7\}$	1	12	3967
$[i \geq 3] = \{3, 4, 5, 6, 7\}$	54	55	”
$[i \geq 4] = \{4, 5, 6, 7\}$	1	13	”
$[i \geq 4] = \{4, 5, 6, 7\}$	65	75	”
$[i \geq 5] = \{5, 6, 7\}$	11	66	3948
$[i \geq 5] = \{5, 6, 7\}$	87	100	”
$[i \geq 6] = \{6, 7\}$	34	194	”
$[i \geq 6] = \{6, 7\}$	144	179	”
$[i \geq 7] = \{7\}$	218	840	3661
$[i \geq 7] = \{7\}$	163	178	”
$[i \neq 3] = [i \geq 4]$	1	13	3967
$[i \neq 3] = [i \geq 4]$	65	75	”
$[i \neq 4] = \{3, 5, 6, 7\}$	11	30	3951
$[i \neq 4] = \{3, 5, 6, 7\}$	87	99	”
$[i \neq 5] = \{3, 4, 6, 7\}$	11	20	3967
$[i \neq 5] = \{3, 4, 6, 7\}$	64	72	”
$[i \neq 6] = \{3, 4, 5, 7\}$	40	80	3855
$[i \neq 6] = \{3, 4, 5, 7\}$	151	161	”
$[i \neq 7] = \{3, 4, 5, 6\}$	27	60	3351
$[i \neq 7] = \{3, 4, 5, 6\}$	29	68	”
$[i \leq 3] = \{3\}$	–	–	–
$[i \leq 3] = \{3\}$	92	555	3058
$[i \leq 4] = \{3, 4\}$	–	–	–
$[i \leq 4] = \{3, 4\}$	103	387	3175
$[i \leq 5] = \{3, 4, 5\}$	–	–	–
$[i \leq 5] = \{3, 4, 5\}$	126	319	3191
$[i \leq 6] = [i \neq 7]$	27	60	3351
$[i \leq 6] = [i \neq 7]$	27	60	”
$[i \leq 7] = [i \geq 3]$	1	12	3967
$[i \leq 7] = [i \geq 3]$	1	12	”

Table 3 Fixed order test on `i2`, with cycle basis $B_{\texttt{bfs}}$ in the gray rows, and cycle basis $B_{\texttt{span}}$ in the white rows. The time limit for each test was 15 minutes.

Test configuration	Time to primal (s)	Gap at 15' mark	Primal bound
$[i \geq 3] = \{3,4,5,6,7\}$	61	5.99%	482429
$[i \geq 3] = \{3,4,5,6,7\}$	139	35.4%	"
$[i \geq 4] = \{4,5,6,7\}$	70	6.50%	"
$[i \geq 4] = \{4,5,6,7\}$	156	36.5%	"
$[i \geq 5] = \{5,6,7\}$	274	8.41%	"
$[i \geq 5] = \{5,6,7\}$	136	37.6%	"
$[i \geq 6] = \{6,7\}$	124	9.67%	"
$[i \geq 6] = \{6,7\}$	394	38.7%	"
$[i \geq 7] = \{7\}$	731	18.5%	482885
$[i \geq 7] = \{7\}$	250	44.1%	482429
$[i \neq 3] = [i \geq 4]$	70	6.50%	482429
$[i \neq 3] = [i \geq 4]$	156	36.5%	"
$[i \neq 4] = \{3,5,6,7\}$	76	7.69%	"
$[i \neq 4] = \{3,5,6,7\}$	164	37.8%	"
$[i \neq 5] = \{3,4,6,7\}$	74	5.34%	"
$[i \neq 5] = \{3,4,6,7\}$	173	30.4%	484741
$[i \neq 6] = \{3,4,5,7\}$	341	9.33%	482429
$[i \neq 6] = \{3,4,5,7\}$	164	36.1%	"
$[i \neq 7] = \{3,4,5,6\}$	79	10.9%	"
$[i \neq 7] = \{3,4,5,6\}$	121	36.5%	"
$[i \leq 3] = \{3\}$	–	–	–
$[i \leq 3] = \{3\}$	–	–	–
$[i \leq 4] = \{3,4\}$	–	–	–
$[i \leq 4] = \{3,4\}$	669	51.8%	484007
$[i \leq 5] = \{3,4,5\}$	–	–	–
$[i \leq 5] = \{3,4,5\}$	133	40.3%	482429
$[i \leq 6] = [i \neq 7]$	79	10.9%	"
$[i \leq 6] = [i \neq 7]$	121	36.5%	"
$[i \leq 7] = [i \geq 3]$	61	5.99%	"
$[i \leq 7] = [i \geq 3]$	139	35.4%	"

Table 4 Tests with σ-constraints on `i1`, in various configurations, using cycle basis $B_{\texttt{span}}$. The "Number of rows" column indicates the number of rows in the model after presolving. Time values to optimality that improve on the baseline model of the first row are shown in bold. The time limit for each test was 1 hour.

Test configuration	Time to primal (s)	Time to optimal (s)	Number of rows
$\{\}$	72	1226	5745
$\{3\}$	68	**275**	5863
$\{4\}$	68	**200**	5964
$\{5\}$	76	**642**	6374
$\{6\}$	134	1693	12476
$\{7\}$	356	2673	17995
$\{3\}+b$	114	**1006**	5981
$\{4\}+b$	110	**482**	6168
$\{5\}+b$	144	**501**	6998
$\{6\}+b$	406	0.13% after 1h	19196
$\{7\}+b$	550	1674	30235
$\{4\}+r$	132	**261**	5951
$\{5\}+r$	60	**608**	6350
$\{6\}+r$	43	**835**	12476
$\{7\}+r$	149	1353	16746
$\{3,4\}$	114	**579**	6082
$\{3,5\}$	73	**571**	6492
$\{3,6\}$	213	1742	12594
$\{3,7\}$	167	**1209**	18112
$\{4,5\}$	180	**496**	6593
$\{4,6\}$	167	**681**	12695
$\{4,7\}$	333	1597	18214
$\{5,6\}$	99	**630**	13105
$\{5,7\}$	200	**625**	18624
$\{6,7\}$	240	1923	24726
$\{3,4\}+l$	190	**462**	6088
$\{3,5\}+l$	75	**550**	6492
$\{3,6\}+l$	140	1447	12598
$\{3,7\}+l$	295	**1221**	18118
$\{4,5\}+l$	113	**932**	6605
$\{4,6\}+l$	147	**670**	12725
$\{4,7\}+l$	392	2025	18232
$\{5,6\}+l$	121	**732**	13153
$\{5,7\}+l$	220	**1171**	18648
$\{6,7\}+l$	860	3212	25206
$\{3,4,5,6,7\}$	684	0.93% after 1h	25692
$\{3,4,5,6,7\}+l$	352	1.29% after 1h	26320
$\{3,4,5,6,7\}+r$	120	**1138**	24406
$\{3,4,5,6,7\}+l+r$	165	1967	24716
$\{3,4,5,6,7\}+b$	1354	1.65% after 1h	45598
$\{3,4,5,6,7\}+l+b$	3039	2.44% after 1h	46226
$\{3,4,5,6,7\}+r+b$	–	–	44313
$\{3,4,5,6,7\}+l+r+b$	3596	6.53% after 1h	44623