

Geometric Advances and Infrastructure Awareness in Periodic Timetabling

Dissertation

zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Enrico Bortoletto

Berlin, 2024

Supervisor: Prof. Dr. Ralf Borndörfer

Reviewers

First reviewer: Prof. Dr. Ralf Borndörfer

Second reviewer: Prof. Dr. Philine Schiewe

Date of defense: July 14th, 2025

Declaration of Authorship

Surname: Bortoletto
Name: Enrico

I declare to the Freie Universität Berlin that I have completed the submitted dissertation independently and without the use of sources and aids other than those indicated. The present thesis is free of plagiarism. I have marked as such all statements that are taken literally or in content from other writings. This dissertation has not been submitted in the same or similar form in any previous doctoral procedure.

I agree to have my thesis examined by a plagiarism examination software.

Furthermore, I hereby confirm that all co-authors of the articles underlying this thesis agree to the detailed attribution of the individual contributions at the end of each article.

Date

Enrico Bortoletto

Acknowledgements

Personal

I am honoured to begin this section thanking Prof. Dr. Ralf Borndörfer, whose guidance kicked-off my very first day at Freie Universität Berlin, during my Master's studies. If not for his lectures, his commitment to higher education, and his enthusiastic passion, I would not have even considered the doctoral path, let alone reach its end.

I also thank Dr. Niels Lindner, whose mentorship has been an invaluable source of daily wisdom and encouragement. Without his support, the contents of this work would have hardly found the required clarity and maturity. Alongside him, I also thank my co-authors Berenike Masing and Dr. Rolf N. van Lieshout, for their time and unwavering brilliance.

A special thank you goes to my colleagues and friends, at Zuse Institute Berlin and beyond, for the many moments and conversations we shared: experiences that not only shaped my interests and academic journey, but also lightened the hard days and brightened the good ones.

Finally, to Giulia, I dedicate all my love.

Funding and Data

This research was funded:

- within the Research Campus MODAL, funded by the German Federal Ministry of Education and Research (BMBF) (fund number 05M20ZBM),
- by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689).

Part of the work and results we present were made possible by data shared with us by DB InfraGO AG, whom we thank.

In conclusion, all of this work has been undertaken at Zuse Institute Berlin, at which I have made use of spaces, technical resources, and the expertise and care of its many employees. I thank them all.

Zusammenfassung

Das Periodic Event Scheduling Problem (PESP) dient als grundlegendes Modell für die Optimierung von Taktfahrplänen in öffentlichen Verkehr. Wir präsentieren Fortschritte im theoretischen Verständnis von PESP und den zugrundeliegenden Geometrien, und zeigen neue Modellierungs- und Optimierungsmethoden für Fahrplanung mit integrierten Infrastrukturaspekten in der Praxis.

Nach einer Formulierung von PESP als gemischt-ganzzahliges Programm untersuchen wir zunächst die geometrische Struktur des Problems. Der Raum der Taktfahrpläne zerfällt in eine Menge von Polytropen – tropische Polytope, die auch im traditionellen Sinne konvex sind – und sich um einen Torus wickeln. Im Gegensatz dazu, im Raum der „Cycle Offsets“, also dem Raum der ganzzahligen Hilfsvariablen, betrachten wir ein spezielles Zonotop, das kombinatorisch von großem Interesse ist. Wir stellen Verbindungen zwischen diesen beiden Geometrien her und wenden unsere neuen Erkenntnisse an, um eine neuartige Heuristik für PESP zu entwickeln, Tropical Neighborhood Search (**tns**). Diese testen und evaluieren wir ausgiebig und erzielen exzellente Rechenergebnisse.

Der zweite Teil der Dissertation legt den Fokus auf die Herausforderung, Infrastrukturbedingungen in die Taktfahrplanung zu integrieren. Wir entwickeln neue MIP-basierte Modellierungsansätze und zeigen, neben anderen Möglichkeiten, eine Konstruktion von relevanten Schnittebenen sowie Ansätze unter Verwendung von Zyklusordnungen und perfekten Matchings, um sicherzustellen, dass der Fahrplan auch auf der verfügbaren Infrastruktur sicher umgesetzt werden kann. Hierbei wird die Zuordnung der Fahrzeuge optimiert, sodass bisher unzureichend ausgeschöpfte Gleiskapazitäten ausgenutzt werden können. Die Umsetzbarkeit und praktische Relevanz unserer Modelle, Infrastructure-Aware PESP und dessen Erweiterungen, wird in mehreren Fallstudien anhand realer öffentlicher Verkehrsnetze demonstriert.

Abstract

The Periodic Event Scheduling Problem (PESP) serves as the foundational framework for optimizing periodic timetables in public transport. We here present advances in the theoretical understanding of PESP and its underlying geometries, as well as novel ways to model and optimize timetabling with integrated infrastructure aspects in practice.

First, having formulated PESP as a mixed-integer program, we study the geometric structure of the problem. In the periodic timetable space we find a collection of polytopes, tropical polytopes that are also traditionally convex, wrapped onto a torus. Instead, in the space of cycle offsets, which are auxiliary integer variables, we find a special zonotope, of high combinatorial interest. We establish connections between these two geometries, and employ these newfound insights to develop a novel local improvement heuristic for PESP, Tropical Neighbourhood Search (**tns**), which we extensively test and evaluate with excellent computational results.

In the second half of the dissertation, we focus on the challenge of properly integrating infrastructure constraints with periodic timetabling. We construct novel MIP-based modelling strategies, relevant cutting planes, and other approaches using cycle orders and perfect matchings, to ensure that found solutions are safely operable on the available infrastructure, while also optimizing the assignment of vehicles to ground resources, making full use of previously untapped track capacity. The viability and practical relevance of our models, Infrastructure-Aware PESP and its extensions, is showcased over multiple case studies on real-world public transport networks.

Contents

1	Introduction	17
1.1	Context and Motivation	19
1.2	The Periodic Event Scheduling Problem	21
1.3	Dissertation Overview and Contributions	23
1.3.1	Novel Geometric Insights	23
1.3.2	A Local Improvement Heuristic	24
1.3.3	Integrating Infrastructural Constraints	25
1.3.4	Optimising Infrastructure Use	25
1.3.5	Errata, Typos, and Formatting	26
2	The Tropical and Zonotopal Geometry of Periodic Timetables	29
2.1	Introduction	31
2.2	The Periodic Event Scheduling Problem	33
2.2.1	Problem Definition	33
2.2.2	The Space of Feasible Periodic Timetables	35
2.2.3	The Space of Cycle Offsets	36
2.3	The Tropical Tiling of the Periodic Timetable Space	40
2.3.1	Weighted Digraphs and Polytopes	41
2.3.2	Slicing a Cube and Tiling the Torus	45
2.3.3	Polytopes in the Periodic Event Scheduling Context	51
2.3.4	An application of the Tropical Neighbourhood	53
2.4	Cycle Offset Zonotopes	60
2.4.1	Cographic Zonotopes	61
2.4.2	Scaled Cographic Zonotopes and the Cycle Offset Zonotope	63
2.4.3	A Duality Between Torus Polytopes and Zonotope Tiles	66
2.4.4	Constructing Zonotopal Tilings	72
2.4.5	Approximating the minimum width of a cycle basis	78
2.5	Outlook	82

3	Tropical Neighbourhood Search: A New Heuristic for Periodic Timetabling	85
3.1	Introduction	87
3.2	Tropical Decomposition of Periodic Timetable Space	88
3.3	Tropical Neighbourhood Search	94
3.4	Implementation details	96
3.4.1	Preparing the <i>exploreList</i>	96
3.4.2	Sorting the <i>exploreList</i>	97
3.4.3	The <i>qualityFactor</i>	98
3.4.4	Subproblem Formulation: Arc Offsets vs. Cycle Offsets	98
3.4.5	Hashing Visited Polytropes	99
3.5	Computational Experiments and Results	99
3.5.1	Impact of Parameter Choices for tns	100
3.5.2	Contribution of tns in Comparison to Other Methods	102
3.5.3	New PESPlib Incumbents	104
3.6	Outlook	105
3.A	Appendix	106
4	Periodic Timetabling with Cyclic Order Constraints	111
4.1	Introduction	113
4.2	The Periodic Event Scheduling Problem	115
4.3	The Infrastructure-Aware Periodic Event Scheduling Problem	117
4.3.1	Infrastructure Awareness	117
4.3.2	Cyclic Orders	124
4.3.3	Propagating Cyclic Orders and Chronological Constraints	127
4.4	Computational Results	129
4.4.1	Instances	129
4.4.2	Maximal Infrastructure Elements	130
4.4.3	Experiments	130
4.4.4	Interpretation of Results	133
4.5	Future Work	134

4.A Appendix	135
5 Periodic Event Scheduling with Flexible Infrastructure Assignment	139
5.1 Introduction	141
5.2 Periodic Event Scheduling and Infrastructure Awareness	142
5.3 General Infrastructure Awareness with Flexible Infrastructure Maps	145
5.3.1 Problem Definition and Periodizability of Assignments	146
5.3.2 Pattern Functions and Conflict-Freeness	148
5.3.3 A MIP Formulation for IPESPA	151
5.4 Partitionable Infrastructure Maps	153
5.5 Experiments	160
5.5.1 Instances	160
5.5.2 IPESP Experiments	161
5.5.3 IPESPC Experiments	162
5.6 Conclusion	164
6 Conclusion	167

CHAPTER **1**

Introduction

Contents

1.1	Context and Motivation	19
1.2	The Periodic Event Scheduling Problem	21
1.3	Dissertation Overview and Contributions	23
1.3.1	Novel Geometric Insights	23
1.3.2	A Local Improvement Heuristic	24
1.3.3	Integrating Infrastructural Constraints	25
1.3.4	Optimising Infrastructure Use	25
1.3.5	Errata, Typos, and Formatting	26

1.1 Context and Motivation

The desire to be elsewhere has been a constant companion of the human experience since before we have records. A sudden surge of will, a daily habit of life, a seasonal rhythm, or a generational longing, the impulse to move has ascribed meaning to our existence and shaped the trajectory of human progress. Advances in available transportation systems, be they technological or institutional, have marked the most disruptive, transformative, and defining moments in human history, by connecting people and their cultures, flourishing trade and opportunity, and expanding the reach of our realities to the confine of our dreams.

As significant as location is as a source of inequality [6], transportation services are key to sustained and equitable growth. High quality transport networks lay as the backbone of regional development, boosting productivity and efficient resource allocation, enabling access to essential services, and fostering social mobility [7]. It is then with conscious responsibility that we must acknowledge how transport optimisation and its architects do not merely deal with sterile models and aseptic mathematics. Rather, by minimising travel times, disruptions, and costs, while increasing usage, coverage, and comfort, they enact a crucial and continuous step for the enhancement of modern society and its strive to a better place.

Internationally, transportation is recognised as a central pillar of sustainable development [8], featuring in frameworks such as the United Nations' Sustainable Development Goals, while also being highlighted for falling short both environmentally and socially in the latest progress report [9]. Also at the European level, transportation stands prominent, with the Trans-European Transport Network (TEN-T) being a founding goal of the Union [10], as well as a project that continues to this day, decades later. Today the transportation sector in the EU employs more than 5% of the working population, moves more people, for longer distances, at lower prices, and more efficiently than ever before [11], and efforts to develop these achievements further are underway [12]. In particular, special attention has been given to the transport of individuals, at all levels of scope and scale, notably with

billions of euros of investment reaching transportation projects across the continent via the NextGenerationEU recovery package.

As our networks grow in usage, size, density, and capacity, it is paramount to also develop alongside them our ability to plan and operate them. The practice and wisdom of the sector has undergone many transformations and upgrades throughout time, and by now it is commonplace to distinguish four macro stages [13], each in turn comprised of several subproblems: strategic planning, tactical planning, operational planning, and real-time control. Broadly speaking, strategic planning answers long-term questions, such as network design, while tactical planning decides upon which exact services to offer, within the potential of the provided infrastructure. Then, the operational stage is tasked with finding the best ways of achieving the desired service levels, and finally real-time management of the network deals with all daily unforeseeable fluctuations, adhering as much as possible to the target state of operations.

Planners see these stages as a complex and polyvalent task, but mathematical theory has initially struggled to undertake them head-on. Instead, it has been much simpler to begin by modelling many sequentially interconnected yet independent problems, each requiring specialised operations research methods to obtain high quality solutions. Mathematicians can then first focus on problems individually, establish theoretical knowledge to understand them, and develop practically relevant tools to solve them. Even without complete integration, these tools prove to be very useful to inform the decision-making process of planners and operators [14, 15]. However, to close the gap towards planners' desires, extensive research is ongoing so that these problems are merged and integrated mathematically as much as possible [16, 17].

The focus of this work will be periodic timetabling, a notorious problem in the tactical planning of transportation networks where service repeats at regular time intervals. Opting for periodic service is an attractive design choice for operators and passengers alike, because of its simplicity and predictability. In fact, it is in use in hundreds of networks around the world, from local to national level.

The Periodic Event Scheduling Problem (PESP) [5] has long been the primary choice to model and optimise periodic timetables in transportation networks [18], but as elegant as PESP is mathematically, it has proven very challenging computationally. Furthermore, solutions should be selected to also cater to the many passengers' travel times while staying robust to disruptions. Finally, the solving methods need to be able to timely re-compute solutions if partial or global re-planning is required. This strongly motivates us to study the mathematical structure of PESP, in the hopes of furthering practical aspects in terms of solving speed, solution quality, and integration with other planning stages.

We will first formally define PESP, our main interest, in Section 1.2, and then in Section 1.3 we present our contributions and the contents of the articles included in this dissertation.

1.2 The Periodic Event Scheduling Problem

As anticipated, the core interest in this work is PESP, and for that we hereby define its mathematical formulation precisely.

Initially formulated by Serafini and Ukovich in their seminal paper [5], the Periodic Event Scheduling Problem broadly proceeded as follows.

Definition 1.1 ([5]). — *A periodic event ε is a countably infinite set of events $e(p)$ indexed by $p \in \mathbb{Z}$, with occurrence times $t(e(p)) \in \mathbb{R}$ such that $t(e(p+1)) - t(e(p)) = T$, for a period $T \in \mathbb{N}$. A periodic event ε is scheduled once associated to an element $\tau(\varepsilon) \in \mathbb{R}/T\mathbb{Z}$, by having $\tau(\varepsilon)$ equal to the canonical projection of $t(e(p))$ into $\mathbb{R}/T\mathbb{Z}$, for every $p \in \mathbb{Z}$. A proper span is a non-surjective projection of a non-empty interval onto $\mathbb{R}/T\mathbb{Z}$, and a span constraint α is given by an ordered pair of periodic events $(\varepsilon_-, \varepsilon_+)$ and a proper span $\Delta(\alpha)$, by requiring that once the events are scheduled, it holds that $\tau(\varepsilon_+) - \tau(\varepsilon_-) \in \Delta(\alpha)$.*

Given then a finite set N of periodic events with common period T , and a finite set A of span constraints, the Periodic Event Scheduling Problem consists in scheduling all events in N while satisfying all span constraints in A .

A priori, the problem presents itself as a general tool to tackle a diverse swath of periodic problems, such as periodic job shops [19], traffic light scheduling [20, 21], periodic manufacturing operations [22], and airline scheduling [23]. Already then, PESP was proven to be NP-complete, then NP-complete with fixed $T \geq 3$ [24], and recently it was proven that even parametrisation by treewidth is of little use, together with other hardness results [25].

We will focus on the transport application, particularly that of railway networks. In said context, it is by now common practice to reform the original formulation slightly, by the use of an *event-activity network* [26], which is a directed graph $G = (V, A)$, where the nodes $v \in V$ are called *events* and the arcs $a \in A$ are called *activities*. We then have the standard PESP definition we will be working with from here on out.

Definition 1.2. — *Given a tuple (G, T, ℓ, u, w) , consisting of an event-activity network G , a period time $T \in \mathbb{N}$, lower and upper bounds $\ell, u \in \mathbb{R}^A$ on the activities, and a vector of weights $w \in \mathbb{R}^A$ on the activities, the Periodic Event Scheduling Problem consists in finding a periodic timetable $\pi \in \mathbb{R}^V$ and a periodic tension $x \in \mathbb{R}^A$ such that*

1. $\pi_j - \pi_i \equiv x_a \pmod{T}$ for all $a = (i, j) \in A$,
2. $\ell \leq x \leq u$
3. $w^\top x$ is minimum,

or to decide that no such π and x exist.

The major difference between the two is of course the weight vector, since Definition 1.1 is formulated as a constraint satisfaction problem, whereas Definition 1.2 adds an optimisation aspect. The latter is more effective also because the modulo constraint is capable of absorbing both the infinite nature of periodic events and the need for quotients and their canonical projections. Finally, the explicit graph layout serves us well in the transportation context, where we often deal with problems with geographic connotation. For example, in public transport, the events in the

graph will be departure and arrival events at stations, while activities will instead model driving activities (departure to arrival), or dwelling and transfer activities (arrival to departure). Consequently, the periodic timetable would be the collection of timestamps at which departures and arrivals happen, while the periodic tension would gather the duration of all activities. Finally, the weights typically condense estimates of passenger flows, so as to prioritise faster operations where they can benefit more people. They focus on the tensions alone, driven by the implicit wisdom of only optimising the perceived travel time of passengers, who will themselves choose at which periodic repetition of the timetable to start their journeys.

Further details about PESP and its use in transportation, whether theoretical or practical, will be developed in the coming chapters.

1.3 Dissertation Overview and Contributions

This work consists of four peer-reviewed and published articles [1–4]. The present author was the main author of each of them. Individually, they stand as the four core chapters in the dissertation, organised in the order of the list above, which corresponds to the order in which they were originally written. Concluding each chapter, there are short statements detailing individual contributions by the authors, according to the CRediT system [27]. The first two contributions study the pure mathematical structure of PESP as a standalone problem. The latter two, instead, study methods to integrate periodic timetabling and adjacent problems of the tactical stage, particularly the use of infrastructure, focusing on safety and efficiency.

Due to the nature of this work, all articles were transcribed verbatim, without the possibility of adjusting content or notation. Only minor typesetting corrections were applied (cf. Section 1.3.5).

1.3.1 Novel Geometric Insights

In Chapter 2 we follow *The Tropical and Zonotopal Geometry of Periodic Timetables* [1]. This article tackles a glaring gap in the literature, particularly in the theoretical study of the geometric structure of PESP.

Considering the problem as formulated in Definition 1.2, it can then be written as a mixed-integer program, by linearising the modulo constraint using auxiliary integer variables, called periodic offsets. Together with the periodic timetable π and the periodic tension x , the geometry of PESP is thus naturally split into three parts. Yet, only one of them received attention in the literature, namely the geometric properties of the periodic tensions [24, 28–30], whereas timetable and offset geometries remained overlooked. To mend this lacuna we study both. First, looking at the set of feasible periodic timetables, we characterise its structure as a disjoint union of polytropes, which are polytopes exhibiting both classical and tropical convexity. Careful examination of this decomposition and its relative arrangement over a torus, helps us envision a novel heuristic approach for PESP, which we analyse in [2], i.e., Chapter 3. Secondly, after reformulating the standard mixed-integer program into an alternative cycle-based model, we discover that the space of fractional cycle offsets forms a zonotope, affinely equivalent to the cographic zonotope of the instance graph. We examine its zonotopal tilings, exploring their connections to the hyperrectangle of fractional periodic tensions and the polytropes of the timetable space, thereby establishing new geometric insights into their interplay, and proving new tight bounds for the width of cycle bases.

1.3.2 A Local Improvement Heuristic

In Chapter 3 we follow *Tropical Neighbourhood Search: A New Heuristic for Periodic Timetabling* [2]. This article makes use of the theoretical foundations laid out in the preceding chapter, to craft and test a new local improvement heuristic for PESP, improving the state of the art.

Given the computational challenge PESP poses, various heuristic strategies are indispensable in aiding a pure MIP-based approach towards optimality. In this context, we propose the tropical neighbourhood search (**tns**) heuristic, and include it within the concurrent PESP solver of [31]. We then proceed to tune and test the performance of our contribution on the standard test instances of the PESPLib

benchmarking library [32], quickly achieving satisfactory computational results, as well as new incumbents for part of the collection.

This contribution was awarded the “Best Paper Award of ATMOS 2022” [33].

1.3.3 Integrating Infrastructural Constraints

In Chapter 4 we follow *Periodic Timetabling with Cyclic Order Constraints* [3]. This article addresses a key operational challenge, that of devising timetables in awareness of infrastructure usage and its limitations, here optimised in an integrated approach.

To ensure safe and robust operability, the planned positions and velocities of vehicles in a public transportation network are a carefully regulated matter, and these requirements find their way into any real-world application of PESP. While many such constraints are easily converted into pure PESP terms, other, such as the track occupation of stationary activities (e.g., dwelling at platforms) require more involved approaches [34]. Addressing this gap in the literature, so as to have a capable and unified tool, we define *Infrastructure-Aware PESP*, study its mathematical properties in relation to the extensive PESP literature, and explore ad hoc solution strategies for its mixed-integer formulation, centred on the cyclic orders of certain infrastructure-specific activities. To showcase the viability of this proposal, a case study is conducted on a real-world instance from the S-Bahn Berlin network. We find that Infrastructure-Aware PESP is a workable model in practice, and the incorporation of cyclic order information can yield considerable computational benefits.

1.3.4 Optimising Infrastructure Use

In Chapter 5 we follow *Periodic Event Scheduling with Flexible Infrastructure Assignment* [4]. Finally, this article furthers the integrated approach of the previous chapter, now not only optimising the timetable, but also offering strategies to most efficiently make use of existing infrastructural resources, thereby achieving a denser service offer at no extra costs.

Inspired by the integration of periodic timetabling and vehicle circulation scheduling found in [35], we study the problem of finding an optimal infrastructure assignment in an Infrastructure-Aware PESP context. Unlike in the previous chapter, the assignment of activities to infrastructure is no longer fixed a priori. Instead, each activity is provided with a range of options, allowing to potentially assign an activity to different infrastructure elements over different periodic repetitions. We therefore define Infrastructure-Aware PESP *with Assignment*, study its mathematical properties, as well as the impact of certain assumption on the assignment structure, and devise matching-based modelling and solution methods. We evaluate our approaches on three real-world instances, demonstrating the viability and the benefits of flexibly modelling infrastructure assignments. Our solutions result in higher timetable quality and better resource utilisation, a gap so far overlooked, while finally providing operators with complete and concise mathematical models for this task.

1.3.5 Errata, Typos, and Formatting

Within the bounds of the *Promotionsordnung des Fachbereichs Mathematik und Informatik der Freien Universität Berlin*, certain changes were applied, departing from the published versions of the articles:

1. Equation (2.40) was incorrect, and thereby mended.
2. A typo in the first paragraph of Section 2.4.4 has been mended.
3. A passage in the proof of Theorem 2.42 has been reworded, to avoid formatting issues.
4. The first line of Table 3.2 has been reworded, to avoid formatting issues.
5. The size of the text in Table 3.6 and Table 3.7 has been reduced, to avoid formatting issues.
6. A typo in two redundant lines of Table 4.2 has been mended.

7. Equation (5.17) was added, to avoid formatting issues. Subsequent equation numbering was adjusted accordingly.
8. The typesetting style of all mixed-integer programs has been harmonised.
9. All bibliographic information has been merged into a single references section, to avoid frequent and needless repetition.

CHAPTER **2**

The Tropical and Zonotopal Geometry of Periodic Timetables

Bortoletto, E., Lindner, N., Masing, B.
Discrete & Computational Geometry 2024.
DOI: 10.1007/s00454-024-00686-2.

This article is licensed under a Creative Commons Attribution 4.0 International License.
To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Abstract The Periodic Event Scheduling Problem (PESP) is the standard mathematical tool for optimizing periodic timetables in public transport. A solution to a PESP instance consists of three parts: a periodic timetable, a periodic tension, and integer offset values. While the space of periodic tensions has received much attention in the past, we explore geometric properties of the other two components. The general aim of this paper is to establish novel connections between periodic timetabling and discrete geometry. Firstly, we study the space of feasible periodic timetables as a disjoint union of polytropes. These are polytopes that are convex both classically and in the sense of tropical geometry. We then study this decomposition and use it to outline a new heuristic for PESP, based on neighbourhood relations of the polytropes. Secondly, we recognize that the space of fractional cycle offsets is in fact a zonotope, and then study its zonotopal tilings. These are related to the hyperrectangle of fractional periodic tensions, as well as the polytropes of the periodic timetable space, and we detail their interplay. To conclude, we also use this new understanding to give tight lower bounds on the minimum width of an integral cycle basis.

Contents

2.1	Introduction	31
2.2	The Periodic Event Scheduling Problem	33
2.2.1	Problem Definition	33
2.2.2	The Space of Feasible Periodic Timetables	35
2.2.3	The Space of Cycle Offsets	36
2.3	The Tropical Tiling of the Periodic Timetable Space	40
2.3.1	Weighted Digraphs and Polytopes	41
2.3.2	Slicing a Cube and Tiling the Torus	45
2.3.3	Polytopes in the Periodic Event Scheduling Context	51
2.3.4	An application of the Tropical Neighbourhood	53
2.4	Cycle Offset Zonotopes	60
2.4.1	Cographic Zonotopes	61
2.4.2	Scaled Cographic Zonotopes and the Cycle Offset Zonotope	63
2.4.3	A Duality Between Torus Polytopes and Zonotope Tiles	66
2.4.4	Constructing Zonotopal Tilings	72
2.4.5	Approximating the minimum width of a cycle basis	78
2.5	Outlook	82

2.1 Introduction

The timetable is the heart of a public transit system. Many public transportation networks across the world are operated in a periodic manner. The creation and optimization of periodic timetables is therefore an ubiquitous and frequent planning task. The standard mathematical model for periodic timetabling in public transport is the Periodic Event Scheduling Problem (PESP) developed in [5]. PESP is a challenging problem in various respects: On the theoretical side, the problem of finding a feasible periodic timetable is NP-hard for a given period time $T \geq 3$ [24, 28] or when the underlying constraint graph is series-parallel [25]. In practice, none of the current 22 benchmarking instances of the library PESP1ib [32] could up to today be solved to proven optimality. However, plenty of heuristic algorithms are available, connecting PESP with a zoo of well-known problems and techniques in combinatorial optimization, e.g., simplex algorithms [36, 37], maximum cuts [38], matchings [39], and Boolean satisfiability [40, 41]. The prevailing approach to solve PESP instances exactly is mixed-integer programming [31, 42]. For example, the subway network of Berlin has been optimized by solving such a mixed-integer programming model [14].

There are two central notions in periodic timetabling in public transport: A *periodic timetable* associates, intuitively speaking, a periodically repeating departure or arrival time to every stop of a trip. A *periodic tension* collects the durations of all activities in a public transport network, such as, e.g., driving between two neighboring stops, or transferring at a stop [18]. A timetable can easily be computed from a tension and vice versa. In the context of mixed-integer programming, the convex hull of feasible periodic tensions has been the central geometric object of study, and several classes of cutting planes have been deduced by studying this polyhedron [24, 28–30]. However, neither the space of timetables, nor of the *cycle offsets* – technical yet essential variables modeling the periodicity properties – has received much attention. The overarching goal of this paper is to provide novel geometric insights on the spaces that periodic timetables, tensions and cycle offsets live in, respectively. We investigate their relations and how the different spaces can

be linked to each other, opening up potential for new combinatorial and geometric tools to address PESP. Our main objects of study are the space of feasible periodic timetables Π , and the fractional offset zonotope Z . It turns out that the space Π is intimately related with tropical geometry. It is naturally embedded into a torus and decomposes into pairwise disjoint polytropes, i.e., polytopes that are also a tropical convex hull of finitely many points [43, 44]. We establish a new link between timetable and tension space, by recognising the tension polytope as the convex hull of the respective polytropes mapped to $\mathbb{R}^{A(G)}$. Furthermore, we analyse the neighbourhood relations of these polytropes, which leads to a new primal heuristic for PESP.

Regarding Z , the space of cycle offsets for the linear programming relaxation of the cycle-based mixed-integer programming formulation, one can observe that it has a specific structure: Z is a zonotope. Moreover, similar to a graphical zonotope, the maximal tiles of any fine zonotopal tiling of Z correspond to spanning trees. For PESP, we are particularly interested in integral points in Z , as these correspond to feasible cycle offsets. In our case, we can show that any tile of a fine zonotopal tiling of Z contains at most one integral point. Moreover, we establish a certain duality of the cycle offset zonotope Z to the space of periodic timetables Π : In any fine zonotopal tiling of Z , the tiles containing lattice points correspond with vertices of the polytropes in the decomposition of Π . Conversely, the tropical vertices of those polytropes can be used to construct a fine zonotopal tiling of Z .

Finally, we show how to use the zonotope Z to approximate the minimum width of integral cycle bases, a notion introduced to estimate the efficiency of the cycle-based mixed-integer programming formulation [45]. As a byproduct, we obtain that the number of spanning trees of a graph is at most the product of the lengths of the cycles in any integral cycle basis.

This paper is structured in three main parts: Firstly, we formally introduce PESP and related basic properties in Section 2.2. Secondly, we show how tropical geometry relates to PESP in Section 2.3. Lastly, in Section 2.4, we show how to utilize tools of zonotope theory on the cycle offset space and how it relates back to

concepts we have established before. We conclude the paper with a short outlook in Section 2.5, and provide a list of symbols with short summaries on page 83.

The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

2.2 The Periodic Event Scheduling Problem

In this section, we formally introduce the Periodic Event Scheduling Problem and cover some of its standard properties. We will work with two equivalent mixed-integer linear programming models, namely an incidence- and cycle-based formulation of PESP. The former will be introduced in Section 2.2.2, which motivates the definition of the timetable space Π – our first geometric object of interest. The latter formulation, as well as the cycle offset zonotope Z , is introduced in Section 2.2.3. This section also covers some further background knowledge on integral cycle bases. In that context, we state a general algebraic theorem on the relation of cycle matrices of integral cycle bases to incidence matrices. This theorem is an algebraic reformulation of a well-known PESP-specific property, namely the *cycle periodicity property*.

2.2.1 Problem Definition

We start by presenting the well-known standard formulation of the Periodic Event Scheduling Problem (PESP), first introduced by Serafini and Ukovich in [5].

Definition 2.1. — *An instance of the Periodic Event Scheduling Problem (PESP) consists of:*

- a directed graph G , with node set $V(G)$ and arc set $A(G)$,
- a period time $T \in \mathbb{N}$,
- lower and upper bounds $\ell, u \in \mathbb{R}^{A(G)}$.

For any arc (i, j) the length $u_{ij} - \ell_{ij}$ of the interval $[\ell_{ij}, u_{ij}]$ is called the span of that arc. Let \equiv_T denote equivalence modulo T . A periodic timetable is a vector $\pi \in \mathbb{R}^{V(G)}$, and any vector $x \in \mathbb{R}^{A(G)}$ such that $x_{ij} \equiv_T \pi_j - \pi_i$ for every $(i, j) \in A(G)$

is called a periodic tension associated to π . A periodic tension x is said to be feasible for the given PESP instance if $\ell \leq x \leq u$, and a periodic timetable π is said to be feasible if there exists a feasible periodic tension x associated to π . Given an instance (G, T, ℓ, u) as above, the feasibility version of the Periodic Event Scheduling Problem (PESP) consists in finding a feasible periodic timetable π and a feasible associated tension x . Similarly, for non-negative arc weights $w \in \mathbb{R}_{\geq 0}$, PESP can be considered as an optimization problem. In this case, the goal is to find a feasible periodic timetable and tension such that the weighted tension $w^\top x$ is minimised, or to decide that the instance is infeasible. Instances of the optimization problem will be denoted by (G, T, ℓ, u, w) .

In the context of periodic timetabling in public transport, the vertices of G can, e.g., represent arrival or departure events of trains at the stations of a public transit network. The arcs of G describe relations between those events, such as driving or dwelling, whose duration is in practice subject to certain time bounds. A periodic timetable π is hence an assignment of arrival and departure times compatible with those bounds, whereas a periodic tension x is simply the associated durations of the activities between events. In these terms, the objective of the optimization problem is to minimize the weighted travel time, where the weights can, e.g., model the passenger volume. Given a timetable π it is immediate to directly compute an associated tension x , but also the converse is trivial: Given a tension x one can arbitrarily assign $\pi_v = 0$ for some node $v \in V(G)$, and then proceed to traverse the rest of the node set in any connected fashion, such as breadth-first search rooted at v , and assign the rest of the timetable π based on the connecting arcs used during the traversal. We refer to Liebchen and Möhring [18] on how to express practical requirements of periodic timetables within the PESP model.

Meaningful real-world timetabling instances are comparably large. In order to be able to visualize the geometric ideas of this paper, we will therefore make use of artificial smaller examples.

Example 2.1. — *To illustrate Definition 2.1, consider the small instance as given by the graph in Figure 2.1, which will serve as a running example.*

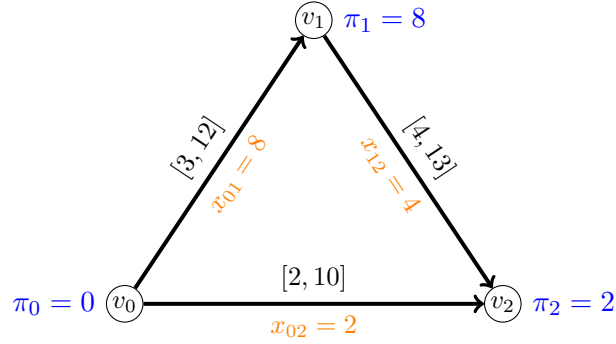


Figure 2.1: Exemplary graph G with labels $[l_{ij}, u_{ij}]$ for the arc-bounds for period time $T = 10$. A feasible timetable and corresponding periodic tension are marked in color.

In practice ℓ and u are often integral. In this case, by a result of Odjik [24], feasibility of the PESP instance implies the existence of an integral optimal periodic timetable and an associated integral periodic tension.

In the sequel, we will make use of some standard preprocessing assumptions [42]: We suppose that G is simple, 2-connected, and has no pair of antiparallel arcs. Moreover, we assume $\mathbf{0} \leq \ell < \mathbf{T}$ and $\mathbf{0} \leq u - \ell < \mathbf{T}$, where $\mathbf{1} := (1, \dots, 1)$, $\mathbf{0} := 0 \cdot \mathbf{1}$ and $\mathbf{T} := T \cdot \mathbf{1}$ indicate the all-ones, all-zeros and all- T vectors, respectively. The assumption $u - \ell < \mathbf{T}$ implies that any feasible timetable has a unique associated tension.

2.2.2 The Space of Feasible Periodic Timetables

For an instance (G, T, ℓ, u, w) as in Definition 2.1, PESP is commonly formulated as a mixed-integer linear program (MIP) [5]. If B denotes the incidence matrix of G , then the formulation is as follows:

$$\min \quad w^\top x \quad (2.1a)$$

$$\text{s.t.} \quad -B^\top \pi + Tp = x, \quad (2.1b)$$

$$\ell \leq x \leq u, \quad (2.1c)$$

$$\pi \in \mathbb{R}^{V(G)}, x \in \mathbb{R}^{A(G)}, p \in \mathbb{Z}^{A(G)}, \quad (2.1d)$$

where $p \in \mathbb{Z}^{A(G)}$ is called *periodic offset*, a vector of auxiliary integer variables used to linearise the modulo constraints. In view of the MIP (2.1), we will study

the following two spaces:

Definition 2.2. — *Given a PESP instance (G, T, ℓ, u) , we define the periodic tension polytope as*

$$X := \text{conv} \left\{ x \in \mathbb{R}^{A(G)} \mid \exists \pi \in \mathbb{R}^{V(G)} \exists p \in \mathbb{Z}^{A(G)} : x = -B^\top \pi + Tp, \ell \leq x \leq u \right\}. \quad (2.2)$$

Definition 2.3. — *Given a PESP instance (G, T, ℓ, u) , we define Π as the space of feasible periodic timetables, i.e.,*

$$\Pi := \left\{ \pi \in \mathbb{R}^{V(G)} \mid \exists p \in \mathbb{Z}^{A(G)}, \forall (i, j) \in A(G) : \ell_{ij} \leq \pi_j - \pi_i + Tp_{ij} \leq u_{ij} \right\}. \quad (2.3)$$

The periodic tension polytope X can be understood as the convex hull of the coordinate projection $(\pi, x, p) \mapsto x$ of all feasible solutions. The set Π arises as well as a coordinate projection $(\pi, x, p) \mapsto \pi$ of all feasible solutions, but without taking the convex hull. As per the introduction, X has been object of extensive study in periodic timetabling [24, 28–30], whereas Π has not received much attention in the past. Notice that no convex hull is taken in Π . We will see that the timetable space Π presents a periodic behaviour, so that its convex hull yields the whole ambient space $\mathbb{R}^{V(G)}$, provided that the PESP instance is feasible. We will explore Π with methods of tropical geometry in Section 2.3.

2.2.3 The Space of Cycle Offsets

Another well-known mixed-integer linear program to formulate PESP, which we will make use of in Section 2.4, uses a cycle-based approach [28, 45]. We will first introduce various standard concepts of the graph cycle literature, which can be further reviewed in [46].

An *oriented cycle* in G is a vector $\gamma \in \{-1, 0, 1\}^{A(G)}$ such that $B\gamma = \mathbf{0}$, where B is the incidence matrix of G . The oriented cycles in G generate a free abelian subgroup of $\mathbb{Z}^{A(G)}$, the *cycle space* of G . The positive and negative parts of an oriented cycle γ are denoted as γ^+ and γ^- . A *directed cycle* in G is a non-negative oriented cycle. The rank of the cycle space is the *cyclomatic number* μ , which can

be computed as $\mu = |A(G)| - |V(G)| + c(G)$, where $c(G)$ indicates the number of connected components of G . As we assumed that G is 2-connected, we will always assume $c(G) = 1$. Notice that, intuitively, one may be seduced to think of oriented cycles as closed simple paths, whereas, in truth, an oriented cycle may self-intersect multiple times, and may even be disconnected, as two connected oriented cycles γ_1 and γ_2 that are vertex-disjoint immediately imply the existence of the oriented cycle $\gamma_1 + \gamma_2$ in the cycle space.

An *integral cycle basis* \mathcal{B} of G is a μ -tuple $(\gamma_1, \dots, \gamma_\mu)$ of oriented cycles in G that generate the cycle space of G as a \mathbb{Z} -module. That is, any cycle γ in the cycle space of G can be expressed as a linear combination of elements of \mathcal{B} with integer coefficients. The matrix $\Gamma = (\gamma_1, \dots, \gamma_\mu)^\top \in \mathbb{Z}^{\mathcal{B} \times A}$ is called the *cycle matrix* of \mathcal{B} . A particular class of integral cycle bases is formed by *fundamental cycle bases*, which consist of the fundamental cycles of a spanning tree of G . That is, given a spanning tree S , the fundamental cycle basis induced by S consists of each cycle formed by an arc $a \in A(G) \setminus A(S)$ and the unique path in S connecting the two endpoints of a , taken with appropriate orientation. One of the reasons fundamental cycle bases are of interest is the fact that they are a subset of integral cycle bases that is very simple to construct, since a spanning tree of G is all that is needed to proceed. We refer to Example 2.2, where the cycle space and corresponding matrix of the running example are discussed.

We now state a well-known fact of the periodic timetabling literature, the *cycle periodicity property*.

Theorem 2.1 ([26, 45]). — *Let \mathcal{B} be an integral cycle basis of G with cycle matrix Γ . A vector $x \in \mathbb{R}^{A(G)}$ is a periodic tension if and only if $\Gamma x \equiv_T \mathbf{0}$.*

We give a new alternative proof of the cycle periodicity property with purely algebraic means.

Theorem 2.2. — *Let \mathcal{B} be an integral cycle basis of G with cycle matrix Γ , and let B denote the incidence matrix of G . Then $\ker \Gamma = \text{im } B^\top$ as \mathbb{Z} -modules.*

Proof. We first use a common inclusion and rank argument, and all that will be left to prove is that a certain torsion group is in fact trivial.

The inclusion $\ker \Gamma \supseteq \operatorname{im} B^\top$ is clear, as the rows of Γ belong to the kernel of B by the definition of oriented cycles. Since Γ has full row rank μ , the rank of $\ker \Gamma$ is then $|A(G)| - \mu = |V(G)| - 1$, which coincides with the rank of the image of B^\top . It follows that $\ker \Gamma / \operatorname{im} B^\top$ is a torsion group, and it remains to show that it is trivial.

Let $x \in \ker \Gamma$. Since $\ker \Gamma / \operatorname{im} B^\top$ is a torsion group, there is a natural number a such that $a \cdot x = B^\top \pi$ for some integral vector $\pi \in \mathbb{Z}^{V(G)}$. We can assume that π is primitive, i.e., that the greatest common divisor of all its entries is 1. Since the columns of B^\top sum to zero, we can actually remove an arbitrary column of B^\top , so that $a \cdot x = B'^\top \pi'$ for some submatrix B' of B and some subvector π' of π . The matrix B'^\top has now full column rank, so that we can choose an invertible submatrix B'' of B such that $a \cdot x' = B''^\top \pi'$ for a suitable subvector x' of x , and π' is the unique solution to this system of linear equations. By Cramer's rule, each entry of π' is the quotient of a multiple of a by the determinant of B'' . Since B and hence B'' is totally unimodular, each entry of π' is hence divisible by a . Since this holds for any dropped column of B^\top , we conclude that each entry of π is divisible by a . As π was primitive, this means that $a = 1$ and hence $x = B^\top \pi$, so that $x \in \operatorname{im} B^\top$. \square

Remark 2.1. — *Theorem 2.2 states that the sequence*

$$\mathbb{Z}^{V(G)} \xrightarrow{B^\top} \mathbb{Z}^{A(G)} \xrightarrow{\Gamma} \mathbb{Z}^{\mathcal{B}} \rightarrow 0 \quad (2.4)$$

is exact. The result can hence be interpreted in terms of graph cohomology. Although the cycle space over \mathbb{Z} has been thoroughly investigated (e.g., [47]), there seems to be no direct proof in the literature.

The cycle periodicity property Theorem 2.1 serves as the key motivation for an alternative MIP formulation as introduced in [26]. To that end, let \mathcal{B} be an

integral cycle basis of the digraph G and let Γ be the corresponding cycle matrix. Then PESP can equivalently be formulated as

$$\min \quad w^\top x \tag{2.5a}$$

$$\text{s.t.} \quad \Gamma x = Tz, \tag{2.5b}$$

$$\ell \leq x \leq u, \tag{2.5c}$$

$$z \in \mathbb{Z}^B, x \in \mathbb{R}^{A(G)}, \tag{2.5d}$$

where $z \in \mathbb{Z}^B$ is called *cycle offset*, a vector of auxiliary integer variables akin to the periodic offset. In fact, given some feasible solution (π, x, p) for the MIP (2.1), we can quickly recover the correct cycle offset z for which the given x is feasible by simply computing

$$z = \frac{\Gamma x}{T} = \frac{\Gamma(-B^\top \pi + Tp)}{T} \stackrel{\text{Theorem 2.2}}{=} \Gamma p. \tag{2.6}$$

In this sense, we can also view (2.5) as a variant of (2.1), where the periodic timetable variables π have been eliminated and the linear independence between the periodic offset variables p has been removed. Although removed as variables, a coherently feasible timetable can easily be reconstructed from a given tension x by fixing $\pi_v = 0$ for some vertex v , and then traversing the input graph and assigning potentials according to the tension of the traversed arcs.

Example 2.2. — *As the exemplary graph in Figure 2.1 consists of a single oriented cycle (up to sign), its cycle space has rank 1 and the corresponding cycle matrix can be expressed as*

$$\Gamma = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix}. \tag{2.7}$$

For the given periodic tension $x = (8, 2, 4)$ this results in $\Gamma x = 8 - 2 + 4 = 10$, i.e., x corresponds to a cycle offset of $z = 1$ for period time $T = 10$.

The space of feasible periodic tensions is the same for the two MIP formulations (2.1) and (2.5). In particular, for the cycle-based formulation (2.5), X can be rewritten as

$$X = \text{conv} \left\{ x \in \mathbb{R}^{A(G)} \mid \exists z \in \mathbb{Z}^B : \Gamma x = Tz, \ell \leq x \leq u \right\}, \tag{2.8}$$

which agrees with Definition 2.2. Since the cycle-based formulation (2.5) eliminates linear dependencies among the periodic offset variables p of the incidence-based formulation (2.1), we will investigate the projection $(x, z) \mapsto z$ onto the cycle offset variables rather than $(\pi, x, p) \mapsto p$.

Definition 2.4. — *Given a PESP instance (G, T, ℓ, u) and an integral cycle basis \mathcal{B} of G with cycle matrix Γ , we define the cycle offset zonotope as*

$$Z := \left\{ z \in \mathbb{R}^{\mathcal{B}} \mid \exists x \in \mathbb{R}^{A(G)} : \Gamma x = Tz, \ell \leq x \leq u \right\}. \quad (2.9)$$

The cycle offset zonotope Z will be the object of study in Section 2.4, where we will see that it is, in fact, a zonotope. Let us stress that Z is the the projection of the natural LP-relaxation, meaning that only its integer points correspond to cycle offsets for a PESP instance.

Having introduced the main geometric objects, namely the timetable space Π and the cycle offset zonotope Z , as well as their link, the tension polytope X , we will proceed to put them in the appropriate geometric context: In Section 2.3 we focus on Π and its relation to X , and interpret it in terms of tropical geometry. Section 2.4 focuses on the zonotope Z and establishes a certain duality tying back to the results of Section 2.3.

2.3 The Tropical Tiling of the Periodic Timetable Space

In this section, we will discuss the space Π of periodic timetables through the lens of tropical geometry. Our inquiry will focus first on establishing more general results, and only later we will fully contextualize them in the optic of periodic event scheduling. In Section 2.3.1, we will develop a few constructions regarding weighted digraphs, tropical convexity and polytropes. These objects will be the centre of our study, and we present some of their crucial properties, with statements about their dimension and characterizing their vertices. Furthermore, in Section 2.3.2, we describe how the aforementioned objects can be sliced out of a hypercube, and

subsequently used to tile a particular torus. In conclusion, in Section 2.3.3, we will show how the entire work so far done applies to periodic event scheduling, relating also to known results of the literature, and finally allow us to develop a new solving heuristic, discussed in Section 2.3.4, successfully implemented and analyzed in [2].

2.3.1 Weighted Digraphs and Polytopes

We begin by following [44], establishing known definitions and properties about weighted digraph polyhedra, otherwise also called shortest path polyhedra in [48].

Definition 2.5. — *Given a digraph D with weights $\kappa: A(D) \rightarrow \mathbb{R}$, the weighted digraph polyhedron of (D, κ) , denoted by $W(D, \kappa)$, is defined by the points $\pi \in \mathbb{R}^{V(D)}$ that satisfy the inequalities*

$$\pi_j - \pi_i \leq \kappa_{ij}, \quad \forall (i, j) \in A(D). \quad (2.10)$$

The points of a weighted digraph polyhedron correspond to *feasible potentials* of the weighted digraph, again see [48]. In particular, $W(D, \kappa)$ is empty if and only if κ is not conservative, i.e., there exists a negative directed cycle in (D, κ) .

A general observation is that any feasible potential of (D, κ) , i.e., any point of $W(D, \kappa)$, can be translated by any real multiple of the all-ones vector $\mathbf{1}$ and remains feasible. Indeed, if $\pi_j - \pi_i \leq \kappa_{ij}$ then $(\pi_j + k) - (\pi_i + k) \leq \kappa_{ij}$ for all $k \in \mathbb{R}$. This implies that any non-empty weighted digraph polyhedron contains $\mathbb{R}\mathbf{1} = \langle \mathbf{1} \rangle$ in its lineality space. In fact, as seen in [44, §2.3-§2.4], we have that if D is weakly connected and $W(D, \kappa)$ is non-empty, then the lineality space of $W(D, \kappa)$ is exactly $\mathbb{R}\mathbf{1}$. Moreover, if D is strongly connected, then the whole recession cone of $W(D, \kappa)$ is just $\mathbb{R}\mathbf{1}$.

Closely connected to weighted digraph polyhedra are *polytropes*. They provide a different point of view, as they arise in the context of tropical geometry, i.e., the algebraic geometry over the *tropical semiring* $\mathbb{T} = (\mathbb{R} \cup \{\infty\}, \oplus, \odot)$. For any two $a, b \in \mathbb{T}$ the tropical operations are defined as

$$a \oplus b := \min\{a, b\} \quad \text{and} \quad a \odot b := a + b. \quad (2.11)$$

The tropical semiring has ∞ and 0 as additive and multiplicative identity elements, respectively. Lastly, as no element in \mathbb{T} except ∞ has an additive inverse, we can understand tropical numbers to have no sign in the sense we may be used to. What mainly interests us is the theory of so-called tropical convexity. We point to [49] for a thorough introduction to tropical convexity, and [50] for a general compendium of tropical geometry and combinatorics. The definition of tropical convexity, first appeared in [49], is similar to the traditional setting.

Definition 2.6. — *A set $S \subset \mathbb{T}^n$ is tropically convex if*

$$(a \odot x) \oplus (b \odot y) \in S \tag{2.12}$$

for every $x, y \in S$, and any $a, b \in \mathbb{T}$. The tropical convex hull of a set $R \subset \mathbb{T}^n$ is the smallest tropically convex set containing R .

No restrictions are posed on the tropical scalar coefficients a and b , which implies that any tropically convex set is closed under tropical scalar multiplication. Alternatively this means that non-empty tropically convex sets are unbounded in \mathbb{T}^n as they must always contain $\mathbb{T} \odot \mathbf{0} = \mathbb{R}\mathbf{1} \cup \{(\infty, \dots, \infty)\}$. We therefore consider the *tropical projective space* \mathbb{TP}^{n-1} , which is defined as the quotient of $(\mathbb{T}^n \setminus \{(\infty, \dots, \infty)\})/\mathbb{T}$ via tropical scalar multiplication. The tropical projective space \mathbb{TP}^{n-1} contains the affine chart $\mathbb{R}^n/\mathbb{R}\mathbf{1} \cong \mathbb{R}^{n-1}$, which is called *tropical projective $(n-1)$ -torus* in [50], or *tropical affine space* in [43]. Note that instead in [49] we have that $\mathbb{R}^n/\mathbb{R}\mathbf{1} \cong \mathbb{R}^{n-1}$ is referred to as tropical projective space.

There also exists an alternative tradition dealing with tropical convexity, which can be traced to [51]. This mirrors Definition 2.6, but also includes the familiar bound $a \oplus b = 0$ on the scalars. By doing that, tropically convex sets are not necessarily unbounded anymore, as they do not always contain $\mathbb{T} \odot \mathbf{0}$, implying that the quotient into the projective space is not necessary anymore. In this light, it is clear that the definition of [49] is rather analogous to that of a convex *cone*, and then de-homogenizing said cone yields the convex set product of the definition of [51]. However, we will go forward adopting the definition without bounds on

the scalars and keep working in tropical projective space. One reason is that in our specific context of periodic event scheduling, periodic timetables indeed can be in practice shifted by any real multiple of $\mathbf{1}$ without changing the actual periodic tension of the PESP solution.

It is interesting to point out that tropical convexity and traditional convexity do not imply each other, which draws attention to the cases where, instead, both properties are upheld.

Definition 2.7 ([43]). — *A polytrope is the tropical convex hull of a finite set of points in $\mathbb{R}^n/\mathbb{R}\mathbf{1}$ that is also traditionally convex. The points of a smallest such generating set are called the tropical vertices of the polytrope.*

The set of tropical vertices is known to be unique and to have fixed size n . This interestingly means that all polytropes are tropical simplices in tropical projective space.

There is a correspondence between polytropes and weighted digraph polyhedra:

Theorem 2.3 ([44], Proposition 48). — *If (D, κ) is a strongly connected weighted digraph on n vertices, then the quotient modulo $\mathbb{R}\mathbf{1}$ of its weighted digraph polyhedron $W(D, \kappa) \subseteq \mathbb{R}^n$ is a polytrope in $\mathbb{R}^n/\mathbb{R}\mathbf{1}$. Moreover, every polytrope in $\mathbb{R}^n/\mathbb{R}\mathbf{1}$ arises this way.*

Taking such a quotient in the context of weighted digraph polyhedra is intuitively very appropriate once we realize that $\mathbb{R}\mathbf{1} = \ker B^\top$ for the incidence matrix B of any weakly connected digraph.

The dimension of polyhedra of this class can be easily computed in a purely graph theoretical manner. The *equality graph* $E(D, \kappa)$ is defined as the undirected graph on the same vertex set as (D, κ) such that two vertices are adjacent if and only if they are contained in a directed cycle of weight 0 in (D, κ) . Then:

Theorem 2.4 ([44], Lemma 5). — *If $p \in \mathbb{Z}^{A(G)}$ is a feasible periodic offset vector, then*

$$\dim W(D, \kappa) = c(E(D, \kappa)) - 1, \quad (2.13)$$

where $c(E(D, \kappa))$ indicates the number of connected components of $E(D, \kappa)$.

In conclusion, we recall the following result about the vertices of these tropical objects, with a slight extension:

Theorem 2.5 ([43], Theorem 7). — *The tropical vertices of a polytrope are in bijection with shortest path trees rooted at each vertex of the digraph from which the polytrope arises as a weighted digraph polyhedron.*

Theorem 2.6. — *The vertices of a polytrope correspond to a unique spanning subgraph of the digraph from which the polytrope arises as a weighted digraph polyhedron.*

Proof. Let D be the digraph in question, with n vertices, and let κ be a weight function on its arcs. Each vertex of the arising polytrope corresponds to a 1-dimensional minimal face F of the weighted digraph polyhedron $W(D, \kappa)$. Said weighted digraph polyhedron is defined by the system

$$\pi_j - \pi_i \leq \kappa_{ij}, \quad \forall (i, j) \in A(D). \quad (2.14)$$

Hence F is defined by at least $n - 1$ active inequalities in (2.14), precisely $n - 1$ of which being linearly independent. Then the subgraph of D induced by all arcs corresponding to these active inequalities is spanning, since the rows are linearly independent, and no two minimal faces F can give rise to the same subgraph, or they would have all the same active inequalities and hence be the same face. \square

While Theorem 2.5 associates only tropical vertices with shortest spanning trees, Theorem 2.6 now allows us to characterize also non-tropical vertices, namely the so-called *pseudo-vertices* of [43]. They thus correspond to spanning trees that are not shortest path trees. Notice that, in fact, the theorem speaks of spanning subgraphs and not just spanning trees. This is attributed to possible degeneracy – even in full dimensional polytropes certain vertices may be contained in more than just $n - 1$ facets.

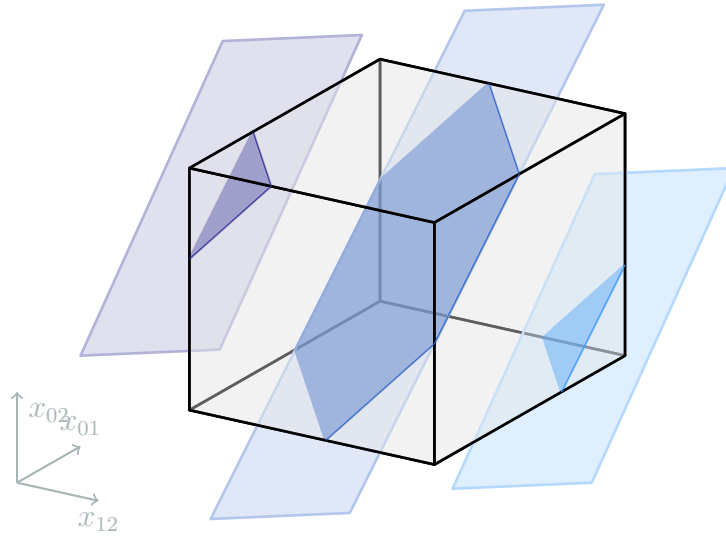


Figure 2.2: A possible C sliced by three distinct S_p , and the corresponding $r(p)$ highlighted.

2.3.2 Slicing a Cube and Tiling the Torus

Let us now consider a graph G and a hyperrectangle in $\mathbb{R}^{A(G)}$ with edges aligned with the coordinate axes. We will intersect this hyperrectangle with a specific class of affine subspaces induced by the graph G and parametrized by certain integer offsets. This will lead to an interesting polyhedral subdivision by polytropes of the space $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$, and will then in turn be of use for understanding the geometry of periodic timetables.

Take $C := [\ell, u]$, a hyperrectangle with $\ell, u \in \mathbb{R}^{A(G)}$. With that, consider the incidence matrix $B \in \mathbb{R}^{V(G) \times A(G)}$ of the graph G , and let $S_p := \text{im } B^\top + Tp$ for $p \in \mathbb{Z}^{A(G)}$ and $T \in \mathbb{N}$. We want to focus on the intersection of these two objects, the hyperrectangle C and the affine subspace S_p , for different values of p . We denote them as

$$r(p) := C \cap S_p = \left\{ x \in \mathbb{R}^{A(G)} \mid \exists \pi \in \mathbb{R}^{V(G)} : -B^\top \pi + Tp = x, \ell \leq x \leq u \right\}, \quad (2.15)$$

where p is integral. An example visualization of these intersections is found in Figure 2.2. As we have observed before, $\ker B^\top = \mathbb{R}\mathbf{1}$, so that in the above definition

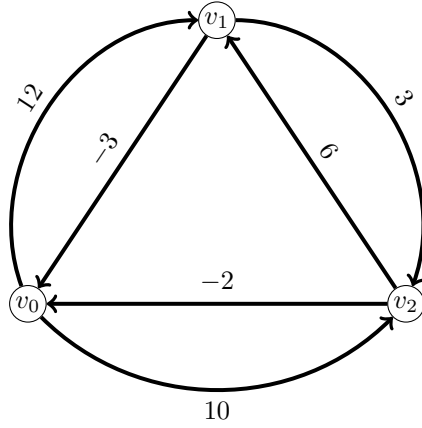


Figure 2.3: Digraph \overline{G} , for G the digraph in Figure 2.1. The weights on the arcs (a_{01}, a_{02}, a_{12}) are set as per the proof of Lemma 2.7, with $\ell = (3, 2, 4)$, $u = (12, 10, 13)$, and $p = (0, 0, 1)$.

of $r(p)$ it is sufficient to only draw π from $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$. Thus, we consider the affine map

$$m_p: \mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1} \longrightarrow \mathbb{R}^{A(G)} \quad (2.16)$$

$$\pi \longmapsto -B^\top \pi + Tp,$$

and define

$$R(p) := m_p^{-1}(r(p)) = \left\{ \pi \in \mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1} \mid \ell \leq -B^\top \pi + Tp \leq u \right\} \quad (2.17)$$

as the preimage of $r(p)$ w.r.t. m_p .

We will make use of the following auxiliary construction.

Definition 2.8. — Given a digraph G , we define its two-way graph \overline{G} as the digraph with vertex set $V(\overline{G}) := V(G)$ and arc set $A(\overline{G}) := A(G) \cup A(G^\top)$, where $A(G^\top) = \{(j, i) \mid (i, j) \in A(G)\}$.

An example of the construction is seen in Figure 2.3.

Lemma 2.7. — Each $R(p)$ is a polytope corresponding to a weighted digraph polyhedron for the graph \overline{G} . If $u < \ell + \mathbf{T}$, then all such polytopes are pairwise disjoint. If $u = \ell + \mathbf{T}$, then the set of all $R(p)$, with $p \in \mathbb{Z}^{A(G)}$, gives a polyhedral subdivision of the space $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$, induced by a hyperplane arrangement.

Proof. Notice that any $R(p)$ is a polyhedron defined by the inequalities

$$\ell_{ij} - Tp_{ij} \leq \pi_j - \pi_i \leq u_{ij} - Tp_{ij}, \quad \forall (i, j) \in A(G), \quad (2.18)$$

which can be seen as two standard weighted digraph polyhedron inequalities for an arc (j, i) and (i, j) respectively (cf. Definition 2.5). Hence $R(p)$ is a polytrope corresponding to a weighted digraph polyhedron for the graph \overline{G} in the sense of Theorem 2.3.

If $u < \ell + \mathbf{T}$, then take $p, p' \in \mathbb{Z}^{A(G)}$, with $p \neq p'$, so there is an arc $(i, j) \in A(G)$ such that $p_{ij} \neq p'_{ij}$. Assume w.l.o.g. that $p_{ij} > p'_{ij}$. Then for any two points $\pi \in R(p)$ and $\pi' \in R(p')$ we have that

$$\pi_j - \pi_i \leq u_{ij} - Tp_{ij} < \ell_{ij} - T(p_{ij} - 1) \leq \ell_{ij} - Tp'_{ij} \leq \pi'_j - \pi'_i, \quad (2.19)$$

hence the points must be distinct, and the polytropes disjoint.

Instead, if $u = \ell + \mathbf{T}$, then for any $\pi \in \mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$ there exists a p such that $R(p) \ni \pi$, simply by choosing $p_{ij} = \lceil \frac{\ell_{ij} - \pi_j + \pi_i}{T} \rceil$. Moreover, for any $p, p' \in \mathbb{Z}^{A(G)}$ with $p \neq p'$, any defining inequality of $R(p')$ that is not one for $R(p)$ is supported by a hyperplane that does not intersect the interior of $R(p)$. Therefore, the hyperplane arrangement given by all defining inequalities of all $R(p)$, for $p \in \mathbb{Z}^{A(G)}$, divides the space $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$ into cells, each one a single polytrope of our collection. \square

Going forward, we will refer to this decomposition of $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$ as the *polytropical decomposition*. The decomposition shows a periodic pattern, which is emphasized by the following:

Lemma 2.8. — *Let $p \in \mathbb{Z}^{A(G)}$ and $q \in \mathbb{Z}^{V(G)}$. Then $R(p + B^\top q) = R(p) + Tq$, and $r(p + B^\top q) = r(p)$.*

Proof. For any $\pi \in R(p + B^\top q)$ holds

$$\ell \leq -B^\top \pi + T(p + B^\top q) = -B^\top (\pi - Tq) + Tp \leq u, \quad (2.20)$$

so that $\pi - Tq \in R(p)$, and vice versa, and so $R(p + B^\top q) = R(p) + Tq$. Moreover, we then also have that

$$r(p + B^\top q) = m_{p+B^\top q}(R(p + B^\top q)) = m_{p+B^\top q}(R(p) + Tq). \quad (2.21)$$

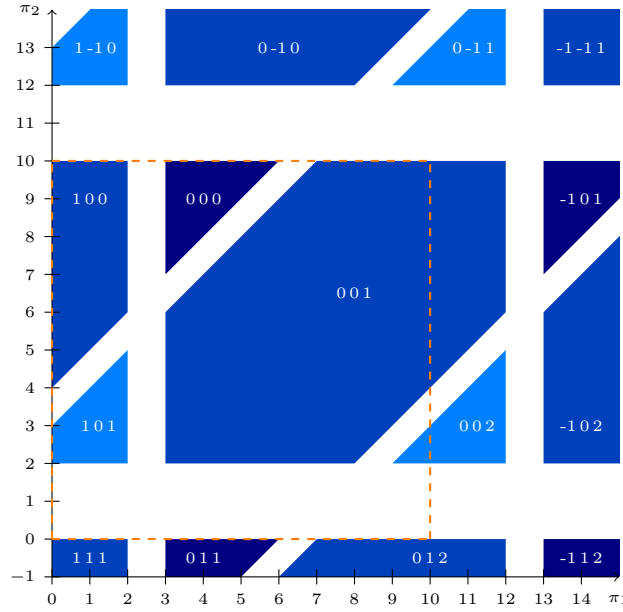


Figure 2.4: Polytropical decomposition with underlying digraph \overline{G} as in Figure 2.3, and $u < \ell + T$. The fundamental domain corresponding to the torus \mathcal{T} is dashed in orange. Each region is labeled by its periodic offset vector $p = (p_{01}, p_{02}, p_{12})$.

Let $\pi \in R(p)$. Then

$$m_{p+B^\top q}(\pi + Tq) = -B^\top(\pi + Tq) + T(p + B^\top q) = -B^\top\pi + Tp = m_p(\pi), \quad (2.22)$$

and so $r(p) = r(p + B^\top q)$. \square

Example 2.3. — A visualization of the polytropical decomposition described in Lemma 2.7 is found in Figure 2.4. Starting with the digraph \overline{G} as in Figure 2.3, with $\ell = (3, 2, 4)$ and $u = (12, 10, 13)$, the space $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1} \cong \mathbb{R}^2$ is filled by pairwise disjoint polytopes. The periodic pattern discussed in Lemma 2.8 becomes evident, visualised by polytopes in different shades of blue.

To get rid of the symmetry described in Lemma 2.8, we finally define

$$\mathbf{R}(p) := \left(R(p) / (T\mathbb{Z})^{V(G)} \right) / \mathbb{R}\mathbf{1}, \quad (2.23)$$

where clearly $\mathbf{R}(p + B^\top q) = \mathbf{R}(p)$ for any choice of $q \in \mathbb{Z}^{A(G)}$. With that, we have:

Lemma 2.9. — Let $p, p' \in \mathbb{Z}^{A(G)}$ such that both $\mathbf{R}(p)$ and $\mathbf{R}(p')$ are non-empty. Then $\mathbf{R}(p) = \mathbf{R}(p')$ if and only if $\Gamma p = \Gamma p'$, where Γ is the cycle matrix of an integral cycle basis of G .

Proof. As above, if $\mathbf{R}(p) = \mathbf{R}(p')$ then $p' = p + B^\top q$ for some $q \in \mathbb{Z}^{V(G)}$. Then, by Theorem 2.2, we have $\ker \Gamma = \text{im } B^\top$, and so $\Gamma p' = \Gamma(p + B^\top q) = \Gamma p$. Conversely, if $\Gamma p = \Gamma p'$, then $p - p' \in \ker \Gamma$, and by the same result there exists a $q \in \mathbb{Z}^{V(G)}$ such that $p - p' = B^\top q$, where q can be found to be integral because B^\top is totally unimodular. Then, we have $\mathbf{R}(p) = \mathbf{R}(p')$. \square

Let us make a few valuable observations as direct results from the lemmata above:

Observation 2.1. — *Our look at $r(p)$ as the intersections of the hyperrectangle C with the spaces S_p allowed us to introduce the map m_p and its preimage, defined in the space $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$. By Lemma 2.8 we know that many $R(p)$ effectively generate the same image in C , which enabled us to take a quotient in $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$ to reflect that. We can actually define*

$$\mathcal{T} := \left(\mathbb{R}^{V(G)} / (T\mathbb{Z})^{V(G)} \right) / \mathbb{R}\mathbf{1}, \quad (2.24)$$

the space where all polytropes $\mathbf{R}(p)$ live, which is a $(|V(G)| - 1)$ -dimensional torus. In the example of Figure 2.4 the fundamental domain corresponding to \mathcal{T} is dashed in orange.

By Lemma 2.9 we have a clear way of distinguishing among $\mathbf{R}(p)$, by noting which p and p' have distinct images under Γ . In fact, with the integral cycle basis \mathcal{B} and cycle matrix Γ , we can finally denote uniquely our objects in \mathcal{T} by indexing them via $z \in \mathbb{Z}^{\mathcal{B}}$, as:

$$\mathfrak{R}(z) := \mathbf{R}(p) \subseteq \mathcal{T}, \quad (2.25)$$

for some p such that $\Gamma p = z$. The then well-defined maps

$$\begin{aligned} \mathbf{m}_z: \mathfrak{R}(z) &\longrightarrow C \\ \pi &\longmapsto -B^\top \pi + Tp \end{aligned} \quad (2.26)$$

for $z \in \mathbb{Z}^{\mathcal{B}}$ filter all corresponding maps m_p , from $R(p)$ through \mathcal{T} and into C , for any p such that $\Gamma p = z$. Finally, as $\mathbb{R}^{V(G)}/\mathbb{R}\mathbf{1}$ has a polytropical decomposition by the various $R(p)$, detailed in Lemma 2.7, also the torus \mathcal{T} decomposes in various $\mathfrak{R}(z)$. A visualization of said decomposition can be found in Figure 2.5.

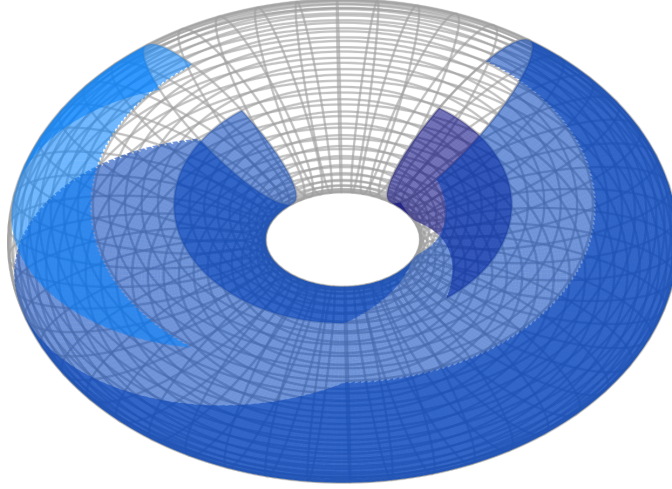


Figure 2.5: The orange fundamental domain shown in Figure 2.4 is wrapped onto the torus \mathcal{T} . Compared to Figure 2.4, this is an essential representation of the space of preimages.

Lemma 2.10. — *Let $z \in \mathbb{Z}^{\mathbb{B}}$. Then*

$$\text{im}(\mathfrak{m}_z) = \{x \in C \mid \Gamma x = Tz\}. \quad (2.27)$$

Proof. By Lemma 2.9, $\text{im}(\mathfrak{m}_z) = r(p) = C \cap S_p$ for any $p \in \mathbb{Z}^{A(G)}$ with $\Gamma p = z$. Now, with Theorem 2.2,

$$\begin{aligned} C \cap S_p &= C \cap (\ker \Gamma + Tp) \\ &= \{x \in C \mid \Gamma(x - Tp) = 0\} \\ &= \{x \in C \mid \Gamma x = T\Gamma p\} \\ &= \{x \in C \mid \Gamma x = Tz\}. \end{aligned} \quad (2.28) \quad \square$$

The following is a direct consequence of Theorem 2.6:

Corollary 2.11. — *Each vertex of each $\mathfrak{R}(z)$ in \mathcal{T} corresponds to a unique spanning subgraph of \overline{G} .*

Example 2.4. — *Considering the weighted digraph of Figure 2.3, but with variable periodic offset p , we focus on the shortest paths rooted at v_1 . As before $\ell = (3, 2, 4)$ and $u = (12, 10, 13)$. Figure 2.6 highlights them, and also specifies a representative of the timetable so obtained. The trees are coloured with the same colours of Figure 2.4, and each timetable is indeed a vertex in the polytrope of corresponding offset and colour.*

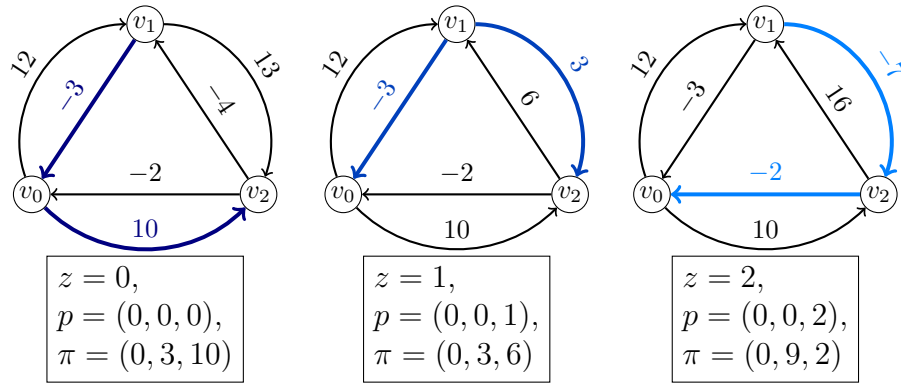


Figure 2.6: Spanning tree structures rooted at v_1 in \overline{G} for the three polytopes.

2.3.3 Polytopes in the Periodic Event Scheduling Context

So far, this chapter has been introduced in terms independent of the PESP context. Now, we briefly go over how the above constructions apply to our problem of interest.

Observation 2.2. — *Let us fix a PESP instance (G, T, ℓ, u) as defined in Definition 2.1. It is quick to obtain the basic tools of the constructions detailed in Section 2.3.2. In fact, the two vectors ℓ and u of lower and upper bounds define the hyperrectangle $C = [\ell, u]$, while the period T and the incidence matrix B of the instance graph G allow us to define $S_p = \text{im } B^\top + Tp$ for $p \in \mathbb{Z}^{A(G)}$ and the maps m_p .*

Compare now the definition of $r(p)$ and $R(p)$ against the mixed integer program formulation of PESP defined in (2.1), as well as the definitions of X and Π (cf. Definition 2.2 and Definition 2.3). The set of feasible tensions of the linear relaxation then reads as

$$X_{LP} := \left\{ x \in \mathbb{R}^{A(G)} \mid \exists \pi \in \mathbb{R}^{V(G)}, p \in \mathbb{R}^{A(G)} : -B^\top \pi + Tp = x, \ell \leq x \leq u \right\}, \quad (2.29)$$

which is just the hyperrectangle C , since for any $x \in C$, we have that $m_{x/T}(\mathbf{0}) = x$.

It is clear that the p of S_p can indeed be seen as a periodic offset vector, and that the maps m_p map timetables to tensions. The feasible timetables $\pi \in R(p) \subseteq \Pi$ are then mapped by m_p to their associated feasible tension $x \in r(p) \subseteq X$. Equivalently, for feasible cycle offset vectors z , the maps \mathbf{m}_z map the torus \mathcal{T} into X , and the

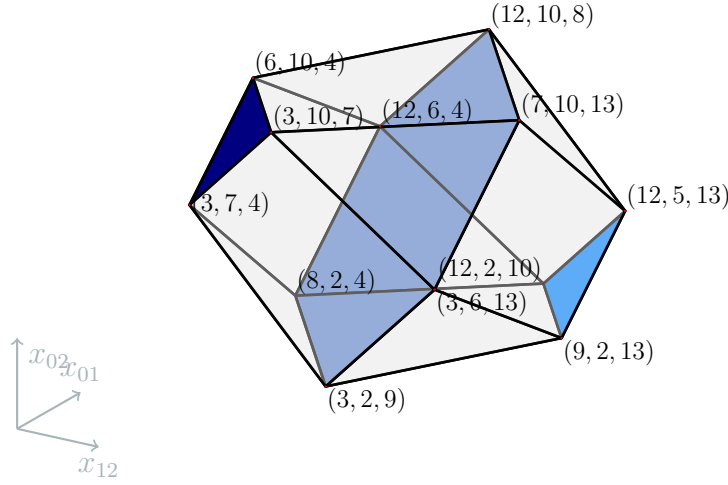


Figure 2.7: Periodic tension polytope X with the embeddings of $\mathfrak{R}(z)$ for $z \in \{0, 1, 2\}$, corresponding to the instance of Example 2.1 and its tiling, shown in Figure 2.4.

tension polytope X is then none other than the convex hull of an appropriate lifting of the polytropes scattered in the torus \mathcal{T} . Conversely, the latter are but a projection of the tensions onto \mathcal{T} . We have by Lemma 2.10:

Theorem 2.12. — *The periodic tension polytope X can be described as*

$$X = \text{conv} \{ \text{im } \mathbf{m}_z \mid z \text{ feasible cycle offset} \}. \quad (2.30)$$

A visualization is found in Figure 2.7. These considerations motivate our interest in the number of feasible cycle offset vectors or, equivalently, on the number of polytropes $\mathfrak{R}(z)$ in \mathcal{T} , which we will discuss in Section 2.4.

Before turning our insights into an algorithmic idea for PESP, we briefly discuss a dimension result. To this end, we recall Theorem 2.4, and express the condition that all polytropes have maximum possible dimension in terms of oriented cycles in G .

Corollary 2.13. — *Suppose that for all oriented cycles γ in G holds $\gamma_+^\top u - \gamma_-^\top \ell \not\equiv_T 0$. Then $\dim \mathbf{R}(p) = |V(G)| - 1$ for all feasible periodic offsets $p \in \mathbb{Z}^{A(G)}$.*

Proof. Let p be a feasible periodic offset and let γ be an oriented cycle in G . Since p is integral, the condition $\gamma_+^\top u - \gamma_-^\top \ell \not\equiv_T 0$ guarantees that $\gamma_+^\top (u - Tp) - \gamma_-^\top (\ell - Tp) \neq 0$. In particular, the weight of $\bar{\gamma}$ is non-zero in \bar{G} . Since this holds for all feasible p and all oriented cycles γ , we conclude that all vertices of the equality graph are isolated. By Theorem 2.4, we conclude $\dim \mathbf{R}(p) = |V(G)| - 1$. \square

Example 2.5. — *There are only two oriented cycles in our running Example 2.1: For the clockwise oriented cycle γ holds $\gamma_+^\top u - \gamma_-^\top \ell = 12 + 13 - 2 = 23$, and for its anti-clockwise counterpart $-\gamma$ holds $(-\gamma)_+^\top u - (-\gamma)_-^\top \ell = 10 - 4 - 3 = 3$. Both numbers are not integer multiples of the period time $T = 10$. By Corollary 2.13, we conclude that all polytropes must have dimension 2, which is confirmed by Figure 2.4.*

2.3.4 An application of the Tropical Neighbourhood

Up until now, our considerations have been rather theoretical in nature. The goal of this section is, instead, to outline a new practical approach to solving PESP. We utilize our novel point of view to construct a procedure and provide the necessary theoretical background and correctness results. The concluding algorithmic idea was successfully realized and analyzed in [2]. The main motivation stems from the following observation:

Observation 2.3. — *Consider*

$$x = -B^\top \pi + Tp, \quad \text{or similarly} \quad \Gamma x = Tz. \quad (2.31)$$

When restricting to a fixed offset vector, p or z , in either problem formulation, incidence-based (2.1) or cycle-based (2.5), we are met with a standard linear program. Therefore, in practice, solving a PESP instance with fixed offset can be done efficiently.

We will now provide a way to sequentially consider fixed offset values, taking advantage of the geometric results that we developed so far.

To proceed we first describe the relationship between polytropes $\mathfrak{R}(z)$ that are close to each other, and how the whole collection of polytropes is positioned inside \mathcal{T} . Having established that, we present a simple consequent definition of a way to encode the collection of polytropes as an undirected graph, and thereby define an appropriate graph-exploration strategy.

A natural first step is to see how a change in the offset vector shifts the focus from a certain polytrope in the torus to another. To this end, we initially consider

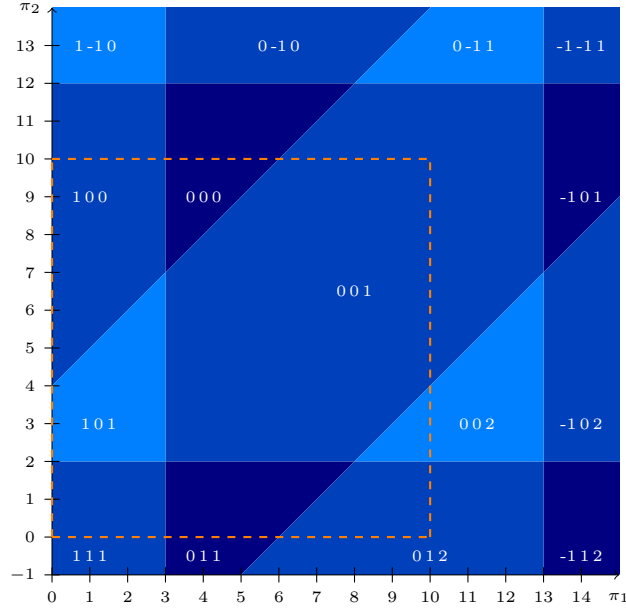


Figure 2.8: Polytropical decomposition in the case of $u = \ell + \mathbf{T}$ with underlying digraph \overline{G} as in Figure 2.3. The fundamental domain corresponding to the torus \mathcal{T} is dashed in orange.

a *limit* PESP instance where $u = \ell + \mathbf{T}$. Such construction is actually excluded a priori in our hypothesis, since we had imposed $u - \ell < \mathbf{T}$, which is a reasonable assumption in practice. Nonetheless, we consider this special limit case anyway, since it helps us in presenting and understanding certain properties of the polytrope arrangement, which we will then easily adapt to the more general and practical case.

By Lemma 2.7 we know that the polytropical decomposition of a limit case such as $(G, T, \ell, \ell + \mathbf{T})$ gives a polyhedral subdivision of the space, induced by a hyperplane arrangement. See Figure 2.8 for an example. In comparison to Figure 2.4, we see that certain “infeasible bands” that encircled the polytropes there, now have disappeared in the case of $(G, T, \ell, \ell + \mathbf{T})$. Said bands are of the form $u_{ij} - Tp_{ij} < \pi_j - \pi_i < \ell_{ij} - T(p_{ij} - 1)$ for each arc $(i, j) \in A(G)$ and $p_{ij} \in \mathbb{Z}$, and their Manhattan-width is the T -complement of the span of the arc (i, j) , namely $T - (u_{ij} - \ell_{ij})$. In the limit case these bands do not exist, since by construction we now set $u - \ell = \mathbf{T}$.

Proceeding now in analyzing changes in the offset vectors we have the two following helpful lemmata.

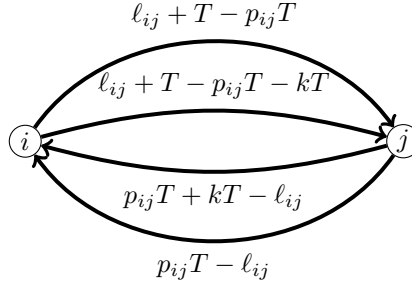


Figure 2.9: The subgraph of \overline{G} induced by the sole vertices i and j , prior to the resolution of the tropical sum of $(\overline{G}, \kappa(p)) \oplus \kappa(p + ke_{ij})$. Note that we applied the substitution $u = \ell + T$.

Lemma 2.14. — Consider the limit instance $(G, T, \ell, \ell + T)$. Let $p \in \mathbb{Z}^{A(G)}$ be an offset vector, k an integer such that $k \notin \{-1, 0, 1\}$, and e_{ij} the canonical basis vector of the arc (i, j) . Then we have that $R(p) \cap R(p + ke_{ij}) = \emptyset$.

Proof. Let $\kappa(p)$ be a weight function on $A(\overline{G})$ such that $\kappa(p)_{ij} := u_{ij} - Tp_{ij} = \ell + T - Tp_{ij}$ for all arcs $(i, j) \in A(G)$, and $\kappa(p)_{ji} := Tp_{ij} - \ell_{ij}$ for all arcs $(j, i) \in A(G)$. Considering $R(p)$ and $R(p + ke_{ij})$ as weighted digraph polyhedra, their weighted digraphs are $(\overline{G}, \kappa(p))$ and $(\overline{G}, \kappa(p + ke_{ij}))$ respectively. Now, making use of [44, Lemma 6], we have that the intersection $R(p) \cap R(p + ke_{ij})$ is a weighted digraph polyhedron, whose weighted digraph is $(\overline{G}, \kappa(p) \oplus \kappa(p + ke_{ij}))$. The only place in \overline{G} where the latter tropical sum is not obvious by idempotency is precisely between nodes i and j , where we have a situation akin to that of Figure 2.9. The first and last of the four arcs are the ones with weight $\kappa(p)$, whereas the second and third have weight $\kappa(p + ke_{ij})$. If $k > 1$ then the minimum κ_{ij} is attained in the second arc and the minimum κ_{ji} is attained in the fourth arc, and so these two arcs are kept. Now the weight of the remaining 2-cycle is $\ell_{ij} + T - p_{ij}T - kT + p_{ij}T - \ell_{ij} = T(1 - k) < 0$, which is negative. If instead $k < -1$ we similarly keep the first and third arc, and again we have a negative weight cycle. In either case, $R(p) \cap R(p + ke_{ij}) = \emptyset$. \square

Lemma 2.15. — Let $p \in \mathbb{Z}^{A(G)}$ be an offset vector, k an integer such that $k \in \{-1, 1\}$, and e_{ij} the canonical basis vector of the arc (i, j) . Then we have that $R(p) \cap R(p + ke_{ij}) = F$, where F is the face of either polytrope in which one of the inequalities given by the arc (i, j) is tight.

Proof. Everything repeats as in the previous proof, except that now the weight of the remaining 2-cycle is $\ell_{ij} + T - p_{ij}T - kT + p_{ij}T - \ell_{ij} = T(1 - k) = 0$ if $k = 1$ and $\ell_{ij} + T - p_{ij}T + p_{ij}T + kT - \ell_{ij} = T(1 + k) = 0$ if $k = -1$. In either case, we do not create an immediate infeasibility in $i - j - i$, and our construction is identical to that in [44, Lemma 7]. By that lemma we have restricted to the face in which one of the inequalities given by the arc (i, j) is tight, in particular the lower bound inequality if $k = 1$ and the upper bound inequality if $k = -1$. Note that F can be the whole of $R(p)$ as well as the empty face. \square

To conclude we can better discuss the dimension of F , and gain further insights. We operate with the same hypothesis of Lemma 2.15, and briefly assume $R(p)$ to be full-dimensional. If F is a facet of $R(p)$, then it is also a facet of $R(p + ke_{ij})$, which must then be full-dimensional as well. If that was not the case we would have $R(p + ke_{ij}) = F$, of dimension $n - 1$, and this could only happen if there was a 0-weight cycle in $(\overline{G}, \kappa(p))$ of length 2, which is incompatible with our assumption of G having no antiparallel arcs. When F instead is not a facet of $R(p)$, then $R(p + ke_{ij}) = F$. If that was not the case there would be a point in the interior of $R(p + ke_{ij})$ that is separated from $R(p)$ by the relevant (i, j) -inequality, which in turn would have to be facet-defining by Farkas' Lemma.

Resetting back to the case where $u - \ell < \mathbf{T}$, now all polytropes are again disjoint. By the preceding lemmata though, we now understand that only when an inequality arising from arc (i, j) is facet-tight in $\mathbf{R}(p)$ can we hope for $\mathbf{R}(p \pm e_{ij})$ to be non-empty, since otherwise we have proven that we would be restricting $\mathbf{R}(p \pm e_{ij})$ to being a face of $\mathbf{R}(p)$, and now the non-zero Manhattan-width of the corresponding (i, j) -band would render $\mathbf{R}(p \pm e_{ij})$ empty. Of course $\mathbf{R}(p \pm e_{ij})$ can still be empty anyway, when the complement of the span of arc (i, j) is too large, meaning the infeasible band is too wide.

Looking at the tiling of the limit case, it is now reasonable to consider two polytropes to be neighbours if they share a common facet. This leads us to extending this notion to the general case.

Definition 2.9. — Consider a PESP instance (G, T, ℓ, u) . Two non-empty polytropes $\mathbf{R}(p), \mathbf{R}(p') \subseteq \mathcal{T}$ are neighbours if and only if in the limit instance $(G, T, \ell, \ell + T)$ the polytropes corresponding to the same offsets p and p' intersect in a facet. The neighbourhood of $\mathbf{R}(p)$ is the set of its non-empty neighbours, or

$$N_p := \{\mathbf{R}(p') \neq \emptyset \mid \exists (i, j) \in A(G): p - p' = \pm e_{ij}\}. \quad (2.32)$$

In our example, we therefore consider $\mathbf{R}((0, 0, 1))$ to be a neighbour of both $\mathbf{R}((0, 0, 0))$ and $\mathbf{R}((0, 0, 2))$, while the two triangular polytropes are not neighbours, as is evidenced in Figure 2.8.

The considerations above allow us to describe neighbouring polytropes in a simpler way, making use of the periodic offsets alone:

Lemma 2.16. — Two non-empty polytropes $\mathbf{R}(p)$ and $\mathbf{R}(p')$ are neighbours whenever there exist representatives \tilde{p} and \tilde{p}' , meaning offset vectors such that $p - \tilde{p}, p' - \tilde{p}' \in \ker \Gamma$, such that $\tilde{p} - \tilde{p}' = \pm e_{ij}$, for some arc $(i, j) \in A(G)$.

In a similar vein, by the basic relationship between p and z as seen in (2.6), we have the following structural result.

Theorem 2.17. — Two non-empty polytropes $\mathfrak{R}(z)$ and $\mathfrak{R}(z')$ are neighbours whenever $z - z'$ is, up to sign, a column of Γ .

Proof. Consider p and p' preimages under Γ of, respectively, z and z' . If $\mathbf{R}(p)$ and $\mathbf{R}(p')$ are neighbours, by Lemma 2.16 we have that there are representatives \tilde{p} and \tilde{p}' such that $\tilde{p} - \tilde{p}' = \pm e_{ij}$, for some arc $(i, j) \in A(G)$. Then

$$z - z' = \Gamma p - \Gamma p' = \Gamma(\tilde{p} - \tilde{p}') = \pm \Gamma e_{ij}. \quad (2.33)$$

□

Similar to N_p we can now also define

$$N_z := \{\mathfrak{R}(z') \neq \emptyset \mid \exists (i, j) \in A(G): z - z' = \pm \Gamma e_{ij}\}. \quad (2.34)$$

We can finally turn to describing the entire tiling in terms of the neighbourhood relations.



Figure 2.10: The neighbourhood graph N of the tiling of the initial example from Figure 2.1, visualizing the neighbourhood relations. The colours correspond to the coloured tiles in Figure 2.4.

Definition 2.10. — *The neighbourhood graph, denoted by N , is an undirected graph, whose nodes are feasible cyclic offsets $z \in \mathbb{Z}^b$, and two nodes z and z' are adjacent if and only if the corresponding polytropes $\mathfrak{R}(z)$ and $\mathfrak{R}(z')$ are neighbours.*

Equivalently, one could define the nodes to be periodic offsets $p \in \mathbb{Z}^{A(G)}$, adjacent to their neighbours, and then take the quotient graph with respect to our usual equivalence relation

$$p \equiv p' \iff z := \Gamma p = \Gamma p' =: z'. \quad (2.35)$$

Example 2.6. — *We can visualize the neighbourhood graph of our running example, shown in Figure 2.10.*

As we anticipated, the neighbourhood graph N serves as the basis of a local improvement procedure, which we name *tropical neighbourhood search*, or **tns**. It explores the torus of Figure 2.5, polytrope by polytrope, using the structure of N as a guide. Basically, given any known feasible solution (x, z) to start with, the focus is put on the offset z , meaning a node in the neighbourhood graph N . Now the neighbourhood N_z can be systematically explored using the insight of Theorem 2.17, solving standard linear programs as we saw in Observation 2.3, one per each neighbour. If a better solution (x', z') is found, the search continues in $N_{z'}$. Of course the whole procedure can be equivalently restated using periodic offsets p instead. In that case only the linear program formulation to find the optima needs adjustment, and of course Lemma 2.16 is then the leading mechanism for exploration. For a detailed implementation and analysis of the procedure, we refer to [2].

Such a strategy is in spirit quite similar to the state of the art approach, the modulo network simplex (also **mns**), see [52]. The major problem of **mns** is that it

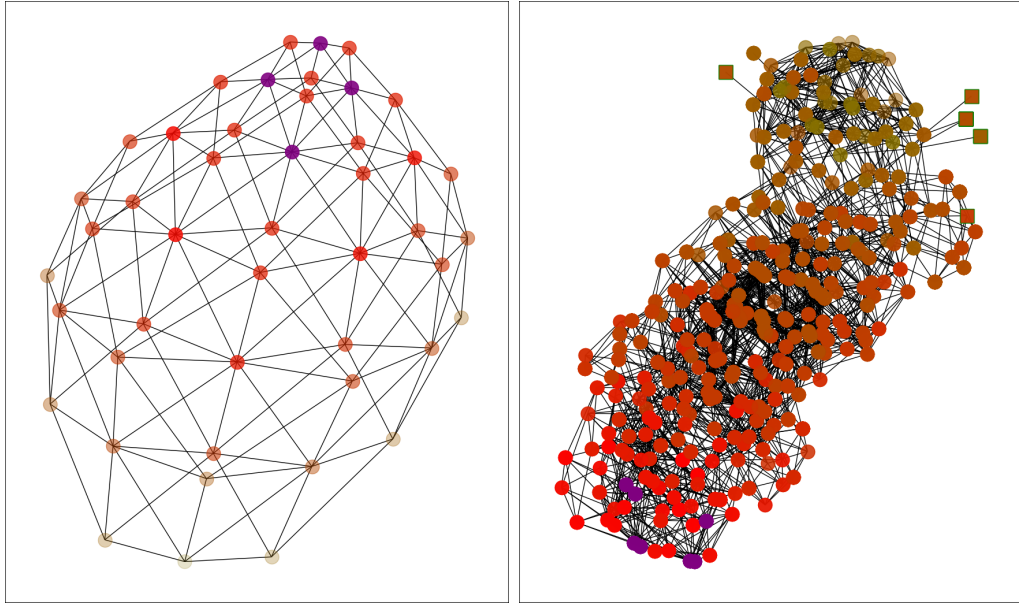


Figure 2.11: The neighbourhoods of the two heuristics compared: **tns** on the left, **mns** on the right.

sometimes reaches local optima, meaning solutions from which no improvement can be found via **mns**, even though the solution is not a global optimum. Examining example instances and comparing the search space of **mns** with that of **tns** we have found some interesting differences. First of all, there are adjacency relations in the **mns** search space that do not appear in the **tns** search space, and vice versa. This means that while **mns** can, in theory, explore solutions starting from (x, z) that are not in a neighbour of $\mathfrak{R}(z)$, also **tns** can find among the neighbours of $\mathfrak{R}(z)$ solutions that **mns** is not able to consider. We also found instances where the **mns** search space presents local optima whereas the **tns** search space has none, although there still exist instances with **tns** local optima.

Example 2.7. — For a certain instance with 8 nodes, 10 arcs, and $\mu = 3$, we computed the neighbourhood graph for the **tns** and **mns** heuristics, respectively shown on the left and on the right of Figure 2.11. For **tns** each node corresponds to a polytrope in the polytropical decomposition, and each arc adjuncts two polytropes whose offsets differ by 1 in a single coordinate. For **mns** each node corresponds to a spanning tree of the PESP instance, and each arc adjuncts two spanning trees that differ only by removing one arc and adding one other. The redder a node, the better

its objective value, with purple nodes being global optima. Squares, indeed here only present for mns , are local optima.

2.4 Cycle Offset Zonotopes

Having explored the space of periodic timetables, we will now turn our attention to the space Z of cycle offsets (Definition 2.4). It turns out that Z is a zonotope and affinely equivalent to the cographic zonotope associated to the directed graph G . We will therefore first recall some properties about zonotopes and zonotopal tilings, and discuss the cographic zonotope in Section 2.4.1. We will then introduce scaled cographic zonotopes and deduce an upper bound on the number of their integer points in terms of spanning trees in Section 2.4.2. Furthermore, we explore how such zonotopes Z and their tilings relate back to the results of Section 2.3: In Section 2.4.3 and Section 2.4.4 we show that a tiling gives rise to a duality relation between a tile in Z and specific points in polytropes in the torus \mathcal{T} and vice versa. We conclude this section by using our findings to approximate the width of a cycle basis, which serves as an estimation for the efficiency of branch-and-bound approaches for PESP. Independent of periodic timetabling, our analysis yields an upper bound on the number of spanning trees in terms of cycle lengths in a general graph, see Section 2.4.5.

We begin right away with the two central definitions for this section.

Definition 2.11 ([53]). — *A zonotope is the image of a hypercube under an affine map. In formulae, a zonotope in \mathbb{R}^μ can hence be written as*

$$Z(A, b) := \{Ac \mid \mathbf{0} \leq c \leq \mathbf{1}\} + b \quad (2.36)$$

for some generating matrix A with μ rows and some translation vector $b \in \mathbb{R}^\mu$. We will abbreviate $Z(A) := Z(A, \mathbf{0})$.

In the following we use the common notion of *polyhedral subdivisions*, see, e.g., [54]. Any face in the subdivision we will often call *cell*. It is a known property of zonotopes that they can be tiled by subzonotopes, see [54, §7.5].

Definition 2.12. — A zonotopal tiling of a zonotope $Z(A)$ is a polyhedral subdivision of $Z(A)$ such that the cells are translates of zonotopes $Z(A_S)$, where A_S is the submatrix arising from A by deleting the columns indexed by S . We call each maximal cell of such a subdivision a tile. The tiling is called fine if all tiles are generated by full-rank matrices, i.e., linearly independent vectors. In such a case, all tiles are parallelepipeds.

Every zonotope has a fine tiling [54, Corollary 7.5.10]. If A has μ rows and full row rank, then the tiles of a fine tiling are given as certain translates of $Z(A_S)$, where A_S ranges over all invertible $(\mu \times \mu)$ -submatrices of A .

2.4.1 Cographic Zonotopes

Let G be a directed graph with cyclomatic number μ . We will consider again an integral cycle basis of G with cycle matrix Γ as in Section 2.2.3. The graph G gives rise to a graphic oriented matroid, whose dual is the cographic oriented matroid $M^*(G)$ [55, 56].

Lemma 2.18. — The matrix Γ represents the cographic oriented matroid $M^*(G)$. In particular, a subset of columns of Γ indexed by $S \subseteq A(G)$ is linearly independent if and only if removing the arcs in S does not disconnect G .

Proof. By Theorem 2.2, the rows of Γ constitute a basis of the orthogonal complement of $\text{im}(B^\top)$ in $\mathbb{R}^{A(G)}$, where B denotes the incidence matrix of G . Since the graphic oriented matroid is represented by B , its dual $M^*(G)$ is hence represented by Γ [53, §6.3c]. \square

Definition 2.13. — The cographic zonotope of G is $Z(\Gamma)$.

This definition obviously depends on the choice of a cycle matrix. However, the combinatorics of the cographic zonotope are already defined by the cographic oriented matroid, and the choice of a different cycle matrix will lead to an affinely equivalent zonotope. In fact, such an affine transformation will also be unimodular, i.e., preserve integer points, as we consider only cycle matrices of integral cycle bases.

Observation 2.4. — $Z(\Gamma)$ is a lattice zonotope, i.e., all vertices are integer, and it is of dimension μ .

For zonotopal tilings, we obtain the following rephrasing of Lemma 2.18:

Lemma 2.19. — Let $Z(\Gamma)$ be the cographic zonotope of G . Fix any fine zonotopal tiling. Then there is a one-to-one correspondence between spanning subgraphs of G with $|A(G)| - k$ arcs and the k -dimensional cells of the tiling such that the arc set S of a spanning subgraph maps to a translate of $Z(\Gamma_S)$.

The cographic zonotope hence shares the property that tiles correspond to spanning trees with graphic zonotopes (see, e.g., [57, §2] or [54, §7.5]), but in a dual sense: The lower-dimensional cells of the zonotopal tiling are given by spanning subgraphs in our case, whereas the cells of a fine zonotopal tiling of a graphic zonotope are given by forests.

Having established this correspondence, we can obtain a formula for the volume of a cographic zonotope, dual to that of [57, Proposition 2.4]. We denote the set of spanning trees of a graph G by $\mathcal{S}(G)$. We will often identify a spanning tree $S \in \mathcal{S}(G)$ with its set of arcs.

Corollary 2.20. — The volume of the cographic zonotope $Z(\Gamma)$ of G is the number of spanning trees of G . In particular, the volume does not depend on the choice of the integral cycle basis.

Proof. By Lemma 2.19, we can compute the volume $\text{vol}(Z)$ as the sum of the volumes $\text{vol}(Z(\Gamma_S))$, for $S \in \mathcal{S}(G)$. As any $Z(\Gamma_S)$ is a parallelotope,

$$\text{vol}(Z(\Gamma_S)) = |\det(\Gamma_S)| = 1, \tag{2.37}$$

because any invertible $(\mu \times \mu)$ -submatrix of the cycle matrix Γ of an integral cycle basis has determinant ± 1 [58]. \square

We can also derive a formula for the number of integer points in the cographic zonotope:

Corollary 2.21. — *The number of integer points in the cographic zonotope $Z(\Gamma)$ of G equals the number of spanning subgraphs of G .*

Proof. By [59, Theorem 2.2] and Lemma 2.19, the number of integer points of $Z(\Gamma)$ is the sum of the greatest common divisors of the maximal minors of Γ_S , where S ranges over the spanning subgraphs of G . Since Γ is the cycle matrix of an integral cycle basis, each Γ_S is unimodular, so that these greatest common divisors need to be 1. □

2.4.2 Scaled Cographic Zonotopes and the Cycle Offset Zonotope

In order to connect the cographic zonotopes with PESP instances, we will now consider a scaled version of the cycle offset zonotope. Let G be a directed graph and let Γ be the cycle matrix of an integral cycle basis \mathcal{B} .

Definition 2.14. — *Let $d \in \mathbb{R}^{A(G)}$ with $\mathbf{0} < d < \mathbf{1}$. The scaled cographic zonotope of G is $Z(\Gamma_d)$, where Γ_d arises from Γ by scaling each column indexed by $a \in A(G)$ with d_a .*

For $d = \mathbf{1}$, we would obtain again the cographic zonotope. We are however now interested in the case $d < \mathbf{1}$: Recall that for a PESP instance (G, T, ℓ, u) , we defined the *cycle offset zonotope* in Definition 2.4 as

$$Z := \left\{ z \in \mathbb{R}^{\mathcal{B}} \mid \exists x \in \mathbb{R}^{A(G)} : \Gamma x = Tz, \ell \leq x \leq u \right\}. \quad (2.38)$$

Setting $d_a := \frac{u_a - \ell_a}{T}$ for all $a \in A(G)$, we obtain

$$Z = \left\{ \frac{\Gamma x}{T} \mid \ell \leq x \leq u \right\} = \{ \Gamma_d \cdot c \mid \mathbf{0} \leq c \leq \mathbf{1} \} + \frac{\Gamma \ell}{T} = Z \left(\Gamma_d, \frac{\Gamma \ell}{T} \right), \quad (2.39)$$

so that Z is indeed a translate of the scaled cographic zonotope $Z(\Gamma_d)$. In particular, both zonotopes are affinely equivalent.

Note that we always assume $u - \ell < T$, so that $d < \mathbf{1}$. We will also assume for this chapter that $u - \ell > \mathbf{0}$, so that indeed $d > \mathbf{0}$. This can always be achieved by contracting arcs with $\ell_a = u_a$ [42]. As a consequence, the rank of Γ_d is still μ , and the dimension of Z is the cyclomatic number $\mu = |\mathcal{B}|$.



Figure 2.12: The cycle offset zonotope of Example 2.1, with a fine tiling.

Example 2.8. — *The cycle offset zonotope of Example 2.1 is rather simple, as shown in Figure 2.12. Given that the graph has (up to sign) only one cycle, the dimension is indeed just 1. Using the lexicographical order on the arcs, we then obtain the scaled cycle matrix in this case as*

$$\Gamma_d = \frac{\begin{pmatrix} 1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 9 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 9 \end{pmatrix}^\top}{T} = (0.9 \quad -0.8 \quad 0.9). \quad (2.40)$$

The translation vector is

$$\frac{\Gamma \ell}{T} = \frac{\begin{pmatrix} 1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 3 & 2 & 4 \end{pmatrix}^\top}{10} = 0.5. \quad (2.41)$$

The resulting zonotope $Z = Z(\Gamma_d, \frac{\Gamma \ell}{T})$ then simply corresponds to the interval $[-0.3, 2.3]$. In this example, we have exactly three different spanning trees for G , namely $S_{01} = \{02, 12\}$, $S_{02} = \{01, 12\}$ and $S_{12} = \{01, 02\}$. The three tiles are then given by the translates of the three zonotopes

$$Z(\Gamma_{d,S_{01}}) = [0, 0.9], \quad Z(\Gamma_{d,S_{02}}) = [-0.8, 0] \quad \text{and} \quad Z(\Gamma_{d,S_{12}}) = [0, 0.9], \quad (2.42)$$

i.e., intervals of length as indicated by the entries of Γ_d , which also correspond to the co-tree arcs of the respective spanning trees. The related fine tiling is shown in Figure 2.12.

The scaling does not affect the combinatorics of fine zonotopal tilings as described in Lemma 2.19. For the volume, we obtain in analogy to Corollary 2.20:

Corollary 2.22. — *The volume of a scaled cographic zonotope $Z(\Gamma_d)$ is*

$$\text{vol}(Z(\Gamma_d)) = \sum_{S \in \mathcal{S}(G)} \prod_{a \in A(G) \setminus A(S)} d_a. \quad (2.43)$$

Again, the volume does not depend on the choice of the cycle basis. However, due to the scaling, the number of integer points drops dramatically in comparison to Corollary 2.21:

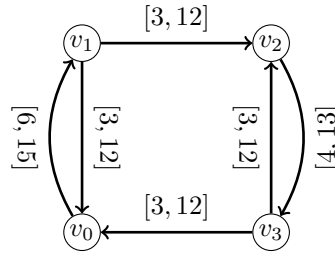


Figure 2.13: A PESP instance with period $T = 10$.

Theorem 2.23. — *Any translate of a scaled offset zonotope $Z(\Gamma_d)$ has at most as many integer points as there are spanning trees of G .*

Proof. Let A be an invertible $(\mu \times \mu)$ -submatrix of the cycle matrix Γ . We claim that $\mathbf{0}$ is the only integer point in the set $Q := \{Ay \mid -\mathbf{1} < y < \mathbf{1}\}$. Indeed, for any such integer point $q \in Q$, there is a unique solution y to $Ay = q$. By Cramer’s rule, y is integral, as the determinant of A is ± 1 . But then $y = \mathbf{0}$ and hence $q = \mathbf{0}$.

Now consider a fine tiling of Z and let $Z(A_d, b)$ be a tile. That is, A_d is an invertible $(\mu \times \mu)$ -submatrix of Γ , and b is some translation vector, possibly non-integral. Suppose that $z, z' \in Z(A_d, b)$ are both integral. Then there are c, c' with $\mathbf{0} \leq c, c' \leq \mathbf{1}$ such that $A_dc + b = z$ and $A_dc' + b = z'$. Thus $A_d(c - c')$ is integral. Let y denote the vector with entries $(c_a - c'_a)d_a$, for $a \in A(G)$. If A is the submatrix of Γ corresponding to A_d , then $Ay = A_d(c - c')$ and $-\mathbf{1} < y < \mathbf{1}$ because $d_a < 1$. We conclude that $y = \mathbf{0}$ and hence $c = c'$. This means that any tile in a fine tiling of Z contains at most one lattice point. In particular, using Lemma 2.19, the number of integer points is bounded by $|\mathcal{S}(G)|$. \square

Example 2.9. — *Consider the PESP instance graph in Figure 2.13. Contrary to our hypothesis it contains antiparallel arcs, which we initially wanted to avoid for the sake of simplicity of certain dimension arguments in Section 2.3.4. In general, these directed cycles of length two do not hinder neither PESP nor the heart of our considerations. In particular, we can construct the cycle offset zonotope of this instance and tile it. As integral cycle basis, we use the cycles induced by the three regions of the planar embedding as in Figure 2.13. The tiled zonotope is shown in Figure 2.14. There are 12 tiles in the picture, but only 11 integer points inside the*

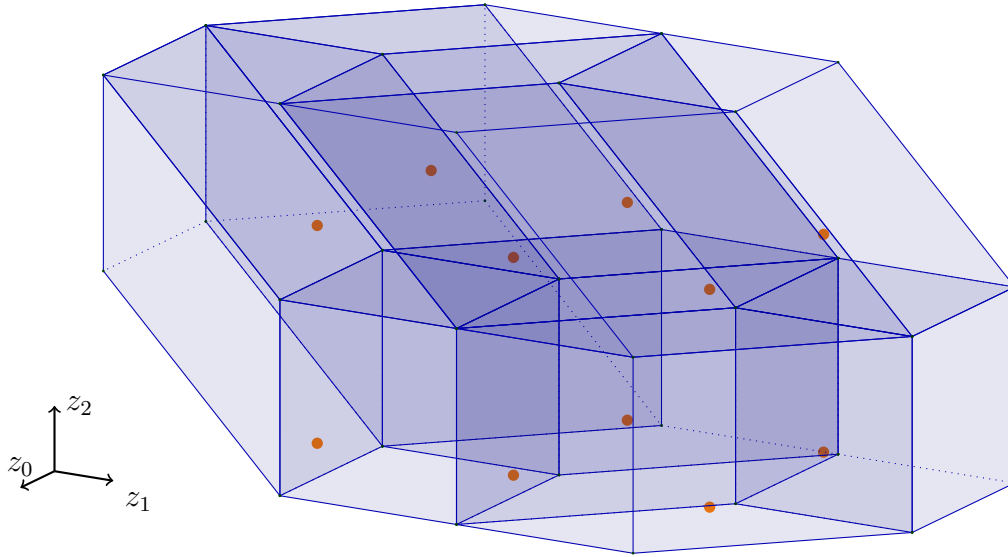


Figure 2.14: The tiled zonotope of the PESP instance of Figure 2.13, with integer points marked in orange.

zonotope, marked in orange. This is coherent with Lemma 2.19 and Theorem 2.23. In fact the graph of Figure 2.13 has exactly 12 spanning trees, and so 12 tiles are found. Moreover each tile has the potential of containing at most 1 integer point, and in fact we have 11 such points. In truth here there is one integer point that lays on a common facet of two tiles, thereby being contained in both. Regardless, it is in general still possible for a tile to contain no integer point altogether.

Example 2.10. — The number of integer points of $Z(\Gamma_d)$ does not depend on Γ , since Γ is unimodular [58], but on d . For example, let G be the directed cycle on n vertices and n arcs. Then G has n spanning trees, and $Z(\Gamma_d)$ is the interval $[0, \sum_{i=1}^n d_i]$. In particular, $Z(\Gamma_d)$ contains exactly n integer points if and only if $\sum_{i=1}^n d_i \geq n - 1$.

2.4.3 A Duality Between Torus Polytropes and Zonotope Tiles

Let G be a directed graph with integral cycle basis \mathcal{B} and cycle matrix Γ . Let further $\ell \in \mathbb{R}^{A(G)}$, $T \in \mathbb{N}$, and $d \in \mathbb{R}^{A(G)}$ with $\mathbf{0} < d < \mathbf{1}$. If we define $u := \ell + Td$, then we have $u < \ell + \mathbf{1}$, and we can consider slicing the hyperrectangle $C := [\ell, u]$ as in Section 2.3.2. There, in Observation 2.1, we constructed a decomposition of a



Figure 2.15: The tiled cycle offset zonotope of Example 2.1, with integer points marked in orange.

torus \mathcal{T} into polytropes $\mathfrak{R}(z)$ parameterized by integral vectors $z \in \mathbb{Z}^{\mathcal{B}}$. We will now relate this decomposition to the translated scaled cographic zonotope $Z\left(\Gamma_d, \frac{\Gamma\ell}{T}\right)$.

In terms of a PESP instance (G, T, ℓ, u) with $0 < u - \ell < T$ and $d := \frac{u-\ell}{T}$, we will hence connect the polytropical decomposition of the space of feasible periodic timetables modulo symmetries as in Section 2.3.3 with the cycle offset zonotope Z . Since there is no difference in concept (cf. Definition 2.14), we will use the perspective of cycle offset zonotopes from now on.

Lemma 2.24. — *The map $z \mapsto \mathfrak{R}(z)$ is a bijection between integer points of the cycle offset zonotope Z and non-empty polytropes in \mathcal{T} .*

Proof. Let z be an integer point of Z . Then, by definition of Z , there is a vector x with $\ell \leq x \leq \ell + Td = u$ such that $\Gamma x = Tz$. By Lemma 2.10, $x \in \text{im } \mathfrak{m}_z$, and so $\mathfrak{R}(z) \neq \emptyset$. Conversely, if $\mathfrak{R}(z)$ is a non-empty polytrope in \mathcal{T} , then we find $x \in \text{im } \mathfrak{m}_z$, and $\frac{\Gamma x}{T} = z$ is an integer point of Z . □

By Theorem 2.23, we immediately obtain:

Theorem 2.25. — *The number of polytropes in the decomposition of \mathcal{T} is at most $|\mathcal{S}(G)|$.*

Looking at Figure 2.15, we see that for the instance of Example 2.1 there are exactly three integral points in its cycle offset zonotope. In fact, there are also three tiles, three polytropes as seen in Figure 2.5, and three possible spanning trees of the graph shown in Figure 2.1.

In view of Lemma 2.24, the “maximal” objects of the polytropical decomposition (cf. Section 2.3.2) of \mathcal{T} hence correspond to certain 0-dimensional objects of cycle offset zonotope Z . We will investigate now how polytrope vertices as 0-dimensional objects relate to the tiles of a zonotopal tiling as top-dimensional objects. We understand these two relations as a kind of duality, but we do not expect that this

extends to all dimensions in between, as the polytopes have dimension at most $|V(G)| - 1$, whereas the zonotope tiles have dimension μ .

We will start with investigating the faces of the hyperrectangle C . For the timetabling application, we choose $C = X_{LP}$, the fractional periodic tension polytope as introduced in (2.29). Since C is attached to a graph, we will describe the cube combinatorics in graph terminology:

Definition 2.15. — *A structure (S, L, U) in G is a triplet of subsets of $A(G)$ such that $S = L \cup U$ and $L \cap U = \emptyset$. If S spans G , we say (S, L, U) is a spanning structure. If $|S| = |V(G)| - 1$, we say (S, L, U) is a spanning tree structure.*

Lemma 2.26. — *The map*

$$(S, L, U) \mapsto \mathcal{F}_{L,U} := \{x \in C \mid x_a = \ell_a \ \forall a \in L, x_a = u_a \ \forall a \in U\} \quad (2.44)$$

is an inclusion-reversing bijection between structures in G and faces of C . A structure with $|S| = k$ corresponds to a face of dimension $|A(G)| - k$.

Observation 2.5. — *Any spanning tree structure may be associated to a distinct spanning tree of \overline{G} . Consequently, by Theorem 2.6, spanning tree structures correspond to vertices in polytopes in \mathcal{T} . Additionally, by Theorem 2.12, the latter are embedded in X , and so spanning tree structures also correspond to vertices in X .*

In these terms, this means that if x is an extreme point of X , it must lie on one $\mathcal{F}_{L,U}$ for some spanning tree structure (S, L, U) .

Remark 2.2. — *For readers that find themselves comfortable with the notions of oriented matroids, one can identify a structure (S, L, U) with the signed covector $\sigma \in \{+, -, 0\}^{A(G)}$, where*

$$\sigma_a := \begin{cases} - & \text{if } a \in L, \\ + & \text{if } a \in U, \\ 0 & \text{otherwise,} \end{cases} \quad \text{for all } a \in A(G). \quad (2.45)$$

In this interpretation, structures correspond to the covectors of the oriented matroid associated to the central hyperplane arrangement given by the coordinate hyperplanes

$x_a = 0$ for all $a \in A(G)$. The poset of these covectors is anti-isomorphic to the face lattice of a cube, and hence to the face lattice of X_{LP} [53]. However, we prefer the notation (S, L, U) , as it is directly related to the features of the PESP instance (G, T, ℓ, u) .

The cycle offset zonotope Z is by definition the image of C under $\frac{1}{T}\Gamma$. As the faces $\mathcal{F}_{L,U}$ are hyperrectangles themselves, $\frac{1}{T}\Gamma$ maps each $\mathcal{F}_{L,U}$ to a subzonotope of Z . A straightforward computation shows the following:

Lemma 2.27. — *Let (S, L, U) be a structure. Then*

$$\frac{1}{T}\Gamma(\mathcal{F}_{L,U}) = Z\left(\Gamma_{d,S}, \frac{\Gamma v}{T}\right), \quad (2.46)$$

where $v_a := u_a$ for $a \in U$ and $v_a := \ell_a$ otherwise.

Proof. We expand the left side of the equation, extract the fixed columns, which are the ones in S , and appropriately set the translation vector to be the base translation $\frac{\Gamma \ell}{T}$ plus a component given by the arcs in U fixed at the upper bound. Let v be as defined above, and we have

$$\left\{ \frac{\Gamma x}{T} \mid x \in \mathcal{F}_{L,U} \right\} = \left\{ \frac{\Gamma x}{T} \mid \ell \leq x \leq u, x_a = \ell_a \forall a \in L, x_a = u_a \forall a \in U \right\} \quad (2.47)$$

$$= \{\Gamma_{d,S} \cdot c \mid \mathbf{0} \leq c \leq \mathbf{1}\} + \frac{\Gamma \ell}{T} + \frac{\Gamma(v - \ell)}{T} \quad (2.48)$$

$$= Z(\Gamma_{d,S}) + \frac{\Gamma v}{T} \quad (2.49)$$

$$= Z\left(\Gamma_{d,S}, \frac{\Gamma v}{T}\right). \quad (2.50)$$

□

We will now tie faces of the cube C to cells of a zonotopal tiling of the cycle offset zonotope. This is similar in spirit to [60, 61], however, we use an a priori more general definition of fine zonotopal tilings and work directly toward the cographic zonotope. Let K be a cell of some zonotopal tiling of $Z = Z(\Gamma_d, \frac{\Gamma \ell}{T})$. Then, by Lemma 2.19, K corresponds to a spanning subgraph S in the following sense: If S is the set of arcs of that subgraph, then K is a translate of $Z(\Gamma_{d,S})$ contained in Z . On the other

hand, Lemma 2.27 yields an assignment of spanning structures (S, L, U) for the very same S to translates of $Z(\Gamma_{d,S})$ contained in Z . The purpose of the following lemma is to show that K can be rediscovered not only up to translation, but exactly as the image under $\frac{1}{T}\Gamma$ of some face $\mathcal{F}_{L,U}$ with $L \cup U = S$ as in Lemma 2.27.

Lemma 2.28. — *Let K be a cell of a fine zonotopal tiling of Z such that K is a translate of $Z(\Gamma_{d,S})$ for some $S \subseteq A(G)$. Then there exist L and U such that (S, L, U) is a structure and $K = \frac{1}{T}\Gamma(\mathcal{F}_{L,U})$.*

Proof. As discussed above, this is only a question of the correct translation vector. We will make use of the result [62, Theorem 3.3] that any fine zonotopal tiling of a centrally symmetric zonotope can be turned into a so-called strong zonotopal tiling. A central symmetric zonotope is a set of the form

$$Z_{\text{sym}}(A) := \{Ac \mid -\mathbf{1} \leq c \leq \mathbf{1}\} \quad (2.51)$$

for some real matrix $A \in \mathbb{R}^{\mu \times m}$. Clearly, $Z_{\text{sym}}(A) = Z(2A, -A\mathbf{1})$ is a zonotope in the sense of Definition 2.11. Consider a fine zonotopal tiling on $Z_{\text{sym}}(A)$. By Definition 2.12, each cell K is then a translate of $Z(2A_S)$, hence a translate of $Z_{\text{sym}}(A_S)$. Then, using [62, Definition 1.2 and Theorem 3.3], one can assign to K a signed covector $\sigma \in \{+, -, 0\}^m$ such that $\sigma_i \neq 0$ if and only if $i \in S$ and

$$K = Z_{\text{sym}}(A_S) + \sum_{i: \sigma_i = +} a_i - \sum_{i: \sigma_i = -} a_i, \quad (2.52)$$

where (a_1, \dots, a_m) are the columns of A .

We apply now this result to our setting. Let K be a cell of a fine zonotopal tiling of the cycle offset zonotope Z such that K is a translate of $Z(\Gamma_{d,S})$ for some $S \subseteq A(G)$. Observe that $Z_{\text{sym}}(\Gamma_d) = 2(Z - \frac{\Gamma \ell}{T}) - \Gamma_d \mathbf{1}$, so that $2(K - \frac{\Gamma \ell}{T}) - \Gamma_d \mathbf{1}$ is a cell of a fine zonotopal tiling of $Z_{\text{sym}}(\Gamma_d)$. Using the interpretation of signed covectors in terms of structures (Remark 2.2) and evaluating (2.52), we find a structure (S, L, U) such that

$$2\left(K - \frac{\Gamma \ell}{T}\right) - \Gamma_d \mathbf{1} = 2Z(\Gamma_{d,S}) - \Gamma_{d,S} \mathbf{1} + \sum_{a \in U} \Gamma_{d,a} - \sum_{a \in L} \Gamma_{d,a}, \quad (2.53)$$

where $\Gamma_{d,a}$ is the column of Γ_d labeled by $a \in A(G)$. Hence

$$K = Z(\Gamma_{d,S}) + \frac{\Gamma\ell}{T} + \frac{1}{2} \left(\sum_{a \in U} \Gamma_{d,a} - \sum_{a \in L} \Gamma_{d,a} + \Gamma_d \mathbf{1} - \Gamma_{d,S} \mathbf{1} \right) \quad (2.54)$$

$$= Z(\Gamma_{d,S}) + \frac{\Gamma\ell}{T} + \Gamma \left(\sum_{a \in U} \frac{u_a - \ell_a}{2T} \cdot e_a - \sum_{a \in L} \frac{u_a - \ell_a}{2T} \cdot e_a + \sum_{a \in S} \frac{u_a - \ell_a}{2T} \cdot e_a \right) \quad (2.55)$$

$$= Z(\Gamma_{d,S}) + \frac{\Gamma\ell}{T} + \Gamma \left(\sum_{a \in U} \frac{u_a - \ell_a}{T} e_a \right) \quad (2.56)$$

$$= Z(\Gamma_{d,S}) + \frac{\Gamma v}{T}, \quad (2.57)$$

with v as defined in Lemma 2.27, and e_a denoting the standard basis vector corresponding to $a \in A(G)$. \square

We are now ready to formulate a duality result between tiles of a zonotopal tiling and polytrope vertices.

Theorem 2.29. — *Fix an arbitrary fine zonotopal tiling of Z . Then any tile is the image of $\mathcal{F}_{L,U}$ under $\frac{1}{T}\Gamma$ for a spanning tree structure (S, L, U) . If such a tile contains an integer point $z \in Z$, then the polytrope $\mathfrak{R}(z)$ has a vertex defined by (S, L, U) .*

Proof. The first statement follows from Lemma 2.19 and Lemma 2.28. If a tile corresponding to (S, L, U) contains a lattice point z , then we find $x \in \mathcal{F}_{L,U}$ with $\Gamma x = Tz$. But then x is a vertex of X and by Lemma 2.10 and Theorem 2.6 also a vertex of $\mathfrak{R}(z)$. \square

Example 2.11. — *Consider the middle tile in Figure 2.15. As discussed in Example 2.8, this tile corresponds to the spanning tree given by the arcs $S_{02} = \{01, 12\}$. Later, in Example 2.12, we will see that it arises from the spanning tree structure (S_{02}, L, U) with $L = \{01\}$ and $U = \{12\}$, meaning that the corresponding translation vector as in Lemma 2.27 is given by $\frac{\Gamma v}{T} = 1.4$, with $v = (3, 2, 13)$. Consequently, the central tile is then $Z\left(\Gamma_{d,S_{02}}, \frac{\Gamma v}{T}\right) = 1.4 + [-0.8, 0]$. As the tile contains an integer point, we know it is also related to a feasible polytrope in the polytropical decomposition. The associated polytrope is, in fact, the hexagon in Figure 2.4, and here we see that*

the polytrope contains the vertex $(3, 6)$, which corresponds to the same spanning tree. In particular, the spanning tree is a shortest path tree rooted at v_1 in Figure 2.3, and the vertex of the polytrope is a tropical vertex.

Now, to complement Theorem 2.23, we have the following statement.

Corollary 2.30. — *If all the arc bounds of the PESP instance are free, i.e. $u - \ell = \mathbf{T} - \mathbf{1}$, and there is no degeneracy, meaning that all spanning structures for which a feasible solution exists, are spanning tree structures, then the bound expressed in Theorem 2.23 is tight.*

Proof. It is well known that if all arcs are free, then for any spanning tree structure there exists a feasible solution [63]. This implies that given an instance with only free arcs, then any tile of any tiling of its cycle offset zonotope Z must contain an integer point. Now, if an integer point in Z is shared by two distinct tiles, this implies by Lemma 2.28 that the two corresponding distinct spanning tree structures are simultaneously tight for the same associated solution, thereby giving the kind of degeneracy we excluded in our hypothesis. We have therefore shown that the tiling of a free instance that has no degeneracy must have each tile containing a point in its interior, concluding the proof. \square

2.4.4 Constructing Zonotopal Tilings

It is natural to ask for a converse of Theorem 2.29: Suppose that for each non-empty polytrope $\mathfrak{R}(z)$ we pick a vertex corresponding to a spanning tree structure (S, L, U) and map $\mathcal{F}_{L,U}$ to the cycle offset zonotope Z via $\frac{1}{T}\Gamma$. When we choose pairwise distinct spanning trees per polytrope, do these μ -dimensional zonotopes form the tiles of a zonotopal tiling of Z ? We will describe a construction that produces a zonotopal tiling from certain spanning tree structures.

Definition 2.16. — *Let G be a directed graph and let $i \in V(G)$. Denote by B the incidence matrix of G . Let $C := [\ell, u] \subseteq \mathbb{R}^{A(G)}$ be a hyperrectangle. We define the unbounded polyhedron*

$$P_i := C + \text{pos}(\widehat{B^\top}_i), \quad (2.58)$$

where $\widehat{B^\top}_i$ is B^\top without the i -th column, $\text{pos}(\widehat{B^\top}_i)$ indicates the positive cone of $\widehat{B^\top}_i$, and the summation is the Minkowski sum.

To start it is practical to recall that $\text{rank } B^\top = n - 1$, where $n = |V(G)|$. Moreover we assumed the graph G to be connected and thereby every column b_i of B^\top is non-zero, and so we have that $-b_i \in \text{pos}(\widehat{B^\top}_i)$ and $\widehat{B^\top}_i$ has full rank for every i . From this we conclude that $\text{pos}(\widehat{B^\top}_i)$ is a $(n - 1)$ -dimensional pointed cone.

Let \mathcal{B} be an integral cycle basis of G with cycle matrix Γ . Let $T \in \mathbb{N}$ with $\ell < u + \mathbf{T}$, $d := \frac{u-\ell}{T}$ and let Z be the cycle offset zonotope as in the previous section. Observe that $\frac{1}{T}\Gamma$ maps P_i to Z , due to $C \subset P_i$ and Theorem 2.2. It will turn out that the bounded faces of P_i produce a fine zonotopal tiling of Z , the μ -dimensional bounded faces giving the tiles. We need a few preparatory lemmata to prove this result.

Lemma 2.31. — *Every bounded face of P_i is a face of C .*

Proof. Let F be a bounded face of P_i . Then there are $d \in \mathbb{R}^{A(G)}$ and $e \in \mathbb{R}$ such that $d^\top x \geq e$ is valid for P_i and $F = \{x \in P_i \mid d^\top x = e\}$. Since $C \subseteq P_i$, $d^\top x \geq e$ is also valid for C , so that $F' := \{x \in C \mid d^\top x = e\}$ is a face of C . Clearly $F' \subseteq F$. Conversely, if $x \in F$, then there are $x' \in C$ and $p \in \text{pos}(\widehat{B^\top}_i)$ such that $x = x' + p$. Then $d^\top x' + d^\top p = e$ and since $d^\top x' \geq e$, $d^\top p \leq 0$. But then $d^\top x + \lambda d^\top p \leq e$ for every $\lambda \in \mathbb{R}_{\geq 0}$, and $d^\top x + \lambda d^\top p \geq e$ due to $x + \lambda p \in P_i$. We conclude that $x + \lambda p \in F$ for all $\lambda \in \mathbb{R}_{\geq 0}$. Since F is bounded, we must have $p = \mathbf{0}$ and hence $x = x' \in F'$. This shows $F \subseteq F'$. \square

We can give a combinatorial description of the bounded faces of P_i :

Lemma 2.32. — *A face $\mathcal{F}_{L,U}$ of C is a bounded face of P_i if and only if there is a vector $f \in \mathbb{R}_{\geq 0}^{A(G)}$ such that $f_a > 0$ for all $a \in L \cup U$, $f_a = 0$ otherwise, and*

$$\forall j \in V(G) \setminus \{i\}: \sum_{a \in \delta^+(j) \cap U} f_a - \sum_{a \in \delta^+(j) \cap L} f_a - \sum_{a \in \delta^-(j) \cap U} f_a + \sum_{a \in \delta^-(j) \cap L} f_a < 0. \quad (2.59)$$

Proof. Let (S, L, U) be a structure. We observe first that the inequality

$$\sum_{a \in U} f_a x_a - \sum_{a \in L} f_a x_a \leq \sum_{a \in U} f_a u_a - \sum_{a \in L} f_a \ell_a \quad (2.60)$$

is valid for C , and

$$\mathcal{F}_{L,U} = \left\{ x \in C \mid \sum_{a \in U} f_a x_a - \sum_{a \in L} f_a x_a = \sum_{a \in U} f_a u_a - \sum_{a \in L} f_a \ell_a \right\} \quad (2.61)$$

for any vector $f \in \mathbb{R}_{\geq 0}^{A(G)}$ with $f_a > 0$ for all $a \in S$ and $f_a = 0$ otherwise.

If $\mathcal{F}_{L,U}$ is a bounded face of P_i , then f can be chosen in such a way that

$$\sum_{a \in U} f_a x_a - \sum_{a \in L} f_a x_a < 0 \quad (2.62)$$

for all generators x of the cone $\text{pos}(\widehat{B^\top}_i)$. Conversely, if we find such a f satisfying (2.62), then $\mathcal{F}_{L,U}$ is bounded in P_i .

The cone $\text{pos}(\widehat{B^\top}_i)$ is generated by the columns of B^\top except the i -th column. By definition of the incidence matrix, (2.62) is equivalent to (2.59). \square

Recall that we constructed the graph \overline{G} in Definition 2.8, where we added reverse copies of arcs to G . In this language, Lemma 2.32 states:

Corollary 2.33. — *A face $\mathcal{F}_{L,U}$ of C is a bounded face of P_i if and only if there is a positive flow with negative balances at all vertices $v \in V(G) \setminus \{i\}$ in the subgraph $\overline{G}_{L,U}$ of \overline{G} with $A(\overline{G}_{L,U}) := \{(j, k) \mid (j, k) \in U\} \cup \{(k, j) \mid (j, k) \in L\}$ and $V(\overline{G}_{L,U}) := V(\overline{G}) = V(G)$.*

Let $\mathcal{F}_{L,U}$ be a bounded face of P_i . Then we find a flow as in the statement of Corollary 2.33 in the subgraph $\overline{G}_{L,U}$. Since the balances of every flow sum up to 0 on each component of $\overline{G}_{L,U}$, and only the balance at node i can be negative, we must have that $\overline{G}_{L,U}$ is connected and spanning. This has the following consequence:

Corollary 2.34. — *Let $\mathcal{F}_{L,U}$ be a bounded face of P_i . Then the corresponding structure (S, L, U) is spanning. In particular, the dimension of $\mathcal{F}_{L,U}$ is at most μ .*

Theorem 2.35. — *There is a bijection between the μ -dimensional bounded faces of P_i and arborescences in \overline{G} rooted at i , given by $\mathcal{F}_{L,U} \mapsto \overline{G}_{L,U}$.*

Proof. Let (S, L, U) be a structure such that $\overline{G}_{L,U}$ is an arborescence rooted at i . Define $f_{(j,k)} := n^{n-d_k}$, where $n = |V(\overline{G}_{L,U})| = |V(G)|$ and d_j is the distance from i to j in $\overline{G}_{L,U}$, i.e., the number of edges of the unique i - j -path in $\overline{G}_{L,U}$. Since every vertex $j \neq i$ has exactly one ingoing arc and at most $n - 1$ outgoing arcs in $\overline{G}_{L,U}$,

$$\sum_{a \in \delta^+(j)} f_a - \sum_{a \in \delta^-(j)} f_a = |\delta^+(j)|n^{n-d_j-1} - n^{n-d_j} < n \cdot n^{n-d_j-1} - n^{n-d_j} = 0. \quad (2.63)$$

Here there is a positive flow in $A(\overline{G}_{L,U})$ with negative balance at all vertices except i , and we conclude by Corollary 2.33 that $\mathcal{F}_{L,U}$ is a bounded face of P_i .

For the converse, suppose that $\mathcal{F}_{L,U}$ is a bounded face of P_i of dimension μ . Then Corollary 2.33 yields the existence of a positive flow f in the spanning tree $\overline{G}_{L,U}$ with negative balances at all vertices except i . For $j \in V(\overline{G}_{L,U})$, let d_j again denote its distance from i (the number of edges of the unique undirected i - j -path in $\overline{G}_{L,U}$). Let $d_{\max} := \max\{d_j \mid j \in V(\overline{G}_{L,U})\}$. For $j \neq i$, we denote by $\text{pred}(j)$ the unique vertex k with $d_k = d_j - 1$ to which j is adjacent.

We prove by induction on $d_{\max} - d_j$ that for all vertices $j \neq i$ the arc $(\text{pred}(j), j)$ is the unique ingoing arc at j in $\overline{G}_{L,U}$. This is sufficient to show that $\overline{G}_{L,U}$ is an arborescence. In the case $d_j = d_{\max}$, j is a leaf of the tree $\overline{G}_{L,U}$. Since the balance of f is negative, the unique arc incident with j must be ingoing at j . If $0 < d_j < d_{\max}$, then all arcs that connect j with vertices of higher distance must be outgoing at j by induction hypothesis. Since the balance of f is negative, the unique arc that connects j with a vertex of lower distance must be ingoing, so that $(\text{pred}(j), j)$ is indeed the unique ingoing arc at j . \square

Lemma 2.36. — *For any k -dimensional bounded face $\mathcal{F}_{L,U}$ of P_i , the image $\frac{1}{T}\Gamma(\mathcal{F}_{L,U})$ is a k -dimensional parallelotope.*

Proof. Since $\mathcal{F}_{L,U}$ is a hyperrectangle, it is enough to show that the restriction of Γ to $\mathcal{F}_{L,U}$ is injective. By Corollary 2.34, $\mathcal{F}_{L,U}$ corresponds to a spanning structure (S, L, U) . In particular, there is a spanning tree S' contained in S . If Γ happens to be the cycle matrix of the fundamental cycles $\gamma_1, \dots, \gamma_\mu$ of S' , the injectivity is clear: For any $x \in \mathcal{F}_{L,U}$, the entries corresponding to arcs of S' are the same, so

that x is determined by its entries for the co-tree arcs of S' . But the k -th entry $\gamma_k^\top x$ of Γx is

$$\sum_{a \in S': \gamma_a \neq 0} \gamma_{k,a} x_a + \sum_{a \notin S': \gamma_a \neq 0} \gamma_{k,a} x_a. \quad (2.64)$$

The first sum is constant, and the second sum only has the summand $\gamma_{k,a'} x_{a'}$ for the unique co-tree arc a' of S' contained in the fundamental cycle γ_k . Since there is one fundamental cycle for each co-tree arc, Γ is injective on $\mathcal{F}_{L,U}$.

If Γ is another integral cycle basis, then there is a \mathbb{Z} -invertible matrix M such that $M\Gamma$ is the cycle matrix of a fundamental cycle basis for S' . Since $M\Gamma|_{\mathcal{F}_{L,U}}$ is injective by the above argument, so is $\Gamma|_{\mathcal{F}_{L,U}}$. \square

Theorem 2.37. — *The bounded faces of P_i are mapped via $\frac{1}{T}\Gamma$ to the tiles of a fine zonotopal tiling of Z .*

Proof. Let \mathcal{F}_i denote the union of all bounded faces of P_i . As $C \subseteq P_i$ and $\Gamma(\text{pos}(\widehat{B^\top}_i)) = 0$ by Theorem 2.2, $\frac{1}{T}\Gamma(\mathcal{F}_i) = \frac{1}{T}\Gamma(P_i) = Z$. In particular,

$$\begin{aligned} \text{vol}(Z) &= \text{vol} \left(\bigcup_{\substack{\mathcal{F}_{L,U} \subseteq \mathcal{F}_i \\ \dim \mathcal{F}_{L,U} = \mu}} \frac{1}{T}\Gamma(\mathcal{F}_{L,U}) \right) \\ &\leq \sum_{\substack{\mathcal{F}_{L,U} \subseteq \mathcal{F}_i \\ \dim \mathcal{F}_{L,U} = \mu}} \text{vol} \left(\frac{1}{T}\Gamma(\mathcal{F}_{L,U}) \right). \end{aligned} \quad (2.65)$$

By Theorem 2.35, the μ -dimensional bounded faces of P_i correspond one-to-one with arborescences in $\overline{G}_{L,U}$ rooted at i . Any such arborescence yields a spanning tree structure (S, L, U) , and conversely, for every spanning tree S of G , there is a unique spanning tree structure (S, L, U) that turns S into an arborescence on $\overline{G}_{L,U}$. With Lemma 2.27, (2.65) implies

$$\text{vol}(Z) \leq \sum_{S \in \mathcal{S}(G)} \text{vol}(Z(\Gamma'_S)), \quad (2.66)$$

and by the proof of Corollary 2.20, (2.66) and (2.65) are in fact equations. In particular, the images of two distinct bounded μ -dimensional faces $\mathcal{F}_{L,U}$ have an intersection of dimension strictly less than μ . It remains to remark that by Lemma 2.36, $\frac{1}{T}\Gamma(\mathcal{F}_{L,U})$ is always a parallelotope. \square

We return to the decomposition of the torus \mathcal{T} into polytropes $\mathfrak{R}(z)$. Any vertex of $\mathfrak{R}(z)$ is defined by a spanning tree structure by Theorem 2.6. Let $i \in V(G)$. Then the i -th tropical vertex is defined by the spanning tree structure corresponding to the shortest path arborescence rooted at i in $\overline{G}_{L,U}$. For any $i \in V(G)$, we can hence assign to any polytrope $\mathfrak{R}(z)$ a spanning structure (S_z^i, L_z^i, U_z^i) such that $\mathcal{F}_{L_z^i, U_z^i}$ is a bounded face of P_i . As a result, we obtain a partial converse to Theorem 2.29:

Theorem 2.38. — *Let $i \in V(G)$. Then there is a fine zonotopal tiling of Z such that any tile containing an integer point z corresponds to a spanning tree structure defined by the i -th tropical vertex of $\mathfrak{R}(z)$.*

Example 2.12. — *To illustrate the properties above, we again refer to our running example, Example 2.1. Dropping the column of vertex v_1 of B^\top results in the unbounded polyhedron as displayed in Figure 2.16. The coloured arrows depict the cone generators corresponding to the columns of vertices v_0 and v_2 in B^\top . The bounded faces are the three blue edges of the cube corresponding to the spanning tree structures (S_i, L_i, U_i) for the pairs*

$$\begin{aligned} L_0 &= \{01\}, U_0 = \{02\}, \\ L_1 &= \{01\}, U_1 = \{12\}, \\ L_2 &= \{02\}, U_2 = \{12\}, \end{aligned} \tag{2.67}$$

respectively. Mapping each of those faces via $\frac{1}{T}\Gamma$ results in the three tiles as seen in Figure 2.15. In this fine zonotopal tiling of Z , each tile contains an integer point, namely $z = 0$, $z = 1$ and $z = 2$ respectively. As we saw with Theorem 2.5, the i -th tropical vertices in each $\mathfrak{R}(z)$ are identified with the shortest path arborescences rooted at v_i in the corresponding weighted digraphs. In Example 2.4 the v_1 -rooted arborescences were discussed. The arcs used in each such arborescence identify the sets L_i and U_i that define the bounded face of the corresponding cycle offset. Again in Example 2.4 three timetables arose. With the appropriate value of z these three timetables are embedded via \mathbf{m}_z into C , precisely to the three coloured bounded faces in Figure 2.16.

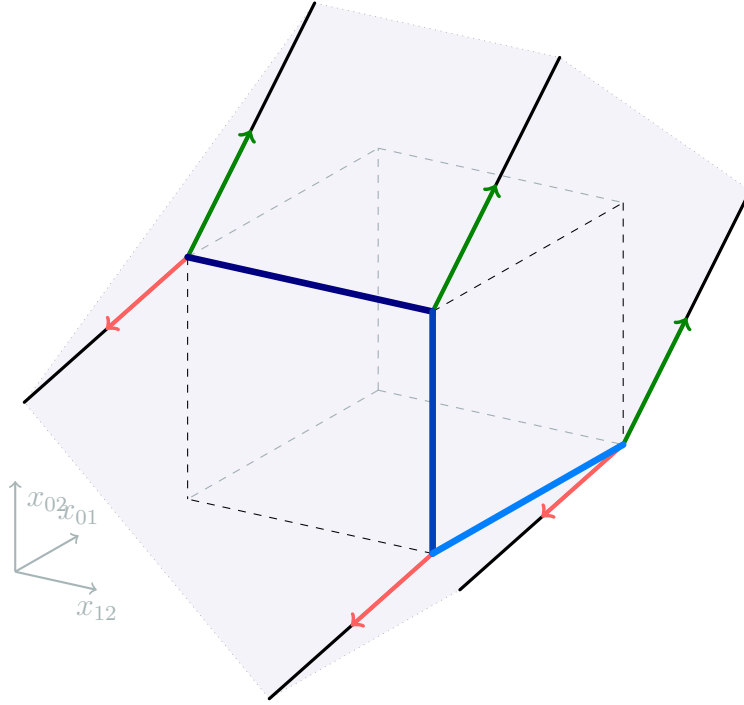


Figure 2.16: The construction, according to Definition 2.16, of the object P_1 for the instance of Example 2.1.

2.4.5 Approximating the minimum width of a cycle basis

The following is an application to a computational phenomenon that arises specifically in the context of periodic timetabling. For every choice of an integral cycle basis, the number of lattice points in the cycle offset zonotope Z is the same by Lemma 2.24. However, in order to solve the cycle-based MIP formulation for PESP (2.5), it has been observed that the choice of a cycle basis has an impact on the performance of branch-and-cut-based solvers.

Definition 2.17. — *Let (G, T, ℓ, u) be a PESP instance. Then, following [58], the width of an integral cycle basis \mathcal{B} of G is defined as*

$$W_{\mathcal{B}} := \prod_{\gamma \in \mathcal{B}} \left(\left\lfloor \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T} \right\rfloor - \left\lceil \frac{\gamma_+^\top \ell - \gamma_-^\top u}{T} \right\rceil + 1 \right). \quad (2.68)$$

Here, we decompose the oriented cycles $\gamma \in \mathcal{B}$ into their positive and negative parts $\gamma_+ := \max(\mathbf{0}, \gamma)$ and $\gamma_- := \max(\mathbf{0}, -\gamma)$, respectively.

The rationale behind this notion is the following: For any feasible solution $(x, z) \in \mathbb{R}^{A(G)} \times \mathbb{Z}^{\mathcal{B}}$ to (2.5) and for $\gamma \in \mathcal{B}$, it follows from $\ell \leq x \leq u$ that

$$\frac{\gamma_+^\top \ell - \gamma_-^\top u}{T} \leq z_\gamma = \frac{\gamma^\top x}{T} \leq \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T}. \quad (2.69)$$

Since z_γ is integer, we can round up the left-hand side and round down the right-hand side, so that there are at most

$$\left\lceil \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T} \right\rceil - \left\lfloor \frac{\gamma_+^\top \ell - \gamma_-^\top u}{T} \right\rfloor + 1 \quad (2.70)$$

possible integer values that the variable z_γ can attain.

The width $W_{\mathcal{B}}$ is an upper bound on the number of lattice points in the cycle offset zonotope Z for \mathcal{B} . Moreover, it is an upper bound on the number of leaves of the branch-and-bound-tree for the MIP (2.5), and this bound is sharp if the tree is fully explored. These considerations lead to the following optimization problem:

Definition 2.18. — *Let (G, T, ℓ, u) be a PESP instance. The minimum width integral cycle basis problem is to find an integral cycle basis \mathcal{B} of minimum width $W_{\mathcal{B}}$.*

The complexity of this problem is open. It turns out that the minimum width integral cycle basis problem can be approximated by finding a minimum weight integral cycle basis, where the weight of an oriented cycle is

$$\sum_{a \in A(G): \gamma_a \neq 0} (u_a - \ell_a), \quad (2.71)$$

see [45]. Unfortunately, also this version has an unresolved complexity status, as finding a minimum weight fundamental cycle basis is NP-hard, whereas the problem is solvable in polynomial time on the larger class of undirected cycle bases [64].

The number of lattice points in Z is not only bounded by $W_{\mathcal{B}}$, but also by $|\mathcal{S}(G)|$, by Theorem 2.23. The aim of this section is to explore how far the number of spanning trees can be from the minimum width of an integral cycle basis.

Let (G, T, ℓ, u) be a PESP instance, \mathcal{B} an integral cycle basis of G , and let Z be the corresponding cycle offset zonotope in $\mathbb{R}^{\mathcal{B}}$. By (2.69), we see that Z is contained in the hyperrectangle

$$R := \prod_{\gamma \in \mathcal{B}} \left[\frac{\gamma_+^\top \ell - \gamma_-^\top u}{T}, \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T} \right]. \quad (2.72)$$

In fact, R is the smallest hyperrectangle containing Z : For a specific $\gamma \in \mathcal{B}$, taking the vector $x \in C$ with $x_a := u_a$ if $\gamma_a > 0$ and $x_a := \ell_a$ otherwise produces a point $z := \frac{\Gamma x}{T} \in Z$ with $z_\gamma = \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T}$, and the argument for the lower bound in the interval is analogous. With the rounding argument above (2.70), we immediately find:

Lemma 2.39. — *The width $W_{\mathcal{B}}$ equals the number of lattice points in R .*

Example 2.13. — *Consider the instance of Example 2.9. Recall that the number of lattice points of the cycle offset zonotope is 11, as depicted in Figure 2.14. The width of the cycle basis induced by the three regions of the planar embedding is*

$$\begin{aligned} W_{\mathcal{B}} &= \left(\left\lfloor \frac{15+12}{10} \right\rfloor - \left\lceil \frac{6+3}{10} \right\rceil + 1 \right) \\ &\quad \cdot \left(\left\lfloor \frac{12-3+12-3}{10} \right\rfloor - \left\lceil \frac{3-12+3-12}{10} \right\rceil + 1 \right) \\ &\quad \cdot \left(\left\lfloor \frac{12+13}{10} \right\rfloor - \left\lceil \frac{3+4}{10} \right\rceil + 1 \right) \\ &= 2 \cdot 3 \cdot 2 = 12. \end{aligned} \quad (2.73)$$

Indeed, 12 is the number of lattice points in the hyperrectangle

$$R = \left[\frac{9}{10}, \frac{27}{10} \right] \times \left[\frac{-18}{10}, \frac{18}{10} \right] \times \left[\frac{7}{10}, \frac{25}{10} \right]. \quad (2.74)$$

We hence want to compare the lattice points in Z with the lattice points in R . We will do so by relating both with their volumes.

Lemma 2.40. — *Let $d \in \mathbb{R}_{\geq 0}^{A(G)}$. Then*

$$\sum_{S \in \mathcal{S}(G)} \left(\prod_{a \in A(G) \setminus A(S)} d_a \right) \leq \prod_{\gamma \in \mathcal{B}} \left(\sum_{a \in A(G): \gamma_a \neq 0} d_a \right). \quad (2.75)$$

Proof. Let Γ be the cycle matrix of \mathcal{B} . Proceeding as in the proof of Corollary 2.20, the left-hand side is the volume of the scaled cographic zonotope $Z(\Gamma_d)$. The hyperrectangle

$$\prod_{\gamma \in \mathcal{B}} [-\gamma_-^\top d, \gamma_+^\top d] \quad (2.76)$$

contains $Z(\Gamma_d)$, and its volume is

$$\prod_{\gamma \in \mathcal{B}} (\gamma_+ + \gamma_-)^\top d = \prod_{\gamma \in \mathcal{B}} \left(\sum_{a \in A(G): \gamma_a \neq 0} d_a \right), \quad (2.77)$$

the right-hand side in the statement. \square

A beautiful consequence of Lemma 2.40 for $d = \mathbf{1}$ is:

Theorem 2.41. — *For any directed graph G and any integral cycle basis \mathcal{B} of G , the number of spanning trees in G is at most $\prod_{\gamma \in \mathcal{B}} |\{a \in A(G) \mid \gamma_a \neq 0\}|$, i.e., the product of the lengths of the oriented cycles in \mathcal{B} .*

Returning to PESP, we find:

Theorem 2.42. — *Suppose that $W_{\mathcal{B}} \geq 1$. Then*

$$|\mathcal{S}(G)| \cdot \left(\frac{\varepsilon}{T}\right)^\mu \leq \text{vol}(Z) \leq \prod_{\gamma \in \mathcal{B}} s_\gamma \leq W_{\mathcal{B}} \cdot \prod_{\gamma \in \mathcal{B}} \frac{s_\gamma}{\max\{\lfloor s_\gamma \rfloor, 1\}} < W_{\mathcal{B}} \cdot 2^\mu, \quad (2.78)$$

where $\varepsilon := \min\{u_a - \ell_a \mid a \in A(G)\}$, and

$$s_\gamma := \sum_{a \in A(G): \gamma_a \neq 0} \frac{u_a - \ell_a}{T}. \quad (2.79)$$

Proof. By Corollary 2.20,

$$\text{vol}(Z) = \sum_{S \in \mathcal{S}(G)} \left(\prod_{a \in A(G) \setminus A(S)} \frac{u_a - \ell_a}{T} \right) \geq \sum_{S \in \mathcal{S}(G)} \left(\frac{\varepsilon}{T}\right)^\mu = |\mathcal{S}(G)| \cdot \left(\frac{\varepsilon}{T}\right)^\mu. \quad (2.80)$$

On the other hand, by Lemma 2.40,

$$\text{vol}(Z) \leq \text{vol}(R) = \prod_{\gamma \in \mathcal{B}} s_\gamma. \quad (2.81)$$

Since for non-negative s_γ we have $s_\gamma < 2 \max\{\lfloor s_\gamma \rfloor, 1\}$, it remains to show $W_{\mathcal{B}} \geq \prod_{\gamma \in \mathcal{B}} \max\{\lfloor s_\gamma \rfloor, 1\}$. Let $\gamma \in \mathcal{B}$. Then, since $W_{\mathcal{B}} \geq 1$, we must have that

$$\left\lfloor \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T} \right\rfloor - \left\lfloor \frac{\gamma_+^\top \ell - \gamma_-^\top u}{T} \right\rfloor + 1 \geq 1. \quad (2.82)$$

Moreover, using the properties of the ceiling and floor functions

$$\left\lceil \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T} \right\rceil - \left\lfloor \frac{\gamma_+^\top \ell - \gamma_-^\top u}{T} \right\rfloor + 1 > \frac{\gamma_+^\top u - \gamma_-^\top \ell}{T} - \frac{\gamma_+^\top \ell - \gamma_-^\top u}{T} - 1 = s_\gamma - 1, \quad (2.83)$$

and we conclude since the left-hand side is an integer. \square

The hypothesis $W_{\mathcal{B}} \geq 1$ is satisfied for all feasible PESP instances: If $W_{\mathcal{B}} = 0$, then R and hence Z do not contain an integer point. If the bounds ℓ and u are integral, having assumed $u - \ell > \mathbf{0}$, then $\varepsilon \geq 1$. However, there is also a large class of practical PESP instances, where each arc $a \in A(G)$ is either fixed ($\ell_a = u_a$) or free ($u_a - \ell_a = T - 1$), coined *reduced* instances in [39]. As the former can be contracted, such that only the free arcs remain, while the cyclomatic number μ is unchanged, Theorem 2.42 then yields that the minimum width of an integral cycle basis is at least $|\mathcal{S}(G)| \left(\frac{T-1}{2T}\right)^\mu$ for these instances. We finally want to remark that the number of spanning trees and the minimum width of an integral cycle basis are exponential in the input size of a PESP instance, so that it makes sense to compare their logarithms. Then Theorem 2.42 states that the logarithms of both numbers differ by $\mu \log(\varepsilon/(2T))$.

2.5 Outlook

Given that the tropical neighborhood search heuristic has been shown to be valuable in practice [2], a related question is how the insights on the cycle offset zonotope (Section 2.4) can be exploited for the purpose of optimization. For example, we expect that these zonotopes are related to the space of feasible solutions of a reformulation of PESP by means of Benders' decomposition [65]. The estimates on the cycle basis width in Section 2.4.5 motivate further investigation of approximation algorithms or hardness results for the minimum width cycle basis problem.

Finally, the understanding of cycle offset zonotopes may be improved. It would be interesting to characterize all fine zonotopal tilings in terms of the polytropical decomposition, or by constructions similar to the one in Section 2.4.4.

Authors' Contributions

Conceptualization, N.L. and E.B.; methodology, E.B. and N.L.; software, E.B. and N.L.; validation, E.B., B.M., and N.L.; formal analysis, E.B., N.L., and B.M.; investigation, E.B., N.L., and B.M.; resources, N.L.; data curation, E.B. and N.L.; writing—original draft preparation, E.B., N.L., and B.M.; writing—review and editing, N.L., B.M., and E.B.; visualization, E.B., B.M., and N.L.; supervision, N.L.; project administration, N.L.; funding acquisition, N.L..

All authors have read and agreed to the published version of the manuscript.

List of Symbols

Symbol	Description	Page
$A(G)$	arc set of graph G	33
B	incidence matrix of graph G	35
\mathcal{B}	integral cycle basis of graph G	37
$c(G)$	number of connected components of graph G	37
C	hyperrectangle in $\mathbb{R}^{A(G)}$	45
d	scaling vector in $\mathbb{R}^{A(G)}$	63
$E(D, \kappa)$	equality graph of the weighted digraph (D, κ)	43
$\mathcal{F}_{L,U}$	face in the polytope X_{LP} corresponding to the structure (S, L, U)	68
G	directed graph of a PESP instance	33
\overline{G}	two-way graph of directed graph G	46
$\overline{G}_{L,U}$	subgraph of two-way graph \overline{G} induced by arc sets L and U	74
K	cell of a zonotopal tiling	69
ℓ	lower bounds on the arc set $A(G)$ of a PESP instance	33
m_p	embedding of $R(p)$ in X	46
\mathbf{m}_z	embedding of $\mathfrak{R}(z)$ in X	49
N	neighbourhood graph of the whole collection of polytropes in \mathcal{T}	58
N_p, N_z	neighbourhood of $\mathbf{R}(p)$, $\mathfrak{R}(z)$, respectively	57
PESP	Periodic Event Scheduling Problem	33
P_i	unbounded polyhedron, Minkowski sum of C and $\text{pos}(B_i^\top)$	72
p	periodic offset	35
$\text{pos}(\widehat{B^\top}_i)$	positive cone of $\widehat{B^\top}_i$	73
$r(p)$	intersection of the hyperrectangle C with the affine subspace S_p	45
$R(p)$	region for a periodic offset $p \in \mathbb{Z}^{A(G)}$ in timetable space Π	46
$\mathbf{R}(p)$	region in torus \mathcal{T} corresponding to polytrope $R(p)/\mathbb{R}\mathbf{1}$	48

$\mathfrak{R}(z)$	region in torus \mathcal{T} corresponding to polytrope $R(p)/\mathbb{R}\mathbf{1}$ with $\Gamma p = z$	49
S	spanning tree or subgraph of G	37, 62
$\mathcal{S}(G)$	set of spanning trees in G	62
(S, L, U)	structure as a triplet of set of arcs in G	68
S_p	affine subspace given by $\text{im } B^\top + Tp$ for $p \in \mathbb{Z}^{A(G)}$ and $T \in \mathbb{N}$	45
T	period time of a PESP instance	33
\mathbf{T}	all- T vector	35
\mathcal{T}	$(n-1)$ -dimensional torus, quotient of tropical projective space \mathbb{TP}^{n-1}	49
\mathbb{T}	tropical semiring	41
\mathbb{TP}^{n-1}	tropical projective space	42
u	upper bound on the arc set $A(G)$ of a PESP instance	33
$V(G)$	vertex set of graph G	33
$\text{vol}(Z)$	volume of polyhedron Z	64
w	weights on the arc set $A(G)$ for the objective function of PESP	34
$W(D, \kappa)$	weighted digraph polyhedron of graph D with arc-weights κ	41
$W_{\mathcal{B}}$	width of cycle basis \mathcal{B}	78
X	periodic tension polytope	36, 39
X_{LP}	periodic tension polytope of the natural LP-relaxation of PESP	51
x	periodic tension	33
Z	cycle offset zonotope, translate of the scaled cographic zonotope	40, 63
z	cycle offset	39
$Z(A, b)$	zonotope generated by matrix A and translation vector b	60
$Z(A)$	zonotope $Z(A, \mathbf{0})$	60
Γ	cycle matrix of an integral cycle basis \mathcal{B}	37
Γ_d	matrix obtained by scaling each column of Γ with d_a	63
γ	oriented cycle	36
γ_+, γ_-	positive and negative parts of γ , respectively	36
$\kappa(p)$	weight function for two-way graph \overline{G} and periodic offset p	55
μ	cyclomatic number, rank of the cycle space	36
Π	space of feasible periodic timetables	36
π	periodic timetable	33
\equiv_T	congruence modulo T	33
$\mathbf{0}, \mathbf{1}$	all-zeros, all-ones vector	35
\oplus, \odot	tropical addition and multiplication	41
A_S	submatrix of A by omitting columns indicated by arc-/index-set S	61
$\widehat{B^\top}_i$	matrix B^\top without the i -th column	73

CHAPTER 3

Tropical Neighbourhood Search: A New Heuristic for Periodic Timetabling

Bortoletto, E., Lindner, N., Masing, B.
22nd Symposium on Algorithmic Approaches for Transportation Modelling,
Optimization, and Systems (ATMOS 2022); 3:1-3:19; 106.
DOI: 10.4230/OASICS.ATMOS.2022.3.

This article is licensed under a Creative Commons Attribution 4.0 International License.
To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Abstract Periodic timetabling is a central aspect of both the long-term organization and the day-to-day operations of a public transportation system. The Periodic Event Scheduling Problem (PESP), the combinatorial optimization problem that forms the mathematical basis of periodic timetabling, is an extremely hard problem, for which optimal solutions are hardly ever found in practice. The most prominent solving strategies today are based on mixed-integer programming, and there is a concurrent PESP solver employing a wide range of heuristics [31]. We present tropical neighborhood search (**tns**), a novel PESP heuristic. The method is based on the relations between periodic timetabling and tropical geometry [1]. We implement **tns** into the concurrent solver, and test it on instances of the benchmarking library PESPLib. The inclusion of **tns** turns out to be quite beneficial to the solver: **tns** is able to escape local optima for the modulo network simplex algorithm, and the overall share of improvement coming from **tns** is substantial compared to the other methods available in the solver. Finally, we provide better primal bounds for five PESPLib instances.

Contents

3.1	Introduction	87
3.2	Tropical Decomposition of Periodic Timetable Space	88
3.3	Tropical Neighbourhood Search	94
3.4	Implementation details	96
3.4.1	Preparing the <i>exploreList</i>	96
3.4.2	Sorting the <i>exploreList</i>	97
3.4.3	The <i>qualityFactor</i>	98
3.4.4	Subproblem Formulation: Arc Offsets vs. Cycle Offsets	98
3.4.5	Hashing Visited Polytropes	99
3.5	Computational Experiments and Results	99
3.5.1	Impact of Parameter Choices for <code>tns</code>	100
3.5.2	Contribution of <code>tns</code> in Comparison to Other Methods	102
3.5.3	New <code>PESPlib</code> Incumbents	104
3.6	Outlook	105
3.A	Appendix	106

3.1 Introduction

Rhythm is to music what the timetable is for a public transit system. Periodicity of transportation networks is a common characteristic, quite useful in practice, and so periodic timetables are of particular importance. Setting up departure and arrival times in a feasible way is quite complicated, and the standard framework to model and optimize such timetables is that of the *Periodic Event Scheduling Problem* (PESP), first devised by Serafini and Ukovich [5]. Other than public transportation, PESP is also useful in automated production systems [66], and more generally in any case where periodicity constraints are in effect. Deciding whether a PESP instance is feasible is known to be NP-hard for any fixed period time $T \geq 3$ [24, 28], or when the underlying graph is series-parallel [25].

Other than a basic MIP formulation, in practice there have been many attempts to tackle the problem by a plethora of techniques [30, 36–41, 63, 65, 67]. However, the most successful method in practice remains the concurrent solver of Borndörfer, Lindner, and Roth [31], which, in parallel, implements MIP-based branch-and-cut [42], the modulo network simplex algorithm (*mns*, [36, 37]) as a local improvement heuristic, and a maximum-cut based heuristic [38], together with other features.

In this paper we introduce a novel heuristic, called *Tropical Neighbourhood Search* (*tns*). The *tns* algorithm is based on the link between the space of feasible periodic timetables and tropical geometry established in [1]. We will recall useful theoretical results in Section 3.2, which provide geometrical insight to the algorithm then described in Section 3.3. Finally, we will describe our implementation of *tns* in Section 3.4 and evaluate it in Section 3.5 on a subset of the instances of the benchmarking library *PESPLib* [32]. We conclude the paper with an outlook in Section 3.6.

3.2 Tropical Decomposition of Periodic Timetable Space

The goal of periodic timetabling in public transport is to assign timestamps to departure and arrival events of, e.g., trains at stations, so that the time between such events is within some given bounds. By the periodic nature, it suffices to consider timestamps modulo a period time T . The standard mathematical model for periodic timetabling is the *Periodic Event Scheduling Problem (PESP)* [5]. A PESP instance is comprised of a tuple (G, T, ℓ, u, w) , whose elements are:

- An *event-activity network* G , a directed graph whose vertices $V(G)$ represent *events* in the network, and whose arcs $A(G)$ represent *activities* between events. In the context of periodic timetabling, these events describe the points in time of departures or arrivals, while the activities model the time durations of driving between stations, dwelling at a station, transferring between lines, turning at terminal stations, or fixing headways [18]. We will assume that G is simple and weakly connected, which is no restriction [42].
- A *period time* $T \in \mathbb{N}$, indicating after what time an event should occur again.
- Vectors $\ell, u \in \mathbb{R}^{A(G)}$ of *lower* and *upper bounds* on the activities, such that $0 \leq \ell_a < T$ and $0 \leq u_a - \ell_a < T$, indicating minimum and maximum durations of $a \in A(G)$.
- A vector $w \in \mathbb{R}^{A(G)}$ of *weights*, often modelling an ascribed importance to a given activity, for example represented by the number of passengers partaking in said activity.

The variables to determine are the *periodic timetable*, a vector $\pi \in \mathbb{R}^{V(G)}$, and the *periodic tension*, a vector $x \in \mathbb{R}^{A(G)}$. A pair (π, x) of timetable and tension is said to be *feasible* if

$$\forall (i, j) \in A(G) : \quad \pi_j - \pi_i \equiv x_{ij} \pmod{T} \quad \text{and} \quad \ell_{ij} \leq x_{ij} \leq u_{ij}, \quad (3.1)$$

where the first constraint models the periodicity property, while the second ensures that the tension is within the given bounds. Note that due to $0 \leq u_a - \ell_a \leq T$ for all $a \in A(G)$, for any given π there is at most one x such that (π, x) is feasible. In this case, we will hence speak of *the* tension associated to a timetable.

Given an appropriate tuple (G, T, ℓ, u, w) , PESP consists in finding a feasible pair (π, x) such that the weighted tension $w^\top x$ is minimized. If ℓ and u are integral, which is true for most practical purposes, by a result of [24] the feasibility of the instance implies the existence of an integral optimal solution.

PESP can be formulated as a mixed-integer program by employing some auxiliary integer variables $p \in \mathbb{Z}^{A(G)}$ to model the modulo constraints by $\pi_j - \pi_i + Tp_{ij} = x_{ij}$ for all $(i, j) \in A(G)$. Then, using the incidence matrix $B \in \{-1, 0, 1\}^{V(G) \times A(G)}$ of G , the problem is as follows:

$$\min \quad w^\top x \tag{3.2a}$$

$$\text{s.t.} \quad -B^\top \pi + Tp = x, \tag{3.2b}$$

$$\ell \leq x \leq u, \tag{3.2c}$$

$$p \in \mathbb{Z}^{A(G)}, \pi \in \mathbb{R}^{V(G)}, x \in \mathbb{R}^{A(G)}, \tag{3.2d}$$

where each p_{ij} is called the (*periodic*) *offset* of the arc (i, j) . If (π, x) is a feasible timetable-tension-pair, it is straightforward to compute the unique corresponding vector of offsets.

Each of the three variables in the problem are of interest in themselves. The space of periodic tensions x has been analysed in-depth, in particular the convex hull X of all feasible tensions, see [24, 28–30]. Also the space of periodic offsets p has received attention, or better that of periodic cycle offsets z , which are analogous to periodic offsets and arise in an alternative MIP formulation of PESP,

namely the cycle formulation

$$\min w^\top x \quad (3.3a)$$

$$\text{s.t. } \Gamma x = Tz, \quad (3.3b)$$

$$\ell \leq x \leq u, \quad (3.3c)$$

$$z \in \mathbb{Z}^{\mathcal{B}}, x \in \mathbb{R}^{A(G)}, \quad (3.3d)$$

where \mathcal{B} is some integral cycle basis with cycle matrix $\Gamma \in \{-1, 0, 1\}^{\mathcal{B} \times A(G)}$, and z is an integer vector of so-called *periodic cycle offsets*, see, e.g., [42] for further details.

In [1] the polytope of feasible fractional cycle offset variables is recognised to be a zonotope, and several properties of PESP are derived via tilings of said zonotope.

What is instead of main interest in this paper is the space of periodic timetables.

Definition 3.1. — *For an instance (G, T, ℓ, u, w) , the set Π of feasible periodic timetables can be written as*

$$\Pi := \left\{ \pi \in \mathbb{R}^{V(G)} \mid \exists p \in \mathbb{Z}^{A(G)}, \forall (i, j) \in A(G): \ell_{ij} \leq \pi_j - \pi_i + Tp_{ij} \leq u_{ij} \right\}. \quad (3.4)$$

In particular, by defining for each $p \in \mathbb{Z}^{A(G)}$ the polyhedron

$$R(p) := \left\{ \pi \in \mathbb{R}^{V(G)} \mid \forall (i, j) \in A(G): \ell_{ij} - Tp_{ij} \leq \pi_j - \pi_i \leq u_{ij} - Tp_{ij} \right\}, \quad (3.5)$$

the feasible timetable space can be expressed as the union

$$\Pi = \bigcup_{p \in \mathbb{Z}^{A(G)}} R(p). \quad (3.6)$$

As introduced in [1], each $R(p)$ is a *weighted digraph polyhedron* [44]. Namely, for any fixed $p \in \mathbb{Z}^{A(G)}$ it can be described as

$$R(p) = \left\{ \pi \in \mathbb{R}^{V(\overline{G})} \mid \forall (i, j) \in A(\overline{G}): \pi_j - \pi_i \leq \kappa(p)_{ij} \right\}, \quad (3.7)$$

for the weighted digraph $(\overline{G}, \kappa(p))$, with the following:

- vertices $V(\overline{G}) := V(G)$,
- arcs $A(\overline{G}) := A(G) \cup A(G^\top)$, where $A(G^\top) = \{(j, i) \mid (i, j) \in A(G)\}$,

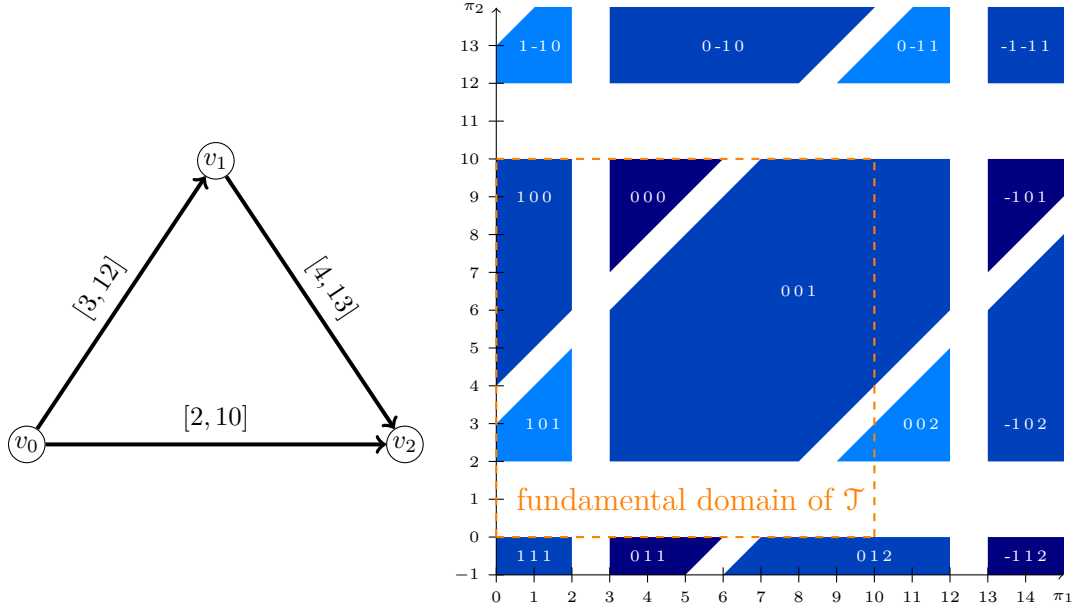


Figure 3.1: A PESP instance and its polytropical decomposition $\Pi/\mathbb{R}\mathbf{1}$ ($T = 10$, w arbitrary).

- weights $\kappa(p)_{ij} := u_{ij} - Tp_{ij}$ for all $(i, j) \in A(G)$, and $\kappa(p)_{ij} := Tp_{ji} - \ell_{ji}$ for all $(i, j) \in A(G^\top)$.

By construction, every $(\bar{G}, \kappa(p))$ is strongly connected, therefore the lineality space of its weighted digraph polyhedron is solely $\mathbb{R}\mathbf{1}$ [44].

Let $\mathbb{T} := \mathbb{R} \cup \{\infty\}$ be the *tropical semiring*, with the tropical sum $a \oplus b := \min\{a, b\}$ and the tropical product $a \odot b := a + b$, see, e.g., [50] for more background. A set $S \subset \mathbb{T}^n$ is *tropically convex* if $(a \odot x) \oplus (b \odot y) \in S$ for every $x, y \in S$, and any $a, b \in \mathbb{T}$. It was shown in [44] that weighted digraph polyhedra arise as the *tropical convex hull* of finitely many points with coordinates in \mathbb{T} . Moreover, when the underlying digraph is strongly connected, no ∞ -coordinates appear. In this case, which is the one interesting to us, all weighted digraph polyhedra can be seen as *polytropes* [43] by quotienting out the trivial lineality space, i.e., $\mathbb{R}\mathbf{1}$. We refer to [1] to see how general properties of polytropes translate to the context of periodic timetabling, e.g., the relation between polytrope vertices, vertices of the tension polytope X , and spanning tree structures. See also [26].

It is clear that $R(p) \cap R(q) = \emptyset$ holds for any two distinct offset vectors p and q , since $u - \ell < T$ by hypothesis. If $\Pi \neq \emptyset$, then the set $\Pi/\mathbb{R}\mathbf{1}$ is therefore a disjoint

union of infinitely many polytropes, as we have visualized for a small exemplary instance in Figure 3.1. However numerous, these polytropes adhere to a certain structure, which we summarise in the following proposition.

Proposition 3.1 ([1], §3.3). — *Consider the PESP instance (G, T, ℓ, u, w) with timetable space Π , denoting as B the incidence matrix of G , and as \mathcal{B} an integral cycle basis of G , with cycle matrix Γ . Then:*

- (a) *For any feasible timetable $\pi \in \Pi$ all of its translations by integer multiples of T are feasible: If $\pi \in R(p)/\mathbb{R}\mathbf{1}$, then $\pi + Tq \in R(p + B^\top q)/\mathbb{R}\mathbf{1}$ for all $q \in \mathbb{Z}^{V(G)}$.*
- (b) *Two feasible timetables $\pi, \pi' \in \Pi$ have the same associated periodic tension if and only if there exists $q \in \mathbb{Z}^{V(G)}$ such that $\pi' = \pi + Tq$.*
- (c) *Two feasible timetables $\pi, \pi' \in \Pi$ have the same associated periodic tension if and only if $\Gamma p = \Gamma p'$ for the associated offsets $p, p' \in \mathbb{Z}^{A(G)}$.*

The unbounded set $\Pi/\mathbb{R}\mathbf{1}$ then turns out to be simpler than expected, since its ambient space can be restricted by another quotient, based on the equivalence relation

$$p \cong p' \iff \Gamma p = \Gamma p' \tag{3.8}$$

implied in the above proposition. In view of the cycle formulation (3.3) of PESP, we consider p and p' equivalent whenever they correspond to the same cycle offset. With $n := |V(G)|$, we define the *torus of feasible periodic timetables* $\mathcal{T} := (\mathbb{R}^n / (T\mathbb{Z})^n) / \mathbb{R}\mathbf{1}$. This is an $(n - 1)$ -dimensional torus of side length T , whose representative can be any full-dimensional hypercube of side length T in $\mathbb{R}^n / \mathbb{R}\mathbf{1}$, called *fundamental domain*. It now makes sense to also have a shorthand notion to refer to the quotient of our weighted digraph polyhedra in \mathcal{T} . We choose $(R(p)/(T\mathbb{Z})^n)/\mathbb{R}\mathbf{1} =: \mathbf{R}(p) \subseteq \mathcal{T}$.

To conclude this recapitulatory section, it is now possible to describe how the polytropes position themselves inside some fundamental domain, along the lines of [1]. Given a PESP instance (G, T, ℓ, u, w) , we define (in breach of our hypothesis

of $u - \ell < T$) its *limit instance*, where all upper bounds u_a are substituted with $\ell_a + T$, and denote it by $(G, T, \ell, w)_\infty$. For a polytrope $\mathbf{R}(p)$ of the base instance we denote as $\mathbf{R}'(p)$ the polytrope in the limit instance that contains it. We now say that two non-empty polytropes $\mathbf{R}(p)$ and $\mathbf{R}(q)$ are *neighbours* when $\mathbf{R}'(p)$ and $\mathbf{R}'(q)$ intersect in a common facet.

Proposition 3.2 ([1], §3.7). — *Let $p \in \mathbb{Z}^{A(G)}$ be an offset vector with $\mathbf{R}(p) \neq \emptyset$, k an integer, and $e_{ij} \in \mathbb{Z}^{A(G)}$ the canonical basis vector of the arc $(i, j) \in A(G)$. Then:*

- (a) *If $|k| > 2$, then $\mathbf{R}(p + ke_{ij})$ is empty.*
- (b) *If $|k| > 1$, then $\mathbf{R}(p)$ and $\mathbf{R}(p + ke_{ij})$ are not neighbours.*
- (c) *If $|k| = 1$ and $\mathbf{R}(p + ke_{ij}) \neq \emptyset$, then $\mathbf{R}(p)$ and $\mathbf{R}(p + ke_{ij})$ are neighbours, and one of the two inequalities defined by the arc (i, j) is facet-defining for $\mathbf{R}(p)$: For $k = 1$ this is the lower bound inequality $\pi_j - \pi_i \geq \ell_{ij} - Tp_{ij}$, for $k = -1$ this is the upper bound inequality $\pi_j - \pi_i \leq u_{ij} - Tp_{ij}$.*
- (d) *Two non-empty polytropes $\mathbf{R}(p)$ and $\mathbf{R}(q)$ are neighbours whenever there exist representatives of the equivalence classes of p and q whose difference is, up to sign, a canonical basis vector. In other words, whenever there exists an arc $(i, j) \in A(G)$ such that $[p]_{\cong} - [q]_{\cong} = [\pm e_{ij}]_{\cong}$.*

This allows the construction of the *neighbourhood graph* of an instance, whose nodes are the equivalence classes of offsets, and two classes are adjacent if their respective polytropes are neighbours in \mathcal{T} . For the limit instance of the instance of Figure 3.1, the polytropical decomposition is depicted in Figure 3.2 next to the neighbourhood graph derived from it: Each dark blue triangle corresponds to the equivalence class $[0, 0, 0]_{\cong}$ and shares a facet only with the hexagon, corresponding to the class $[0, 0, 1]_{\cong}$. This, in turn, has both $[0, 0, 0]_{\cong}$ and $[0, 0, 2]_{\cong}$ as neighbours.

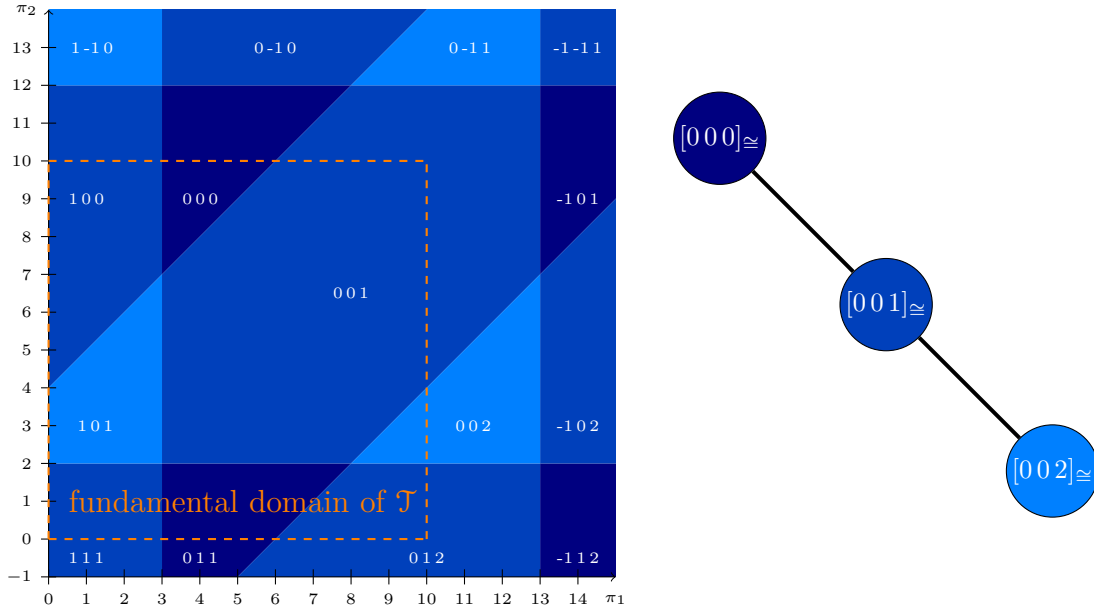


Figure 3.2: Limit instance and neighbourhood graph for the same instance as in Figure 3.1.

3.3 Tropical Neighbourhood Search

We can now outline the core steps undertaken by the promised heuristic, which we call *Tropical Neighbourhood Search* (**tns**). It is a local improvement heuristic that operates within the framework of the concurrent solver [31]. The solver keeps a *pool* of the feasible solutions it finds, ordered by objective value, and various local improvement heuristics use the pooled solutions as starting points. In particular, **tns**, once given a starting solution (π^*, x^*, p^*) , identifies the polytrope $\mathbf{R}(p^*)$ and proceeds to explore some neighbouring polytropes, i.e., it determines the optimal weighted tension over each $\mathbf{R}(p)$ for (a subset of) neighbours p of p^* . If an improving solution is found, it is added to the pool. Doing so, it in fact operates on the neighbourhood graph of the given instance.

Formally, our heuristic can be described by Algorithm 1, where

- $\text{PESP}|_p$ is simply PESP restricted to the specific offset vector p , i.e., we solve (3.2) with all integer variables fixed to p . This is a linear program, which is dual to an uncapacitated minimum cost network flow problem [36].

Algorithm 1 Tropical Neighbourhood Search **tns**

Require: PESP instance (G, T, ℓ, u, w)

```

1:  $(\pi^*, x^*, p^*) \leftarrow$  pick a starting solution from the pool
2:  $x_{\text{tns}} \leftarrow x^*$ 
3: for arc  $(i, j)$  and direction  $k \in \{-1, +1\}$  in exploreList do
4:   fix offset  $p \leftarrow p^* + ke_{ij}$ 
5:   solve  $\text{PESP}|_p$ 
6:   if  $\text{PESP}|_p$  feasible then
7:      $(\pi_{\text{opt}}, x_{\text{opt}}, p_{\text{opt}}) \leftarrow$  optimal solution of  $\text{PESP}|_p$ 
8:      $\text{improv} \leftarrow (w^\top x_{\text{tns}} - w^\top x_{\text{opt}})/(w^\top x_{\text{tns}})$ 
9:     if  $\text{improv} > 0$  then
10:      Add  $(\pi_{\text{opt}}, x_{\text{opt}}, p_{\text{opt}})$  to the pool
11:      if  $\text{improv} > \text{qualityFactor}$  then
12:        break
13:       $x_{\text{tns}} \leftarrow x_{\text{opt}}$ 

```

- *exploreList* is a list of arc-direction tuples, indicating which neighbours to explore. It may contain only a subset of all possibilities.
- The solution picking method could vary in principle, although in our implementation it always selects the solution in the pool with smallest weighted tension.
- *qualityFactor* $\in [0, 1]$ is a factor utilized as a preemptive exit condition, which triggers when the percentage improvement of a newly found solution exceeds this factor.

Note that Algorithm 1 is a description of **tns** with the incidence matrix formulation of PESP (3.2). One can equivalently work with the cycle formulation (3.3) instead, which changes the LP subproblem of PESP to $\text{PESP}|_z$ for a vector z of periodic cycle offsets. This poses no issue, and we refer to the next section for more details.

It is known that **tns** can be used to escape local minima reached via the modulo network simplex, as there even exist such instances where the neighbourhood graph has none [1].

3.4 Implementation details

As anticipated, there are various elements in Algorithm 1 that may alter the overall behaviour and performance of the algorithm depending on how they are adjusted. Therefore, before moving forward with the computational experiments, we will now detail the characteristics of such elements, what settings and strategies we decided to employ, the motivations that moved us, and other minor implementation details.

3.4.1 Preparing the *exploreList*

A deciding factor for both the speed and the behaviour of our `tns` heuristic is determining the search space. Clearly, choosing which or how many neighbouring polytopes to explore is a factor that deserves consideration, but even the order of exploration may affect the overall performance, since it has potentially positive interplay with concurrency.

As we know from Proposition 3.2, only arcs (i, j) whose inequalities are facet-defining for $\mathbf{R}(p^*)$ can yield feasible neighbours, and only in the appropriate direction. Unfortunately, knowing a priori which inequalities are facet-defining is not trivial, and we therefore detail two strategies: one precise but slow, one fast but imprecise. When setting up *exploreList* in our tests, we decided to either scan all possible neighbours, i.e., all pairs $(a, +1)$ and $(a, -1)$ for all arcs $a \in A(G)$, or to restrict ourselves to a subset of the facet-defining inequalities, namely those that are tight in the starting solution (π^*, x^*, p^*) , i.e., pairs $(a, +1)$ if $x_a^* = \ell_a$ and pairs $(a, -1)$ if $x_a^* = u_a$. This second way only the faces on a particular side of the polytope are considered. We mark the first *exploreList* strategy by `all`, and the second one by `side`. The `side` strategy is quick to set up, but has the defect of not considering all facet-defining inequalities. Note that when a simplex-based LP solver is invoked on $\text{PESP}|_p$ or $\text{PESP}|_z$, then (π^*, x^*, p^*) will be a vertex of $\mathbf{R}(p^*)$.

Given the equivalence relation \cong (3.8), we know that polytopes are uniquely determined by their cycle offset $z = \Gamma p$. Given then neighbouring periodic offsets $p + e_{i_1 j_1}$ and $p + e_{i_2 j_2}$, for arcs (i_1, j_1) and (i_2, j_2) in $A(G)$, it may happen that

$\Gamma(p + e_{i_1j_1}) = \Gamma(p + e_{i_2j_2})$ and the two explorations end up being identical. Therefore, another way of avoiding irrelevant explorations is to fix any cycle matrix Γ of the instance graph and pre-process all arcs, storing a unique representative for all arcs whose columns in Γ are identical.

3.4.2 Sorting the *exploreList*

Another choice to be made while preparing *exploreList* is the order in which to consider the arc-direction pairs. This can be influential because if a good solution is put into the pool earlier, then it is earlier available to other methods in the concurrent solver. In particular, in combination with the quality factor, this can lead to **tns**-loops that are shorter but still improvement-dense. We decided to use four different strategies to sort the arcs:

- s1 descending weight w_a , to prioritize exploration of heavy and hence influential arcs;
- s2 descending span $u_a - \ell_a$, to prioritize exploration of neighbours that are close-by, and therefore more likely feasible, since two neighbouring polytropes $\mathbf{R}(p)$ and $\mathbf{R}(p \pm e_{ij})$ have distance at most $T - (u_{ij} - \ell_{ij})$;
- s3 descending weighted span $w_a(u_a - \ell_a)$, to combine the two sorting strategies above;
- s4 descending average improvement, so as to prioritize exploration via arcs that on average have given good improvements in previous iterations. While the previous three strategies are pre-processed at the beginning, this is a dynamic sorting strategy, which keeps track of the average (positive and negative) improvements given by each arc throughout the various iterations. The rationale behind this is to prioritize all those arcs which provided net improvements but not the best improvement overall. Initially all averages are set to 0, and no changes is made in case of infeasibility. This is similar to pseudocost branching in mixed-integer programming [68].

3.4.3 The *qualityFactor*

The quality factor can be interpreted as a percentage, based on which the `tns`-loop is terminated early in case of a percentage-improvement that exceeds the given bound. As limit cases this means that any positive improvement whatsoever is enough to conclude the search when the factor is set to 0%, and that no quality-based exit can happen when the factor is set to 100% or more. In our tests, we perform our tests using two quality factors:

- `q0.001`, meaning 0.1% quality factor.
- `q1`, meaning 100% quality factor: every arc-direction tuple of `exploreList` is considered.

3.4.4 Subproblem Formulation: Arc Offsets vs. Cycle Offsets

As mentioned, Algorithm 1 is `tns` with respect to the incidence formulation of PESP (3.2), but one can equivalently perform `tns` using the cycle formulation (3.3) instead. The algorithm then reads the same as Algorithm 1, except that the cycle offsets are computed and used instead, with line 4 changing to $z \leftarrow z^* + k\Gamma e_{ij}$, and line 5 now solving $\text{PESP}|_z$. Notice that Γe_{ij} is indeed the (i, j) -th column of the cycle matrix. In this, the choice of which cycle basis to use can be quite influential on the solving speed of the linear programs $\text{PESP}|_z$. Preliminary tests showed that for each instance there can be impressive differences, up to a factor 14, between the average solving times of different problem formulations and different cycle bases.

In order to choose which formulation to use for each instance, we compared the average `for`-loop iteration time of each of them and then simply picked the fastest one. The formulations tested were the incidence matrix formulation, and four variants of the cycle matrix formulation. One used a minimum width cycle basis [45], whereas three used different fundamental cycle bases: from a minimum span, minimum weight, and a minimum weighted span spanning tree, respectively. Since the average iteration time appeared very consistent throughout the tests and short even in the worst cases, tests of less than a minute per formulation are

more than enough to process hundreds of linear programs and thereby compute an applicable average iteration time. In particular, the cycle formulation performed well overall, with the fundamental cycle bases of a minimum weighted span spanning tree being the fastest in all but two instances, where the fundamental cycle bases of a minimum span spanning tree were best instead.

Regardless of the specific cycle bases used in our tests, these evaluations were fast to obtain, and it can be suggested that a similar pre-evaluation strategy could be systematically used in the future.

We use Gurobi 9.5 [69] to solve each iteration’s linear program.

3.4.5 Hashing Visited Polytopes

Throughout repeated use of `tns`, in particular in the exploration of different neighbourhoods, it is possible to explore the same polytrope multiple times, since the neighbourhoods of any two polytopes may have non-trivial intersection. A way to prevent this from happening is then to progressively keep a record of every processed offset vector and skip it whenever it is encountered again. In our preliminary performance evaluations this tracking method seemed to have little effect, positive or negative. We therefore decided to maintain it, hoping for a stronger impact in longer tests.

3.5 Computational Experiments and Results

We conducted several tests on eight `PESPlib` instances [32] of varying size, namely `R1L1`, `R2L2`, `R3L3`, `R4L4`, `R1L1v`, `R4L4v`, `BL1`, `BL3`. The last two are bus timetabling instances, whereas the rest are based on railway networks. For each we used both `warm` starts, employing initial solutions close to the `PESPlib` current best primal bounds (cf. Table 3.1), and `cold` starts without any initial solution.

Overall the various `tns` parameters are summarised in Table 3.2. Each combination was tested within the concurrent solver [31]. Going forward we will refer to `mns` tests when the modulo network simplex method works alone, `tns+mns` tests

R1L1	31 099 786	R2L2	43 404 232	R3L3	44 837 461	R4L4	38 836 756
R1L1v	43 258 386	R4L4v	64 408 523	BL1	8 457 513	BL3	8 502 382

Table 3.1: Initial solution values for the warm starts.

Instances	<i>exploreList</i>	sorting	<i>qualityFactor</i>	Initial Solution
R1L1, R2L2, R3L3, R4L4, R1L1v, R4L4v, BL1 , BL3	all side	s1 s2 s3 s4	q0.001 q1	cold warm

Table 3.2: Parameter combinations for `tns`.

when the two heuristics run concurrently, and `complete` tests when `tns` and all the methods implemented in the concurrent solver are used together.

3.5.1 Impact of Parameter Choices for `tns`

In a first step, we want to evaluate how much the choice of arc-direction tuples and their sorting influence our results. We run `tns+mns` and compare it to `mns` alone, for all parameter combinations, see Table 3.2. For a meaningful comparison of the test runs, we disabled multi-node cuts within the `mns` implementation, because of their randomizing character. To complete the analysis we also run `complete` tests. The computation time per configuration is one hour wall time each, performed on an Intel i7-9700K CPU with 64 GB RAM.

3.5.1.1 `mns+tns` vs. `mns`

We can make the following observations: In combination with `mns`, our new heuristic was able to beat `mns` alone for all instances, as becomes evident from Table 3.6. Highlighted entries correspond to the best objective in comparison to the other parameter choices per instance. The last column, corresponding to the objective value obtained by `mns` alone is never in the first place, while any other column is the winner at least once.

We point out that for one instance, namely the warm-started R4L4, `mns` was not able to find any improvement, while all four sortings of `all` arcs with low

qualityFactor found the same improvement. This supports our claim that **tns** can be used to escape local minima.

To assess each heuristic’s performance, we rank them by their objective value after 6 minutes (i.e., 10% of the total running time) and after 1 hour (100%), such that the best objective is ranked in first place, and assign the same placement number for equal objectives. When comparing the average ranking values, it is hard to discern a clear ranking in between the **tns** parameter choices: **all** with pseudocost-like improvement (**s4**) and *qualityFactor* **q1** seems best on average, but is a clear winner only for the **cold R3L3** instance.

The two exemplary plots for the two instances **R1L1v** and **BL3**, see Figure 3.3, show the development of the objective for **mns** vs. **tns+mns**. It is evident that each of the parameter choices is reasonable, and depending on the instance may perform well or not. For example, **side-s1-q1** is the best for **BL3-cold**, yet one of the worst heuristics in the **R1L1v-cold** run.

Another property which can be seen in the figures is that in the beginning, the **side** instances tend to perform better. After a while however, the **all**-runs become competitive.

3.5.1.2 complete Runs

This phenomenon is even stronger when evaluating the parameter choices in the **complete** runs: When comparing the average ranking of the methods after 6 minutes with the final state after an hour, as indicated in Table 3.7, one can observe that – with the exception of **all-s3-q0.001** – the **all**-heuristics rank better, while **side** methods worsen.

This behaviour can be also observed when looking at the graphs for the **complete** case in Figure 3.4. What catches the eye in these figures is that the **all** runs seem to have the same shape in objective development as the **side** runs, but lag behind. After a while however, the dark (**all**) strands catch up to the light (**side**) strands. A similar pattern can be observed in most of the instances, particularly for the larger ones. An explanation for this could be that in the beginning, improvements

	warm			cold		
	avg.	min	max	avg.	min	max
BL1	7.43	0.0	17.97	3.68	0.02	8.52
BL3	4.68	0.0	14.59	2.92	0.0	6.04
R1L1	7.66	6.38	10.04	2.95	0.04	5.15
R1L1v	0.81	0.0	5.02	2.96	0.03	5.92
R2L2	6.89	0.0	31.64	2.76	0.05	5.46
R3L3	15.05	6.98	21.71	2.95	1.05	6.21
R4L4	9.52	1.16	12.45	1.89	0.89	3.23
R4L4v	0.56	0.0	1.5	1.48	0.0	3.76

Table 3.3: `tns`' contribution to the improvement gained in the `complete` runs in %.

are easily found, and `side` will quickly update the pool and restart with a better solution, while `all` will continue to iterate through all options, even though better solutions have already been obtained (possibly) by other concurrent methods. In contrast, in the later stages, when improving solutions are hard to find, it pays off to search through all of the neighbouring polytropes. In contrast to `mns+tns`, in `complete` a low quality factor produces better results on average. This can be explained in a similar way as above: A low quality factor disrupts unnecessary explorations when larger improvements are found in the beginning. With time, the improvements in objective become smaller, such that `qualityFactor` has less of an impact, so that most of the arc-direction pairs in `exploreList` are explored anyway.

We conclude that all sorting factors are relevant, as each one performs well for some instances. Which one is the best choice is hard to predict in advance, and overall – particularly in the interplay with other concurrent heuristics – their influence is not large. Both `side` and `all` are valid choices for `exploreList`: The former is better suited for earlier stages of solving, while the latter performs well once improvements become hard to find.

3.5.2 Contribution of `tns` in Comparison to Other Methods

Aside from the behaviour as discussed in the previous section, we want to analyze the quality of the contributions of `tns` in the scope of the concurrent solver. To this end, we compare the improvement of the objective value obtained by the different algorithms in the `complete` runs. What should be noted first, is that the total

	warm			cold		
	avg.	min.	max.	avg.	min.	max.
BL1	4.97	0.06	11.87	4.73	0.0	13.78
BL3	6.74	0.0	26.61	5.32	0.01	26.48
R1L1	0.0	0.0	0.0	8.28	0.0	41.78
R1L1v	9.46	0.0	72.64	7.41	1.15	28.49
R2L2	2.76	0.0	11.18	2.22	0.0	10.61
R3L3	1.79	0.47	4.02	7.15	0.0	37.09
R4L4	2.84	0.0	15.62	1.37	0.0	3.45
R4L4v	0.42	0.0	1.38	2.59	0.0	14.44

Table 3.4: `tns`' contribution to the improvement gained in the `complete` runs in % after 6 minutes.

improvements of `cold` starts are significantly larger than those of `warm` starts, and this also holds for `tns` in the `complete` runs. In relation to the total improvements however, the contribution of `tns` is larger for `warm` starts. We see evidence of that in Table 3.3: It shows the average, minimum and maximum improvement found by any of the `exploreList`'s choices in relation to the total improvement in the `complete` run in percent. With the exception of `R4L4v` and `R1L1v`, the average (and in most cases also the maximal) values of the `warm` started instances is larger than of the `cold` started ones. Table 3.4 displays the same, except that the first 6 minutes are excluded from the improvements. One can observe that compared to Table 3.3 the contribution of `tns` increases for the `cold` instances. This observation suggests that `tns` is particularly well suited for the later stages of solving a PESP instance, namely when improvements increment more slowly. At the beginning, when still far from a (local) minimum, `tns` is dominated by other algorithms in the solver.

Very noticeable in Table 3.3 and Table 3.4 is the wide range of `tns`' contribution for the different choices of `exploreList`. In almost all instances there is at least one choice which provides close to zero improvement, while the maximum value goes up to double-digit percentages. This property can also be observed in Figure 3.5. Here, we have chosen the exemplary instance `R3L3` and displayed the fractional contributions of all used heuristics in the `complete` solver. The `warm` started instance (left) has significantly more contribution through `tns` (green parts) in comparison to the `cold` started one. The top plots display the contributions for the whole time

frame, while the lower plots show them for the last 54 minutes. When comparing the upper to the lower plots, it becomes evident, that some of *exploreList*'s choices gain in importance, while others seem to perform particularly badly. E.g., for the `cold` run, each of the sortings with low *qualityFactor* seem to contribute similarly in the beginning, but after the first 6 minutes have passed, `s4` clearly contributes the most to the concurrent solution, yet the largest total improvement is found by `s1`, with only little direct `tns` contribution. Which one of the sortings provide the best solution is not clear however, our experiments did not show any clear indication. We therefore conclude that it may be worth it to try different sorting techniques in `tns` if no good improvements are found.

Based on Figure 3.5, we observe that the runs with high *qualityFactor* result in less improvement than with low *qualityFactor*, and the `tns` contribution is also often higher for low quality factors. While not the case for each instance and sorting, this seems to be a general tendency. When comparing `tns`' influence over time, this hierarchy is less prominent.

This supports again our interpretation of the previous section: Low quality factors are advantageous in the beginning. At a later stage, when the objective improvements become smaller, the *qualityFactor* exit condition is rarely triggered, regardless of low or large choice.

3.5.3 New PESPlib Incumbents

Based on the observation that `tns` contributes significantly to finding better solutions for PESP instances within the framework of the concurrent solver, we were able to compute new best primal solutions for 5 out of the 8 considered PESPlib instances. For some of these instances, we could find such a solution already within one hour in our `complete` experiments (see, e.g., BL3 in Table 3.7). We then let the solver run for another 8 hours to further improve the timetables. We summarize the objective values of these new incumbents in Table 3.5.

Instance	New Value	Old Value	Time (s)
BL3	6 675 098	6 999 313	25 732
R1L1v	42 591 141	42 667 746	9 110
R3L3	40 483 617	40 849 585	3 547
R4L4	36 703 391	36 728 402	11 122
R4L4v	61 968 380	64 327 217	3 625

Table 3.5: New incumbents for 5 PESPlib instances found with the help of `tns`. The old values are as of July 7, 2022. The last column shows the (wall) time of discovery.

3.6 Outlook

The `tns` algorithm turns out to be a valuable supplement to the already enormous zoo of periodic timetabling heuristics, being capable to provide timely and practical schedule improvements. For future research, it seems reasonable to embed `tns` in a metaheuristic such as tabu search or simulated annealing in order to overcome local optima. Another branch of research would be to employ automated algorithm configuration techniques [70] to find out which parameters work best for a given instance.

Authors' Contributions

Conceptualization, N.L. and E.B.; methodology, N.L. and E.B.; software, E.B., N.L., and B.M.; validation, E.B. and N.L.; formal analysis, E.B. and B.M.; investigation, E.B.; resources, N.L.; data curation, E.B. and N.L.; writing—original draft preparation, E.B., B.M.; writing—review and editing, N.L., B.M., and E.B.; visualization, E.B. and B.M.; supervision, N.L.; project administration, N.L.; funding acquisition, N.L..

All authors have read and agreed to the published version of the manuscript.

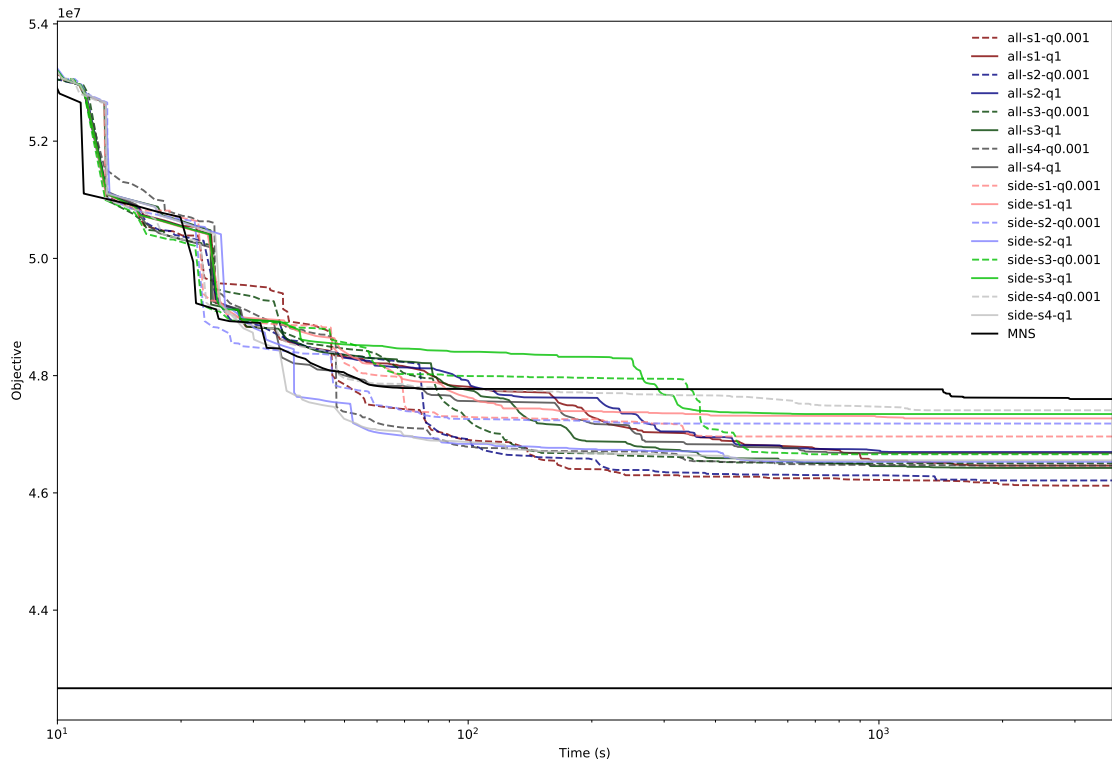
3.A Appendix

	all								
	s1-q0.001	s1-q1	s2-q0.001	s2-q1	s3-q0.001	s3-q1	s4-q0.001	s4-q1	
BL1-warm	8258561	8235826	8186321	8300527	8182495	8238180	8208581	8213667	
BL1-cold	8477888	8403606	8549631	8857331	8275775	8520448	8817740	8346471	
BL3-warm	8359354	8348109	8098761	8312052	8062572	8348109	8327236	8379410	
BL3-cold	8842185	9239538	9075754	8684087	8999692	8522210	9012239	8861540	
R1L1-warm	30678496	30610793	30600296	30588100	30678496	30578866	30600296	30583524	
R1L1-cold	35788003	35103477	35300490	35646174	35588509	35374751	35589367	35382965	
R1L1v-warm	42943355	42943355	42943355	42943355	42943355	42943355	42943355	42943355	
R1L1v-cold	46122798	46465421	46212022	46696787	46504815	46425825	46467141	46678146	
R2L2-warm	43398483	43382565	43382736	43382736	43398483	43382565	43398483	43382736	
R2L2-cold	45545963	46106414	44143731	45270818	45016237	45032259	44405920	44233438	
R3L3-warm	44546204	44544442	44591244	44539593	44544442	44544948	44548577	44593350	
R3L3-cold	44296073	43407015	42430742	42488680	43840438	42806733	42610123	42176416	
R4L4-warm	37387179	37460489	37405396	37492682	37312244	37367970	37366297	37363497	
R4L4-cold	42975571	41336513	42929915	42261165	42627952	41546954	42003636	42335060	
R4L4v-warm	64403669	64408523	64403669	64406909	64403669	64408523	64403669	64408523	
R4L4v-cold	65376186	66594246	66351066	66694524	65949084	66391164	66296248	65823105	
6 min. ranking	9.88	10.62	7.25	8.38	8.62	9.12	8.38	6.69	
final ranking	9.0	8.06	6.75	9.0	7.06	6.5	7.62	6.19	

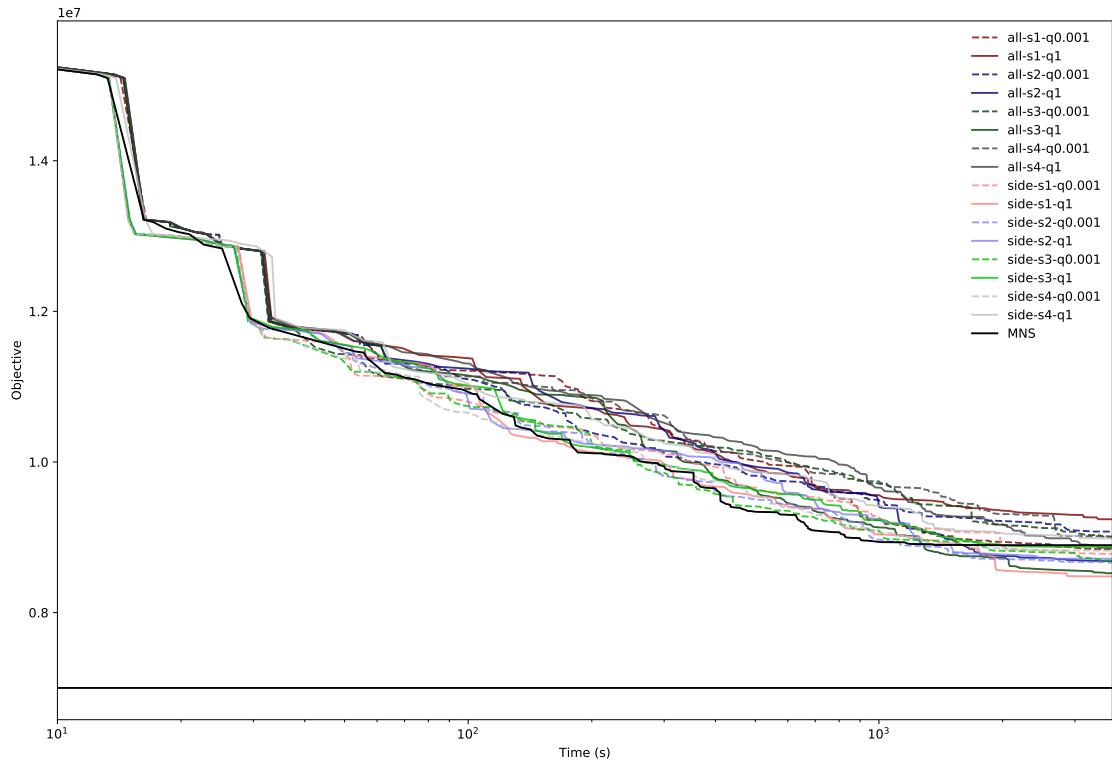
	side								MNS
	s1-q0.001	s1-q1	s2-q0.001	s2-q1	s3-q0.001	s3-q1	s4-q0.001	s4-q1	
BL1-warm	8156407	8217310	8273813	8197344	8204102	8214464	8145927	8177211	8362237
BL1-cold	8790388	8464797	8523165	8509967	8681972	8375229	8688826	8676485	8973473
BL3-warm	8376740	8263350	8408354	8065263	8408354	7946851	8193045	8029258	8359914
BL3-cold	8780061	8479738	8664483	8715578	8692029	8873040	8821544	8985128	8895307
R1L1-warm	30674972	30658531	30691866	30688021	30756833	30688021	30756833	30681160	30684785
R1L1-cold	36423144	35686177	36133582	35353728	36044751	36078420	36541854	35633823	36390414
R1L1v-warm	42943355	42943355	42943355	42943355	42943355	42943355	42943355	42943355	42946450
R1L1v-cold	46961984	47270557	47182681	46530813	46658767	47345018	47409082	46555105	47600910
R2L2-warm	43398483	43364985	43398483	43386980	43386980	43398483	43398483	43364985	43385954
R2L2-cold	44667116	44593903	44502873	44640728	44496851	44428158	44403440	44522515	44504010
R3L3-warm	44795451	44795451	44795451	44795451	44795451	44795451	44795451	44795451	44810246
R3L3-cold	42187095	43170308	44316260	43665920	43376728	43507592	43430305	43574669	45577898
R4L4-warm	37415040	37399063	37356432	37386139	37314559	37288889	37363831	37311361	37444171
R4L4-cold	42846346	42044976	41788252	41528536	41952148	41772120	42575956	42763956	42622532
R4L4v-warm	64408523	64408523	64408523	64408523	64408523	64408523	64408523	64408523	64408523
R4L4v-cold	66228809	65578506	65739359	65741111	66134278	66032110	65175140	65877024	66270894
6 min. ranking	8.12	6.19	7.75	6.88	8.62	6.56	7.56	7.88	6.81
final ranking	9.94	7.06	9.5	7.19	8.69	7.81	8.69	7.5	13.44

Table 3.6: Objective values of `tns` and `mns` in parallel after 1h wall time, in comparison to `mns` alone.

3. Tropical Neighbourhood Search: A New Heuristic for Periodic Timetabling 107

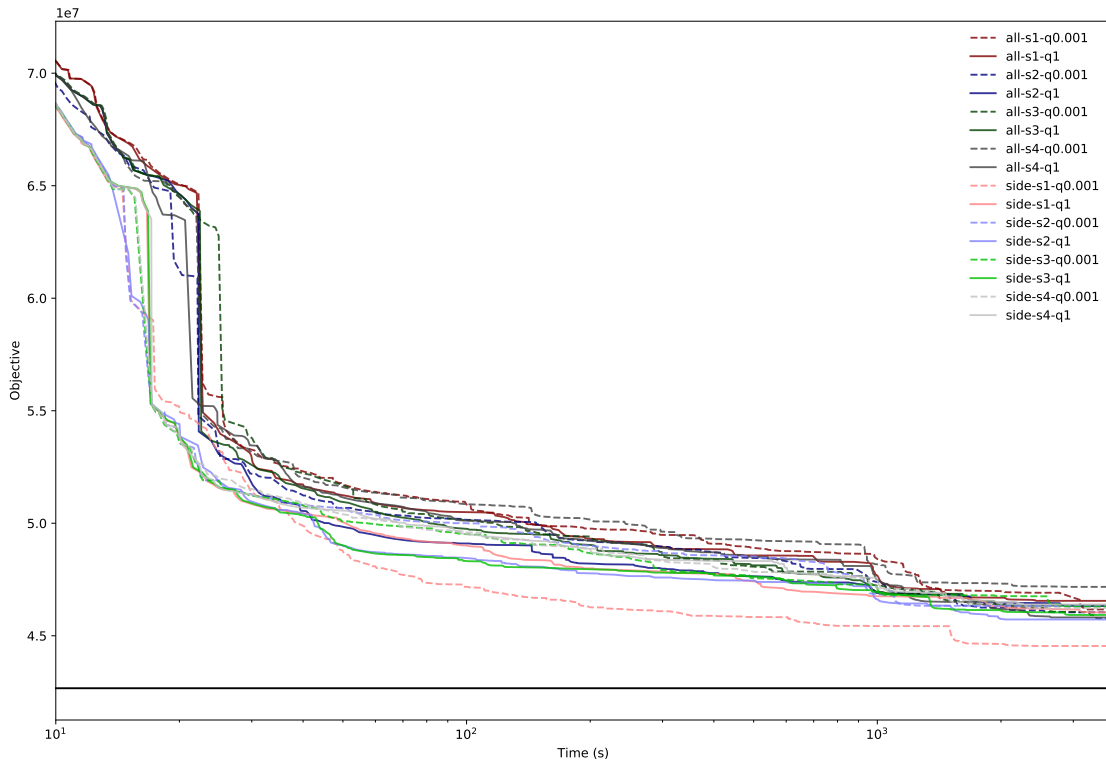


(a) R1L1v-cold

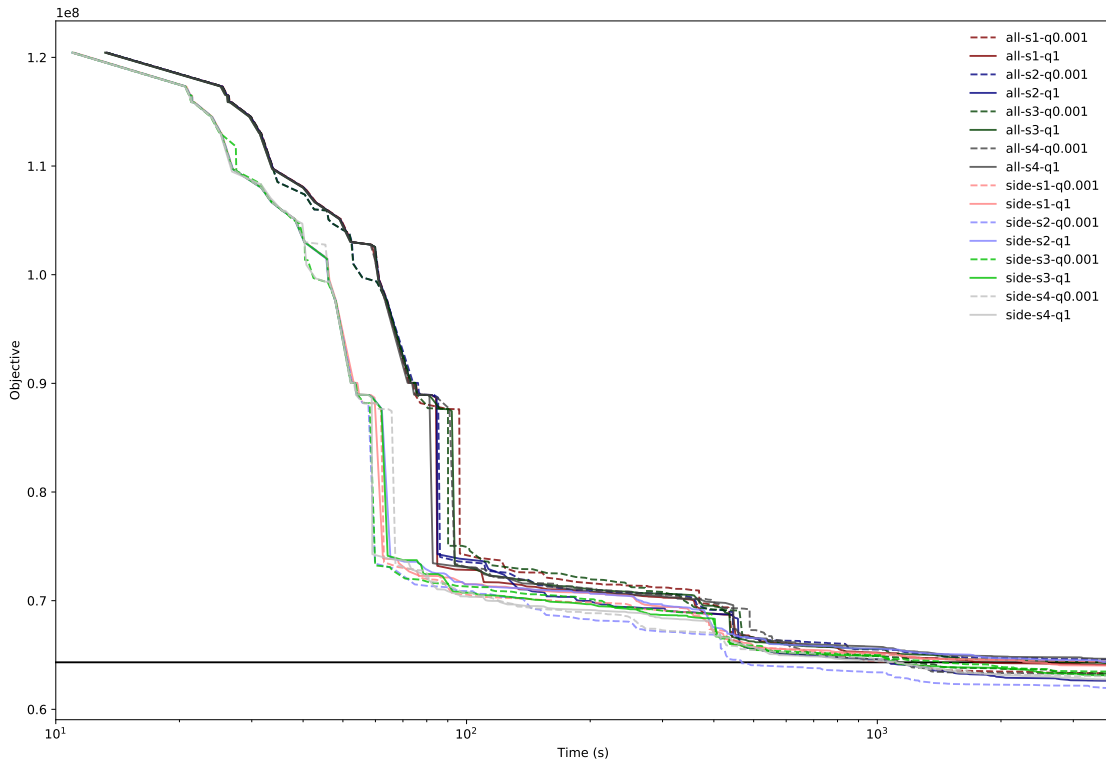


(b) BL3-cold

Figure 3.3: Examples of the objective progression in comparison to different parameter choices for parallel $mns+tns$ in comparison to mns .



(a) R1L1v-cold



(b) R4L4v-cold

Figure 3.4: Examples objective progression in comparison to different heuristics in the complete concurrent solver.

3. Tropical Neighbourhood Search: A New Heuristic for Periodic Timetabling 109

	all							
	s1-q0.001	s1-q1	s2-q0.001	s2-q1	s3-q0.001	s3-q1	s4-q0.001	s4-q1
BL1-warm	6792526	6935103	6547850	6697639	6572855	6740896	6562891	7025590
BL1-cold	7621147	7457877	6465738	6758000	7144032	6699148	7147572	6911380
BL3-warm	7334701	7140000	7259769	7023881	6974763	7206833	7553069	7164378
BL3-cold	7406444	7727433	7629390	7753854	7768767	7233719	7462949	7716933
R1L1-warm	30426994	30423140	30423140	30426994	30426994	30426994	30431036	30426994
R1L1-cold	34020450	33522247	31692344	34177686	31856836	33795188	34125594	33794486
R1L1v-warm	42943355	42943355	42943355	42814750	42801531	42943355	42943355	42943355
R1L1v-cold	46169674	46551486	46041342	46326249	45768586	46341535	47172536	45813504
R2L2-warm	43207702	43319135	43206244	43275789	43263142	43329691	43047738	43329691
R2L2-cold	42361958	42416233	41640213	43131819	42337570	42512473	42586419	41960342
R3L3-warm	44408363	44397465	44379012	44399516	44378435	44371830	44397486	44411156
R3L3-cold	41222660	42739161	41228048	42440292	42384919	40483617	42758160	44132022
R4L4-warm	36909735	36916544	36901735	36935621	36928689	36960219	36997153	36990506
R4L4-cold	41823843	41243736	43339597	42061213	42174340	40282996	43336050	41504147
R4L4v-warm	64330043	64328991	64340252	64285960	64340252	64339747	64339747	64340252
R4L4v-cold	63222568	64090696	64580054	62632707	63302814	64225355	63173171	64649625
6 min. ranking	12.5	8.62	9.5	9.88	7.56	10.19	10.38	10.31
final ranking	8.69	9.19	6.75	9.19	7.94	8.06	11.0	10.5

	side							
	s1-q0.001	s1-q1	s2-q0.001	s2-q1	s3-q0.001	s3-q1	s4-q0.001	s4-q1
BL1-warm	6527346	6593280	6429697	6504754	6483936	6508182	6570960	6533503
BL1-cold	6542043	7101531	6650416	7195907	6455312	6791979	6808078	6649762
BL3-warm	6871983	7308628	7341305	7030706	7362216	7040927	6909267	6903738
BL3-cold	7163591	7504180	7349736	7614397	7514692	7232455	7212076	7247561
R1L1-warm	30426994	30426994	30426994	30423140	30425260	30426994	30426994	30425260
R1L1-cold	32816267	33578895	33551641	33642348	33857174	33321098	33785910	33708393
R1L1v-warm	42886458	42943355	42943355	42943355	42943355	42943355	42943355	42943355
R1L1v-cold	44544618	46040263	46330633	45723589	46295094	45923497	46228814	46390710
R2L2-warm	43203025	43318930	43191843	43004597	43318930	43222313	43100634	43047991
R2L2-cold	41095503	42522032	41856192	42195345	42390560	42168432	41914851	42347940
R3L3-warm	44430322	44382720	44433727	44414666	44321035	44404928	44385201	44400190
R3L3-cold	41391010	41773723	41985509	41473233	40993187	42284859	40959638	44300876
R4L4-warm	36974381	36923867	36953228	36915142	36935274	37064318	36913014	36814620
R4L4-cold	42207683	41930820	41605374	42155011	41124227	41549938	41784069	40541428
R4L4v-warm	64340252	64340252	64250155	64339747	64339747	64339747	64339747	64339747
R4L4v-cold	64389979	64026343	61968380	64348631	63482236	63127690	63036902	62757263
6 min. ranking	4.19	6.31	5.69	5.31	4.75	6.06	5.31	6.25
final ranking	6.5	9.0	6.69	7.06	7.0	6.81	5.56	6.06

Table 3.7: Objective values after 1h runtime with the complete strategy.

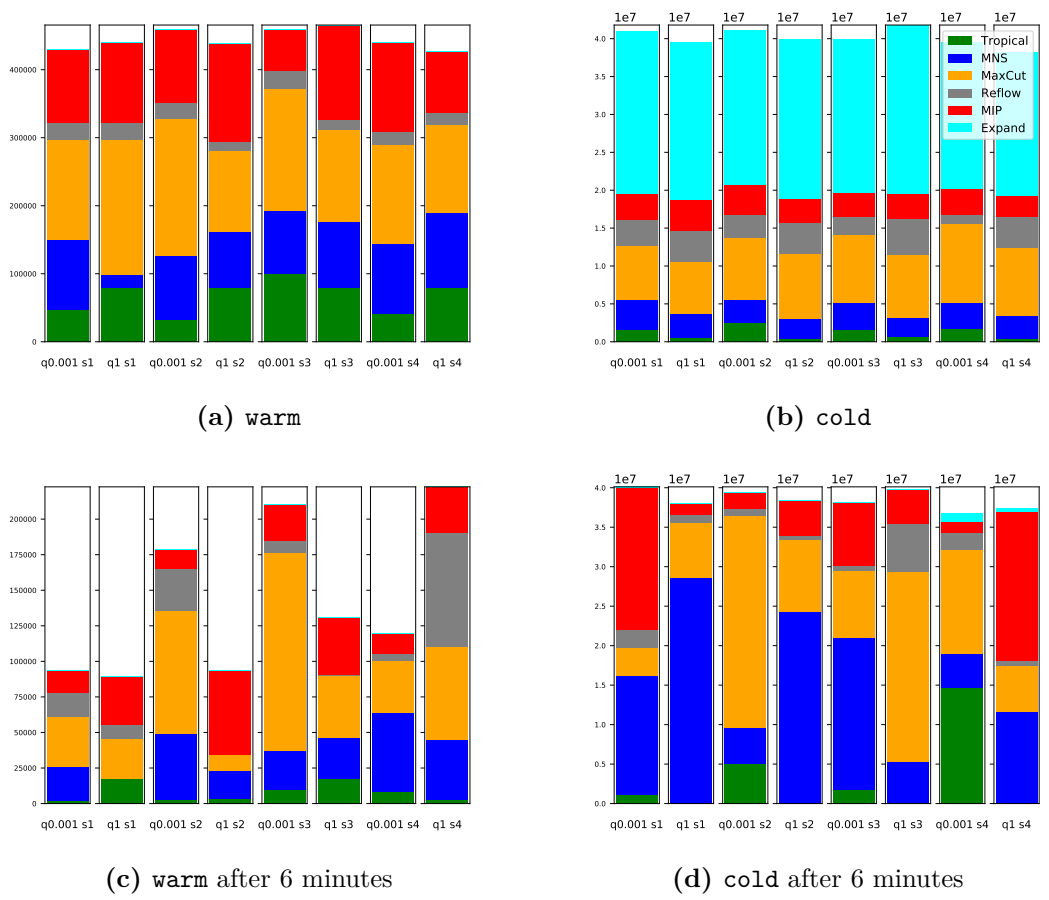


Figure 3.5: Contribution of the individual methods in the concurrent solver and *tns* (green) for the instance R3L3.

CHAPTER **4**Periodic Timetabling
with Cyclic Order Constraints

Bortoletto, E., Lindner, N., Masing, B.
23rd Symposium on Algorithmic Approaches for Transportation Modelling,
Optimization, and Systems (ATMOS 2023); 7:1-7:18; 115.
DOI: 10.4230/OASICS.ATMOS.2023.7.

This article is licensed under a Creative Commons Attribution 4.0 International License.
To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

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.

Contents

4.1	Introduction	113
4.2	The Periodic Event Scheduling Problem	115
4.3	The Infrastructure-Aware Periodic Event Scheduling Problem	117
4.3.1	Infrastructure Awareness	117
4.3.2	Cyclic Orders	124
4.3.3	Propagating Cyclic Orders and Chronological Constraints	127
4.4	Computational Results	129
4.4.1	Instances	129
4.4.2	Maximal Infrastructure Elements	130
4.4.3	Experiments	130
4.4.4	Interpretation of Results	133
4.5	Future Work	134
4.A	Appendix	135

4.1 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., [71]), 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% [72]. 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 [71], 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) [5]. There is a decent amount of literature on periodic timetabling using PESP (e.g., [18, 24, 26, 42, 73]), 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 [18]. 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 [34]. 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., [74]).

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 [34] that resolve the track occupation problem.

Not unexpectedly, solving *Infrastructure-Aware* PESP is challenging: PESP alone is an NP-hard optimization problem [5], and even medium-sized instances have withstood attempts to solve them to optimality. For example, none of the instances of the benchmark library PESPLib [32] have been solved to optimality, even though a variety of algorithms is available [2, 24, 31, 36, 40]. 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 4.2 we recall the basics of PESP. We introduce Infrastructure-Aware PESP and investigate a few theoretical properties in Section 4.3.1. Cyclic orders enter the picture in Section 4.3.2, and we describe how to work with them algorithmically in Section 4.3.3. Moving forward, we dedicate Section 4.4.1 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 4.4.2, we finally present in Section 4.4.3 the results and interpretations of our computational experiments. Section 4.5 ends the paper with our ideas for further research.

4.2 The Periodic Event Scheduling Problem

The *Periodic Event Scheduling Problem* (PESP) [5] 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 $w \in \mathbb{R}_{\geq 0}^{A(G)}$.

Definition 4.1 ([5]). — *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*

$$(a) \quad \pi_j - \pi_i \equiv x_a \pmod{T} \text{ for all } a = (i, j) \in A(G),$$

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

$$(c) \quad w^\top x \text{ is minimum,}$$

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., [42, Theorem 9.8]).

We assume that ℓ and u are integral, so that by [24] 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 [42].

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 4.3.1. 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 [18] for further modeling aspects of PESP.

Several mixed-integer programming formulations for PESP are known [42]. We focus on the cycle-based formulation, which relies on the cycles of an integral cycle basis B of G [45]:

$$\min \quad \sum_{a \in A(G)} w_a x_a \quad (4.1a)$$

$$\text{s.t.} \quad \sum_{a \in A(G)} \gamma_a x_a = T z_\gamma \quad \gamma \in B, \quad (4.1b)$$

$$\ell_a \leq x_a \leq u_a \quad a \in A(G), \quad (4.1c)$$

$$z_\gamma \in \mathbb{Z} \quad \gamma \in B. \quad (4.1d)$$

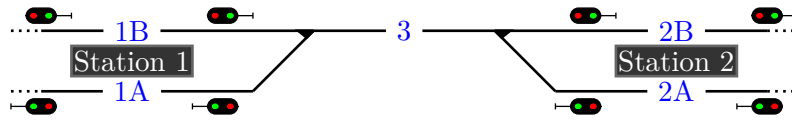
4.3 The Infrastructure-Aware Periodic Event Scheduling Problem

4.3.1 Infrastructure Awareness

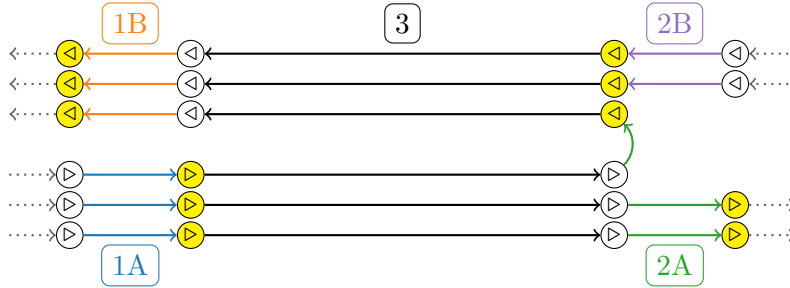
Having railway timetabling in mind, we will be working with a special version of PESP that is “*infrastructure-aware*”. Along with a PESP instance (G, T, ℓ, u, w) we also have an *infrastructure map* $\eta: \mathcal{A} \rightarrow E$, where $\mathcal{A} \subseteq A(G)$, and E is a set of *infrastructure 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 4.1.

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 4.2):



(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 4.1a. 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 4.1: An interpretation of Infrastructure-Aware PESP in the context of railway timetabling.

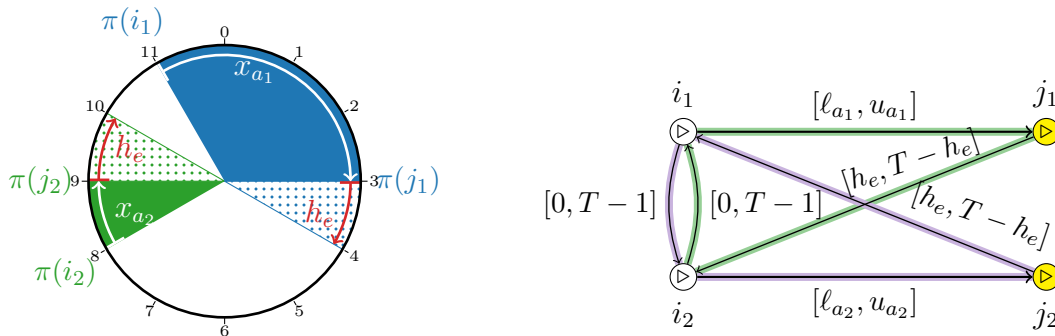


Figure 4.2: A visualization of two scheduled activities $a_1 = (i_1, j_1), a_2 = (i_2, j_2) \in A_e$ for some $e \in E$ on a clock, $T = 12$. Definition 4.2 requires that the distance $[\pi(i_2) - \pi(i_1)]_T$ must be at least x_{a_1} (filled blue sector) + h_e (dotted blue sector).

Figure 4.3: 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 [34], the Q3 constraints are equivalent to the activity separation constraints (4.2).

Definition 4.2. — Let (G, T, ℓ, u, w) be a PESP instance, let $\eta: \mathcal{A} \rightarrow E$ be an infrastructure map, and let $h \in \mathbb{R}_{\geq 0}^E$. 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

$$[\pi_{i_2} - \pi_{i_1}]_T \geq x_{a_1} + h_e \quad (4.2)$$

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 4.1. — To avoid the degeneracy that arises when $x_{a_1} = h_e = 0$ in (4.2), 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 (4.2) 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 [34]. For reasons that will become apparent later, we do not model the activity separation constraints (4.2) directly, but we choose to use the equivalent “Q3” formulation introduced in [34]. To do so (see also Figure 4.3), 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 $a^I = (j_1, i_2)$ with bounds $[h_e, T - h_e]$, as well as a headway arc $a^{II} = (i_2, i_1)$ with bounds $[0, T - 1]$, thereby creating a directed 3-cycle $q(a_1, a_2)$ on the arcs a_1, a^I, a^{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:

$$\min \quad \sum_{a \in A(G^h)} w_a x_a \quad (4.3a)$$

$$\text{s.t.} \quad \sum_{a \in A(G^h)} \gamma_a x_a = T z_\gamma \quad \gamma \in B^h, \quad (4.3b)$$

$$\ell_a \leq x_a \leq u_a \quad a \in A(G^h), \quad (4.3c)$$

$$z_\gamma \in \mathbb{Z} \quad \gamma \in B^h, \quad (4.3d)$$

$$\text{(Q3 constraints)} \quad \sum_{a \in q(a_1, a_2)} x_a = T \quad e \in E, a_1, a_2 \in A_e, a_1 \neq a_2 \quad (4.3e)$$

Note that we express the Q3 constraints in terms of periodic tensions rather than of periodic offset variables as was done in [34].

Remark 4.2. — *The number of Q3 constraints in (4.3) 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 4.1. — *Consider a feasible instance for Infrastructure-Aware PESP. Then there is an optimal solution (x, z) to (4.3) 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 (4.3). 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

$$P := \{x \in \mathbb{R}^{A(G^h)} \mid \Gamma x = T z^*, Qx = T, \ell \leq x \leq u\}, \quad (4.4)$$

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. \square

We quickly note that the Q3 constraints and the positivity assumption (Remark 4.1) have implications on upper bounds.

Lemma 4.2. — *Let (x, z) be a feasible solution to (4.3), 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^1 in (4.3), and hence $0 \leq \ell_a \leq x_a \leq T - x_{a^1} \leq T - h_e < T$. 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 (4.2), we find $x_{a_1} \leq [\pi_{i_2} - \pi_{i_1}]_T < T$, so that again $x_{a_1} = [\pi_{j_1} - \pi_{i_1}]_T$. \square

With that, we can derive the following degree bounds.

Lemma 4.3. — *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*

$$\forall i \in V(G) : \quad \deg_{\mathcal{A}'}(i) \leq |E'|, \quad (4.5)$$

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*

$$\forall i \in V(G) : \quad \max \{ \delta_{\mathcal{A}'}^+(i), \delta_{\mathcal{A}'}^-(i) \} \leq |E'|, \quad (4.6)$$

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 (4.2), we have

$$0 < x_{a_1} + h_e \leq [\pi_i - \pi_i]_T = 0, \quad (4.7)$$

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 (4.2),

$$[\pi_k - \pi_j]_T \geq x_{a_1} + h_e \quad \text{and} \quad [\pi_j - \pi_k]_T \geq x_{a_2} + h_e. \quad (4.8)$$

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 4.2,

$$[\pi_k - \pi_j]_T \geq x_{ji} + h_e = [\pi_i - \pi_j]_T + h_e = [\pi_i - \pi_k]_T + [\pi_k - \pi_j]_T + h_e > [\pi_k - \pi_j]_T, \quad (4.9)$$

which cannot be either. Finally, they could be of the form $a_1 = (j, i)$ and $a_2 = (i, k)$, and again by (4.2) and noting that $x_{a_1} = [\pi_i - \pi_j]_T$ due to Lemma 4.2,

$$x_{a_1} = [\pi_i - \pi_j]_T \geq x_{a_1} + h_e > x_{a_1} \quad (4.10)$$

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 (4.7) is still valid due to $x_{a_1} \geq \ell_{a_1} > 0$. Moreover, (4.9) holds because $[\pi_i - \pi_k]_T = x_{a_2} \geq \ell_{a_2} > 0$ by Lemma 4.2. 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). \square

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 4.4. — Consider an instance of Infrastructure-Aware PESP with infrastructure map $\eta: \mathcal{A} = A(G) \rightarrow \{e\}$ 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 4.3. □

Corollary 4.5. — 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 (4.3) to satisfy $\sum_{a \in A(G)} x_a \leq T$ in order to separate all arcs from each other. By Theorem 4.4, 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

$$\min\{w^\top x \mid \gamma^\top x = T, \ell \leq x \leq u\}, \quad (4.11)$$

□

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

4.3.2 Cyclic Orders

We have seen in Corollary 4.5 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 4.3 ([75]). — *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 4.4. — *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 4.6. — *Let $(G, T, \ell, u, w, \eta, h)$ be an Infrastructure-Aware PESP instance, let x be a feasible solution to (4.3) 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

$$\sum_{a \in A_e} x_a + \sum_{k=0}^{n-1} [\pi_{i_{[k+1]_n}} - \pi_{j_k}]_T = T. \quad (4.12)$$

(c) The following constraint implies that π respects Δ_e and all Q3 constraints associated to e in (4.3):

$$\sum_{a \in Q(\Delta_e)} x_a = T \quad (4.13)$$

where $Q(\Delta_e)$ is the directed cycle in G^h consisting of the arcs in A_e and the headway arcs $a_{a_k, a_{[k+1]_n}}^l$ 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 4.2.

(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

$$[\pi_{j_0} - \pi_{i_0}]_T + [\pi_{i_1} - \pi_{j_0}]_T + \dots + [\pi_{j_{n-1}} - \pi_{i_{n-1}}]_T + [\pi_{i_0} - \pi_{j_{n-1}}]_T = T, \quad (4.14)$$

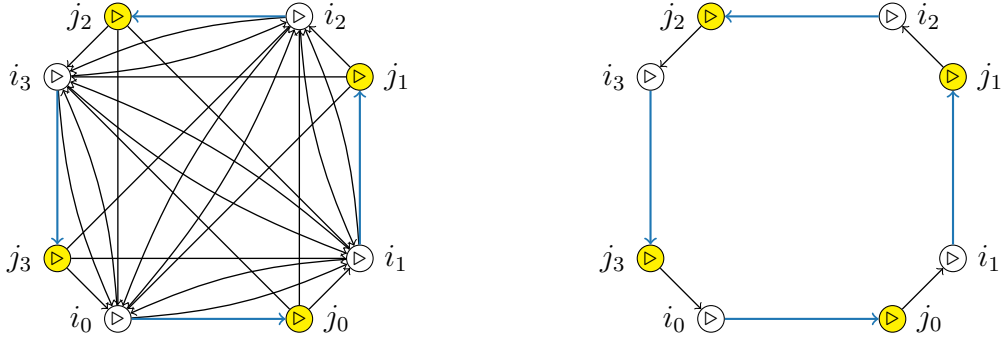
because the left-hand side is congruent to 0 modulo T , and $[\cdot]_T$ can in fact be omitted except at exactly one summand. Due to the positivity assumption, $[\pi_{j_k} - \pi_{i_k}]_T = x_{a_k}$ for all k , so that (4.12) is equivalent to (4.14).

(c) We first note $x_{a_{a_k, a_{[k+1]_n}}^l} = [\pi_{j_{[k+1]_n}} - \pi_{i_k}]_T$. Hence, if (4.13) holds, then (4.12) 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 ,

$$\sum_{a \in q(a_k, a_l)} x_a = [\pi_{j_k} - \pi_{i_k}]_T + [\pi_{i_l} - \pi_{j_k}]_T + [\pi_{i_k} - \pi_{i_l}]_T = T. \quad (4.15)$$

□

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



(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 4.4: Arcs in the Q3 formulation for Infrastructure-Aware PESP vs. (4.13) 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 4.4b in the digraph Figure 4.4a built by the union of the cycles $q(a_k, a_l)$ for $a_k, a_l \in A_e$, $k \neq l$.

Definition 4.5. — 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 4.3.3).

Theorem 4.6 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 (4.13) to (4.3). A very elegant consequence of Theorem 4.6 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 4.4 visualizes this effect.

Remark 4.3. — *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 4.2. 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 4.1 carries over to Infrastructure-Aware Fixed-Cycle-Order PESP.*

4.3.3 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.

4.3.3.1 Identifying Maximal Infrastructure Elements

Let \mathcal{T} denote a set of trips and let $\tau : \mathcal{A} \rightarrow \mathcal{T}$ 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 4.5.

4.3.3.2 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 $\sigma_\Delta^e \in \{0, 1\}$ for each cyclic order Δ on A_e , and enforce Δ or not via the big- M constraints

$$\sum_{a \in Q(\Delta)} x_a \leq T\sigma_\Delta^e + T|A_e|(1 - \sigma_\Delta^e) \quad e \in E, \Delta \text{ cyclic order on } A_e \quad (4.16)$$

$$\sum_{a \in Q(\Delta)} x_a \geq T\sigma_\Delta^e + 2T(1 - \sigma_\Delta^e) \quad e \in E, \Delta \text{ cyclic order on } A_e \quad (4.17)$$

$$\sum_{\Delta \text{ cyclic order on } A_e} \sigma_\Delta^e = 1 \quad e \in E \quad (4.18)$$

which are derived from (4.13). If $\sigma_\Delta^e = 1$, then (4.16) and (4.17) 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 (4.17) 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 (4.3). 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

$$\sigma_{\Delta}^e \leq \sum_{\substack{\Delta' \text{ cyclic order on } A_{e'} \\ \text{restricting to } \Delta \text{ on } A_e}} \sigma_{\Delta'}^{e'} \quad e \in E, \Delta \text{ cyclic order on } A_e \quad (4.19)$$

$$\sum_{\substack{\Delta \text{ cyclic order on } A_e \\ \text{induced by } \Delta' \text{ on } A_{e'}}} \sigma_{\Delta}^e \leq \sigma_{\Delta'}^{e'} \quad e \in E, \Delta' \text{ cyclic order on } A_{e'} \quad (4.20)$$

Note that using the sum in (4.19) and (4.20) is justified by (4.18).

4.4 Computational Results

4.4.1 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

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

Table 4.1: Characteristics of our two instances.

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 4.1.

4.4.2 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 4.3.3.1 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 4.5.

4.4.3 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 [5], which proved to be unreasonably slower than the cycle formulation

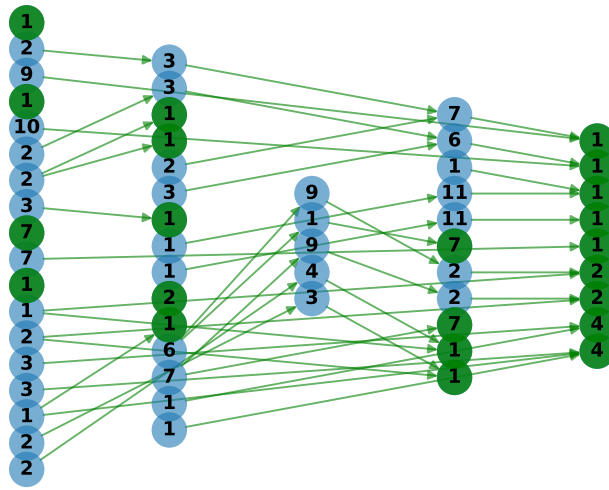


Figure 4.5: The directed acyclic graph induced by \preceq for both instances has 22 sinks (green), which identify the maximal infrastructure elements as in Section 4.3.3.1. The vertex labels indicate the number of infrastructure elements that are equivalent w.r.t. \preceq . The five columns show from left to right the infrastructure elements used by 3 to 7 trains within 20 minutes, we omitted the ones with 2 trains or less.

even at solving trivial instances, and so all tests presented here use the cycle formulation, as in (4.1). 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_{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_{span} .

First and foremost, we tested `i1`, modeled as seen in (4.3). Using B_{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_{bfs} , no primal solution was found within 3 hours, with almost no dual bound to speak of either. Nonetheless B_{bfs} proved to be quite good when many orders are fixed.

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_{bfs} and B_{span} . For the transfer-less instance, results of these tests can be seen in Table 4.2. The same tests were conducted on `i2`, whose results can be found in Table 4.3. 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_{span} and `i1`, in Table 4.4 we show various test results that include the σ variables introduced in Section 4.3.3. 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 (4.16) and (4.18). Test configurations marked by the letter *b*, also include equations as seen in (4.17), bounding below. Test configurations marked by the letter *l*, also include equations as seen in (4.19) and (4.20), linking orders for compatibility. Finally, test configurations marked by the letter *r* are ones where 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.

4.4.4 Interpretation of Results

As expected, fixing order information greatly improves solving time, as seen in Table 4.2, although the cycle basis choice remains quite influential. In general it can be observed that the more orders are fixed, the more B_{bfs} is faster than B_{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 4.3 confirm the same pattern shown in the previous table. In fact, faster times in Table 4.2 are almost invariably matching to smaller optimality gaps in Table 4.3.

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 4.4, 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 (4.17), 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.

4.5 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.

Authors' Contributions

Conceptualization, N.L.; methodology, N.L. and E.B.; software, E.B.; validation, E.B., N.L., and B.M.; formal analysis, E.B. and N.L.; investigation, E.B. and N.L.; resources, B.M.; data curation, E.B.; writing—original draft preparation, E.B., N.L., and B.M.; writing—review and editing, N.L., B.M., and E.B.; visualization, N.L. and B.M.; supervision, N.L.; project administration, N.L.; funding acquisition, N.L..

All authors have read and agreed to the published version of the manuscript.

4.A Appendix

Test configuration	Time to primal (s)	Time to optimal (s)	Opt. 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
$[i \leq 6] = [i \neq 7]$	29	68	"
$[i \leq 7] = [i \geq 3]$	1	12	3967
$[i \leq 7] = [i \geq 3]$	54	55	"

Table 4.2: Fixed order test on *i1*, with cycle basis B_{bfs} in the white rows, and cycle basis B_{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)	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 4.3: Fixed order test on i2, with cycle basis B_{bfs} in the gray rows, and cycle basis B_{span} in the white rows. The time limit for each test was 15 minutes.

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

Table 4.4: Tests with σ -constraints on i1, in various configurations, using cycle basis B_{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.

CHAPTER 5

Periodic Event Scheduling
with Flexible Infrastructure Assignment

Bortoletto, E., van Lieshout, R. N., Masing, B., Lindner, N.
24th Symposium on Algorithmic Approaches for Transportation Modelling,
Optimization, and Systems (ATMOS 2024); 4:1-4:18; 123.
DOI: 10.4230/OASICS.ATMOS.2024.4.

This article is licensed under a Creative Commons Attribution 4.0 International License.
To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Abstract We present novel extensions of the Periodic Event Scheduling Problem (PESP) that integrate the assignment of activities to infrastructure elements. An application of this is railway timetabling, as station and platform capacities are limited and need to be taken into account. We show that an assignment of activities to platforms can always be made periodic, and that it can be beneficial to allow larger periods for the assignment than for the timetable. We present mixed-integer programming formulations for the general problem, as well as for the practically relevant case when multiple platforms can be considered equivalent, for which we present a bipartite matching approach. We finally test and compare these models on real-world instances.

Contents

5.1	Introduction	141
5.2	Periodic Event Scheduling and Infrastructure Awareness	142
5.3	General Infrastructure Awareness with Flexible Infrastructure Maps	145
5.3.1	Problem Definition and Periodizability of Assignments	146
5.3.2	Pattern Functions and Conflict-Freeness	148
5.3.3	A MIP Formulation for IPESPA	151
5.4	Partitionable Infrastructure Maps	153
5.5	Experiments	160
5.5.1	Instances	160
5.5.2	IPESP Experiments	161
5.5.3	IPESPC Experiments	162
5.6	Conclusion	164

5.1 Introduction

Out of the many interacting pieces of public transportation services, a key determinant in the puzzle is the timetable. This is an omnipresent yet flexible concern in the operators' minds [76], as well as one of the data of most interest to passengers [77]. These are but two of the reasons for which timetabling and in particular periodic timetabling has received substantial attention in the past. The modeling framework of the Periodic Event Scheduling Problem (PESP) was initially formulated by Serafini and Ukovich [5], quickly found to have rich and interesting underlying structures [2, 26, 36, 78], and studied ever since, also in integration with various other problems, such as [34, 35, 74, 79, 80].

Of special interest in this work is the question of solving PESP while making sure that the produced timetable accounts for various infrastructural constraints of high practical interest in railway operations, involving safety of operations and physical occupation of tracks. In particular, we focus on what is called *track occupation problem* in [34], which entails ensuring that no two vehicles are ever scheduled to occupy the same point in time and space. This is of particular interest, for example, when planning dwelling activities of trains at the same platform. The need to respect a given association of the activities to be scheduled to infrastructure elements led to *Infrastructure-Aware PESP* (IPESP), which can be formulated as a mixed-integer linear program, by using well-known PESP constraints as foundation [3, 34].

We present two main contributions: At first, we generalize IPESP to *Infrastructure-Aware PESP with Assignment* (IPESPA), by integrating the assignment of activities to infrastructure elements into the periodic timetabling problem. We show that such an assignment can always be made periodic, without impact on the timetable, and highlight how it can be advantageous for the period of such an infrastructure assignment to be larger than the one of the timetable, with an elucidatory example. We then also present a mixed-integer programming formulation for IPESPA, that generalizes the so-called Q0-constraints of [34].

The setting above is for the general case, in which we allow any map of activities to sets of infrastructure elements. In a second step, we then restrict our inquiry to a more common and practical use-case, where multiple infrastructure elements can be considered as equivalent. We then assume that activities can only be assigned to one element, but this element is allowed to have a capacity larger than one. For example, the two sides of a platform are oftentimes equivalent options to choose, and we might consider such a platform as an element with capacity two. The main achievement is a mixed-integer programming formulation for this *Infrastructure-Aware PESP with Capacities* (IPESPC), a model much more compact than the IPESPA model, based on matchings in certain auxiliary bipartite graphs. Finally, this allows us also to derive two new alternative mixed-integer programming formulations for standard IPESP, i.e., IPESPC with unit capacities, beyond those of [3, 34].

We compare our new formulations on three realistic instances, both in the case of unit and larger capacities, and demonstrate their computational feasibility and practical benefit.

section 5.2 recalls the Periodic Event Scheduling Problem and its infrastructure-aware extension IPESP. section 5.3 introduces IPESPA, discusses the theory of general infrastructure assignments, and presents a mixed-integer programming formulation. In section 5.4, we restrict to IPESPC and present a matching-based MIP model, along with the resulting new formulations for IPESP. We evaluate the computational power of our models in section 5.5, before concluding the paper in section 5.6.

The present work is a direct consequence of the fruitful connections and conversations that were had during ATMOS 2023 [81], and we thereby thank the organizing committee for fostering the transport optimization community.

5.2 Periodic Event Scheduling and Infrastructure Awareness

The Periodic Event Scheduling Problem (PESP) is the standard model to compute and optimize timetables for public transport. It is formulated as follows.

Definition 5.1 ([5]). — Consider a directed graph G with vertex set $V(G)$ and arc set $A(G)$, together with $T \in \mathbb{N}$, vectors $\ell, u \in \mathbb{R}^{A(G)}$, and $w \in \mathbb{R}_{\geq 0}^{A(G)}$. The Periodic Event Scheduling Problem (PESP) is to find vectors $\pi \in \mathbb{R}^{V(G)}$ and $x \in \mathbb{R}^{A(G)}$ such that

$$(a) \quad \pi_j - \pi_i \equiv x_a \pmod{T} \text{ for all } a = (i, j) \in A(G),$$

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

$$(c) \quad w^\top x \text{ is minimum,}$$

or to decide that no such π and x exist.

In public transportation practice the directed graph G is oftentimes a so called *event-activity network*. There nodes $V(G)$ are the events, typically arrival or departure events, whereas arcs $A(G)$ are the activities, typically driving from a departure to an arrival, or dwelling from an arrival to a departure. The number $T \in \mathbb{N}$ is the *period time*, and determines after how long each event should repeat. Then the vectors π and x sought by PESP are called *periodic timetable* and *periodic tension*, respectively, where the former represents T -periodic timestamps denoting at which point of each period each event should occur, and the latter instead denotes the duration of the activities in-between events. PESP instances are commonly denoted as (G, T, ℓ, u, w) .

Note that the simultaneous use of timetable and tension variables is primarily for ease of expression, since one can always be recovered from the other. In fact, given a periodic timetable π , a corresponding tension is quickly found by setting $x_a := [\pi_j - \pi_i - \ell_a]_T + \ell_a$ for every $a = (i, j) \in A(G)$, where $[\cdot]_T$ denotes the modulo T operator with values in $[0, T)$. Likewise, given a periodic tension, a corresponding timetable is quickly found by a connected graph traversal [42, Theorem 9.8].

For in-depth analysis of the many properties of PESP, we refer to the literature, starting with [18]. Multiple mixed-integer program formulations for PESP are known [42]. For simplicity, here we choose the *standard formulation* [5], which

models PESP by linearizing the modulo constraints by use of auxiliary integer variables p_{ij} , called periodic offsets:

$$\min \quad \sum_{(i,j) \in A(G)} w_{ij} x_{ij} \quad (5.1a)$$

$$\text{s.t.} \quad \pi_j - \pi_i + T p_{ij} = x_{ij} \quad \forall (i, j) \in A(G), \quad (5.1b)$$

$$0 \leq \pi_i < T \quad \forall i \in V(G), \quad (5.1c)$$

$$\ell_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A(G), \quad (5.1d)$$

$$p_{ij} \in \mathbb{Z} \quad \forall (i, j) \in A(G). \quad (5.1e)$$

Now we continue with the basic extension of PESP with infrastructure awareness, as per [3]. First of all we define an *infrastructure map* $\eta: \mathcal{A} \rightarrow E$, mapping certain arcs $\mathcal{A} \subseteq A(G)$ to a set of *infrastructure elements* E . This map encodes an assignment of activities to infrastructure, implying where those activities will physically take place within the network. We denote $\mathcal{A}_e := \eta^{-1}(e)$, for $e \in E$. For each infrastructure element we also have a minimum headway time $h \in \mathbb{R}_{\geq 0}^E$, indicating how long this element needs to be unoccupied between uses of different vehicles. As in [3], we assume that for each $e \in E$ either $h_e > 0$ or $\ell_a > 0$ for all $a \in \mathcal{A}_e$, to avoid pathological cases.

Considering then arcs $a_1 = (i_1, j_1)$ and $a_2 = (i_2, j_2)$ in \mathcal{A} , and such that $a_1 \neq a_2$ and $\eta(a_1) = \eta(a_2)$, we say that a_1 does not *h-overlap* a_2 if it holds that

$$[\pi_{i_2} - \pi_{i_1}]_T \geq x_{a_1} + h_e. \quad (5.2)$$

The constraint (5.2) is called Q0-constraint in [34], but there is also another possible equivalent formulation, namely the Q4-constraint (“butterfly constraint”). These entail, for each pair of arcs a_1 and a_2 as above, the addition of auxiliary arcs (j_1, i_2) and (j_2, i_1) with lower bound h_e and upper bound $T - h_e$, and then imposing total tension exactly equal to T along the 4-cycle $q(a_1, a_2) := (i_1, j_1, i_2, j_2, i_1)$. That is,

$$\sum_{a \in q(a_1, a_2)} x_a = T. \quad (5.3)$$

Then (5.3) holds for $q(a_1, a_2)$ if and only if a_1 and a_2 do not *h-overlap* each other [34].

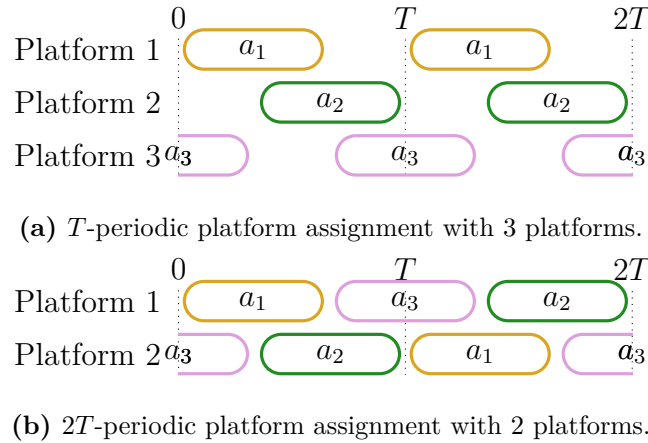


Figure 5.1: Two platform assignments of the same three activities.

We say that a set $S \subseteq \mathcal{A}$ is h -conflict-free if no arc in S h -overlaps another, and furthermore $x_a + h_e \leq T$ for all $e \in E$ and all $a \in S \cap \mathcal{A}_e$.

Definition 5.2 ([3]). — Let (G, T, ℓ, u, w) be a PESP instance, let $\eta: \mathcal{A} \rightarrow E$ be an infrastructure map, and let $h \in \mathbb{R}_{\geq 0}^E$. The Infrastructure-Aware PESP (IPESP) is to find a solution (π, x) to PESP on (G, T, ℓ, u, w) such that \mathcal{A} is h -conflict-free and the solution is optimal, or to decide that no such solution exists.

The PESP mixed-integer program (5.1), together with either all necessary Q0-constraints (5.2) or Q4-constraints (5.3), solves Infrastructure-Aware PESP.

This extended form of PESP implicitly assumes two rather strict properties. Firstly, the map η is, by definition, mapping each arc in \mathcal{A} to a single $e \in E$, thereby presuming that such an infrastructure assignment has already been fixed. This implies that every activity must repeat every period always on the same infrastructure. However common, this need not be the case, and as we will see in the next section, it shall not.

5.3 General Infrastructure Awareness with Flexible Infrastructure Maps

Let us begin by considering the following illustrative example, motivating our work.

Example 5.1. — Consider an IPESP situation where we have $T = 30$ minutes, and three dwelling activities $a_k = (i_k, j_k)$ for $k \in \{1, 2, 3\}$. Suppose further that there are three platforms e_1, e_2, e_3 with $\eta(a_k) = e_k$ for all $k \in \{1, 2, 3\}$, without headway requirements. Let $\pi_{i_1} = 0$, $\pi_{i_2} = 10$, $\pi_{i_3} = 20$, and $\pi_{j_1} = 20$, $\pi_{j_2} = 0$, $\pi_{j_3} = 10$, meaning that each dwelling activity is scheduled for 20 min. In Figure 5.1a, we see how this would play out. Crucially, this leaves each platform unoccupied for 10 minutes per period of 30 minutes.

Suppose we were now to allow all activities to be assigned to either platform, and possibly to different platforms at different periodic repetitions. Then, the configuration of Figure 5.1b would be possible. With the same timetable as before, only two platforms are required now. This enables a more efficient use of the existing infrastructure.

Moreover, IPESP always assumes that no infrastructure element can be occupied for longer than T . This might however be practically necessary, e.g., due to regulations on minimum turnaround times.

5.3.1 Problem Definition and Periodizability of Assignments

We now present the tools to do timetabling while ensuring efficient use of the underlying infrastructure. Consider the flexible infrastructure map $\eta: \mathcal{A} \rightarrow \mathcal{E} \subseteq 2^E$, and so having the option to choose where to have activities occur. Given a periodic timetable π and tension x on a PESP instance (G, T, ℓ, u, w) , we call

$$\mathcal{I} := \left\{ I_a^{(k)} \mid a \in \mathcal{A}, k \in \mathbb{Z} \right\} \quad (5.4)$$

the *realisation* of (π, x) , where $I_a^{(k)} := [\pi_i + kT, \pi_i + kT + x_{ij}]$ for $k \in \mathbb{Z}$ and $a = (i, j) \in \mathcal{A}$. Now, an *infrastructure assignment* is any map $\nu: \mathcal{I} \rightarrow E$, and we say it is *valid* if we have that $\nu(I_a^{(k)}) \in \eta(a)$ for all $I_a^{(k)} \in \mathcal{I}$. Furthermore, for a given infrastructure assignment ν and an infrastructure element $e \in E$ we define

$$\mathcal{I}_e := \left\{ [\pi_i + kT, \pi_i + kT + x_{ij} + h_e] \mid \forall I_a^{(k)} \in \mathcal{I}: \nu(I_a^{(k)}) = e \right\}, \quad (5.5)$$

and we say ν is *h-conflict-free* if the intervals in \mathcal{I}_e are pairwise disjoint for every $e \in E$.

It now comes natural to formulate the following.

Definition 5.3. — *Let (G, T, ℓ, u, w) be a PESP instance and $\eta: \mathcal{A} \rightarrow \mathcal{E} \subseteq 2^E$ an infrastructure map, and let $h \in \mathbb{R}_{\geq 0}^E$. The Infrastructure-Aware PESP with Assignment (IPESPA) is to find a solution (π, x) to PESP on (G, T, ℓ, u, w) , together with a valid and *h-conflict-free* infrastructure assignment ν , such that the solution is optimal, or to decide that no such solution exists.*

It is clear that were η not to be actually flexible, meaning $|\eta(a)| = 1$ for every $a \in \mathcal{A}$, then we would fall back into Definition 5.2 by fixing the only possible assignment $\nu(I_a^{(k)}) := \eta(a)$ for all intervals in the realisation. Otherwise, this problem formulation allows for full flexibility in the choice of infrastructure, which can change after any periodic repetition, within the limits of η . This lack of structure and predictability may seem to be an issue of design, since the solutions could even become indescribable in finite terms. Thankfully, this will turn out not to be an issue. We say an infrastructure assignment ν is ω -periodic, for some $\omega \in \mathbb{N}$, if

$$\nu(I_a^{(k)}) = \nu(I_a^{(k)} + \omega T), \quad (5.6)$$

for every $I_a^{(k)} \in \mathcal{I}$. In such a case, we call ω the infrastructural period of the assignment. As it turns out, we are always able to restrict to such repeating patterns without losing any underlying PESP solution.

Theorem 5.1. — *Consider an instance (G, T, ℓ, u, w) of IPESPA with $\eta: \mathcal{A} \rightarrow \mathcal{E} \subseteq 2^E$, and a solution (π, x) together with a valid and *h-conflict-free* infrastructure assignment ν . Then, there exist $\omega \in \mathbb{N}$, with $\omega \leq |E|^{|E|}$, and a valid and *h-conflict-free* infrastructure assignment σ such that σ is ω -periodic.*

Proof. Let us consider \mathcal{I} for the above instance and solution. All activities have a single representative interval in \mathcal{I} whose lower bound is in $[pT, pT + T)$, with $p \in \mathbb{Z}$. This set of representatives, which we denote by \mathcal{F}_p , is finite. Even more so, considering for each $e \in E$ the one interval in \mathcal{F}_p that is ν -assigned to e and

with the minimum lower bound, there are at most $|E|$ such leading intervals. There are at most $|E|^{|E|}$ ways to assign these intervals, and so there exists a $p' \in \mathbb{Z}$ such that $p - |E|^{|E|} \leq p' \leq p + |E|^{|E|}$, and such that $\nu|_{\mathcal{F}_{p'}}$ mirrors $\nu|_{\mathcal{F}_p}$ on all the leading intervals of \mathcal{F}_p . By that, mirroring the whole of $\nu|_{\mathcal{F}_p}$ on $\mathcal{F}_{p'}$ can be done without h -conflict. Then, without loss of generality, we assume that $p < p'$, and set $\omega := p' - p$. By construction $\omega \leq |E|^{|E|}$, and we can construct a valid, h -conflict-free, and ω -periodic infrastructure assignment σ by setting

$$\sigma \left(I_a^{(k)} \right) := \nu \left(I_a^{([k-p]_\omega)} \right). \quad (5.7)$$

□

The theorem ensures that IPESPA can be solved by restricting to periodic assignments, whose maximum period is bounded by the instance. Note that the bound on ω can be significantly improved if all headways are the same, i.e., $h_e = h_{e'}$ for any $e, e' \in E$. In that case, much along the lines of [82, Theorem 3.1], the bound becomes $\omega \leq |E|!$.

5.3.2 Pattern Functions and Conflict-Freeness

We will now construct a finitely described object from which a periodic infrastructure assignment can be derived, and conclude this chapter by showing how to use said object to formulate IPESPA as a mixed-integer program. As in Definition 5.3 we have a PESP instance, an infrastructure map, and a vector of headways. Choosing some maximum infrastructural period $M \in \mathbb{N}_{>0}$, we construct a *pattern function* $\mathcal{H}: \mathcal{A} \rightarrow \mathcal{P}$, that assigns to each arc in \mathcal{A} a *pattern* in

$$\mathcal{P} := \bigcup_{i|M} E^i = \{(e_1, \dots, e_i) \mid e_1, \dots, e_i \in E \text{ and } i \text{ divides } M\}. \quad (5.8)$$

A pattern function is said to be *valid* if every image $\mathcal{H}(a)$ only contains elements of $\eta(a)$. The prescribed pattern is intended to be repeated ad infinitum. A corresponding infrastructure assignment $\nu_{\mathcal{H}}$ is quickly extracted from a pattern function \mathcal{H} , by setting

$$\nu_{\mathcal{H}} \left(I_a^{(k)} \right) := \mathcal{H}(a)_{[k]_{m_a}} \quad \forall a \in \mathcal{A}, \forall k \in \mathbb{Z}, \quad (5.9)$$

where m_a is the length of $\mathcal{H}(a)$, i.e., the number of entries. So constructed, $\nu_{\mathcal{H}}$ is periodic, of period at most M . Note that the choice of M is up to the planner and the model becomes more flexible the more M is divisible.

Example 5.2. — *Let us consider again Example 5.1 in the case where we allow to assign each activity to each platform, i.e., $\eta(a_k) = \{e_1, e_2\}$ for all $k \in \{1, 2, 3\}$. The valid pattern function associated to the infrastructure assignment in Figure 5.1b is given by $\mathcal{H}(a_1) = \mathcal{H}(a_3) = (e_1, e_2)$, $\mathcal{H}(a_2) = (e_2, e_1)$, so that $m_{a_1} = m_{a_2} = m_{a_3} = 2$.*

We can now use pattern functions to formulate linear modulo constraints, that generalize the Q0-constraints as presented in (5.2).

Theorem 5.2. — *Consider a PESP instance (G, T, ℓ, u, w) , an infrastructure map $\eta: \mathcal{A} \rightarrow \mathcal{E} \subseteq 2^E$, headways $h \in \mathbb{R}_{\geq 0}^E$, and some PESP solution (π, x) . Let \mathcal{H} be a pattern function, and denote by m_a the length of the pattern $\mathcal{H}(a)$. Then, the infrastructure assignment $\nu_{\mathcal{H}}$ is h -conflict-free if and only if:*

(a) *For every arc $a \in \mathcal{A}$ and infrastructure element $e \in \mathcal{H}(a)$, we have*

$$x_a + h_e \leq m_a T. \quad (5.10)$$

(b) *For every $a_1, a_2 \in \mathcal{A}$, with $a_1 = (i_1, j_1)$ and $a_2 = (i_2, j_2)$, with images under \mathcal{H} both containing the same infrastructure element e at indices p_1 and p_2 respectively, such that either $a_1 \neq a_2$ or $p_1 \neq p_2$, we have*

$$\begin{aligned} & [\pi_{i_2} + (p_2 + k_2 m_{a_2})T - \pi_{i_1} - (p_1 + k_1 m_{a_1})T]_{mT} \geq x_{a_1} + h_e, \\ & \forall k_1 \in \left\{0, \dots, \frac{m}{m_{a_1}} - 1\right\}, k_2 \in \left\{0, \dots, \frac{m}{m_{a_2}} - 1\right\}, \end{aligned} \quad (5.11)$$

where $m := \text{lcm}(m_{a_1}, m_{a_2})$, and the indexing of the patterns starts at 0.

Proof. (\implies): If (a) is violated, then $\nu_{\mathcal{H}}(I_a^{(p)}) = e = \nu_{\mathcal{H}}(I_a^{(p+m_a)})$, for p the index of e in $\mathcal{H}(a)$. Then we find in \mathcal{I}_e the intervals

$$\begin{aligned} & [\pi_i + pT, \pi_i + pT + x_{ij} + h_e) \text{ and} \\ & [\pi_i + (p + m_a)T, \pi_i + (p + m_a)T + x_{ij} + h_e), \end{aligned} \quad (5.12)$$

which intersect, since $\pi_i + pT + x_{ij} + h_e > \pi_i + pT + m_a T$.

Suppose instead (a) holds, and (b) is violated, for some $k_1 \in \{0, \dots, m/m_{a_1} - 1\}$ and $k_2 \in \{0, \dots, m/m_{a_2} - 1\}$. By construction in (5.9), note that the images under $\nu_{\mathcal{H}}$ of $I_{a_1}^{(p_1+k_1m_{a_1})}$ and $I_{a_2}^{(p_2+k_2m_{a_2})}$ are both e , in fact for any integral k_1 and k_2 . Since $\pi_v \in [0, T)$ for every $v \in V(G)$, we have that

$$[\pi_{i_2} + (p_2 + k_2m_{a_2})T - \pi_{i_1} - (p_1 + k_1m_{a_1})T]_{mT} < x_{a_1} + h_e, \quad (5.13)$$

and by construction the content of the modulo operator is either already in $[0, mT)$, or it is in $[-mT, 0)$. If it is non-negative, we find in \mathcal{I}_e the intervals

$$\begin{aligned} &[\pi_{i_1} + (p_1 + k_1m_{a_1})T, \pi_{i_1} + (p_1 + k_1m_{a_1})T + x_{a_1} + h_e] \text{ and} \\ &[\pi_{i_2} + (p_2 + k_2m_{a_2})T, \pi_{i_2} + (p_2 + k_2m_{a_2})T + x_{a_2} + h_e], \end{aligned} \quad (5.14)$$

and they intersect. If instead the content is negative, then the modulo operator will add mT , and we find in \mathcal{I}_e the intervals

$$\begin{aligned} &[\pi_{i_1} + (p_1 + k_1m_{a_1})T, \pi_{i_1} + (p_1 + k_1m_{a_1})T + x_{a_1} + h_e] \text{ and} \\ &[\pi_{i_2} + (p_2 + k_2m_{a_2} + m)T, \pi_{i_2} + (p_2 + k_2m_{a_2} + m)T + x_{a_2} + h_e], \end{aligned} \quad (5.15)$$

and they intersect.

(\Leftarrow): Suppose now that $\nu_{\mathcal{H}}$ is not h -conflict-free. There is then an element $e \in E$ such that the set \mathcal{I}_e contains two intersecting intervals. Let these be $I_{a_1}^{(s_1)}$ and $I_{a_2}^{(s_2)}$, and without loss of generality let us assume that $\min I_{a_2}^{(s_2)} \in I_{a_1}^{(s_1)}$. We then have that

$$0 \leq \pi_{i_2} + s_2T - \pi_{i_1} - s_1T < x_{a_1} - h_e \leq m_{a_1}T, \quad (5.16)$$

where the last inequality holds if (a) does. Then, since $m_{a_1}T \leq mT$, the $[\cdot]_{mT}$ operator is freely applied, and we have $[\pi_{i_2} + s_2T - \pi_{i_1} - s_1T]_{mT} < x_{a_1} + h_e$. For both intervals, another way to write s_i is as $p_i + k_i m_{a_i} + \beta_i m$, with integral β_i and minimal positive integral k_i , by which we have

$$[\pi_{i_2} + (p_2 + k_2m_{a_2} + \beta_2m)T - \pi_{i_1} - (p_1 + k_1m_{a_1} + \beta_1m)T]_{mT} < x_{a_1} + h_e. \quad (5.17)$$

Then, the two summands β_2mT and β_1mT can be deleted since they do not affect the modulo operation, and we have found a violation of (b). \square

Remark 5.1. — In the IPESP case, i.e., $|\eta(a)| = 1$ for all $a \in \mathcal{A}$, all patterns \mathcal{H}_a have length $m_a = 1$, so that $m = 1$. Theorem 5.2 then states that an infrastructure assignment is h -conflict-free if and only if $x_a + h_e \leq T$ for all $e \in E$ and $a \in \mathcal{A}_e$ and the $Q0$ -constraint (5.2) holds for all pairs of distinct arcs assigned to the same infrastructure element. Indeed, this was our definition of h -conflict-freeness in Section 5.2. In general, the constraints (5.11) can be interpreted as $Q0$ -constraints that implicitly capture a time expansion up to period mT .

5.3.3 A MIP Formulation for IPESPA

We now want to use the platforming period bound of Theorem 5.1 and the inequalities of Theorem 5.2 to extend the PESP mixed-integer program (5.1) to IPESPA. We model a pattern function \mathcal{H} by introducing binary variables $\tau_{a\rho}$ for all possible $a \in \mathcal{A}$ and images $\rho \in \mathcal{P} = \bigcup_{i|M} E^i$, whenever ρ is valid for a , i.e., all entries of ρ are in $\eta(a)$. By the bound expressed in Theorem 5.1 we can choose a finite but sufficiently large M , thereby having finitely many variables $\tau_{a\rho}$. Exactly one pattern has to be activated for each arc, which we express by

$$\sum_{\rho \in \mathcal{P}} \tau_{a\rho} = 1, \quad \forall a \in \mathcal{A}. \quad (5.18)$$

We include the inequalities (5.10), but only activate them if relevant, by having

$$x_a + \max_{e \in \rho} h_e \leq m_a T + (1 - \tau_{a\rho})B, \quad (5.19)$$

for every variable $\tau_{a\rho}$. Here $\max_{e \in \rho} h_e$ and m_a are scalars, determined by the pattern ρ , and $B := \max_{a \in \mathcal{A}} u_a + \max_{e \in E} h_e$ is a scalar globally determined by the instance itself. This way, if $\mathcal{H}(a) = \rho$, then (5.19) is effective, but otherwise it is trivially satisfied.

To linearize the modulo operation in (5.11) we also introduce binary variables $s_{\rho_1 \rho_2}$. These are analogous to the periodic offsets p_{ij} (5.1e) and can indeed be restricted to $\{0, 1\}$, as in this case the modulo operator is applied to a number

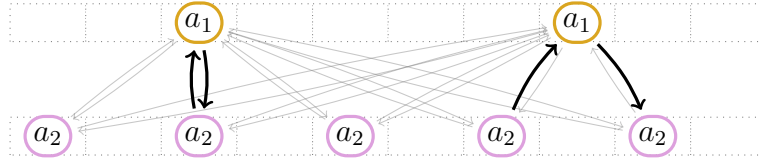


Figure 5.2: Two time expansions over 10 periods, with occurrences of two activities, a_1 and a_2 , with different pattern lengths. Each arc symbolizes one constraint like (5.20). Only those highlighted in black are needed in case there were upper bounds $u_{a_1}, u_{a_2} \leq T$.

in $[-mT, mT]$. Activating the constraint if and only if both patterns ρ_1 and ρ_2 are selected, we have

$$\pi_{i_2} + (p_2 + k_2 m_{a_2})T - \pi_{i_1} - (p_1 + k_1 m_{a_1})T + mT s_{\rho_1 \rho_2} \geq x_{a_1} + h_e - (2 - \tau_{a_1 \rho_1} - \tau_{a_2 \rho_2})B, \quad (5.20)$$

for every pair of arcs $a_1, a_2 \in \mathcal{A}$, respectively with \mathcal{H} -images ρ_1, ρ_2 , of lengths m_{a_1}, m_{a_2} , both containing $e \in E$ at indices p_1, p_2 (0-indexed), for every $k_1 \in \{0, \dots, m/m_{a_1} - 1\}$ and $k_2 \in \{0, \dots, m/m_{a_2} - 1\}$, and where $m = \text{lcm}(m_{a_1}, m_{a_2})$. Note that p_i, m_{a_i}, m , and h_e , here are all scalars, determined by ρ_1 and ρ_2 .

Example 5.3. — For an illustration of which constraints (5.20) are applied, we refer to Figure 5.2. There we have two activities a_1 and a_2 , assignable to the same infrastructure element, with $p_1 = 2, m_1 = 5$, and $p_2 = 0, m_2 = 2$. Each arc symbolizes one constraint like (5.20), of which there is two per pairing, i.e., 20 in total. However, in many cases, if $p_2 + k_2 m_{a_2} - p_1 - k_1 m_{a_1} > \lceil (u_{a_1} + h_e)/T \rceil$, then (5.11) is always satisfied, and we can exclude it a priori. This means that, depending on the PESP upper bounds, a significant drop in the number of constraints is possible. The bold arcs in fig. 5.2 are the 4 constraints that would be kept if we had, for instance, $u_{a_1}, u_{a_2} \leq T$.

Including in a PESP mixed-integer program such as (5.1) these two types of binary variables $\tau_{a\rho}$ and $s_{\rho_1 \rho_2}$, together with (5.18), (5.19), and (5.20), yields a

mixed-integer program formulation for IPESPA. In (5.21) we can see it in full.

$$\min \sum_{a \in A(G)} w_a x_a \quad (5.21a)$$

$$\text{s.t.} \quad (\pi, x) \text{ solves PESP on } G, \quad (5.21b)$$

$$\sum_{\rho \in \mathcal{P}} \tau_{a\rho} = 1 \quad \forall a \in \mathcal{A}, \quad (5.21c)$$

$$x_a + \max_{e \in \rho} h_e \leq m_a T + (1 - \tau_{a\rho}) B \quad \forall a \in \mathcal{A} \text{ and} \\ \forall \text{ valid } \rho \in \mathcal{P}, \quad (5.21d)$$

$$\begin{pmatrix} \pi_{i_2} + (p_2 + k_2 m_{a_2}) T \\ -\pi_{i_1} - (p_1 + k_1 m_{a_1}) T \\ +mT s_{\rho_1 \rho_2} \end{pmatrix} \geq \begin{pmatrix} x_{a_1} + h_e \\ -(2 - \tau_{a_1 \rho_1} - \tau_{a_2 \rho_2}) B \end{pmatrix} \quad \forall a_1, a_2 \text{ s.t. } (\star), \quad (5.21e)$$

$$\tau_{a\rho} \in \{0, 1\} \quad \forall a \in \mathcal{A} \text{ and} \\ \forall \text{ valid } \rho \in \mathcal{P}, \quad (5.21f)$$

$$s_{\rho_1 \rho_2} \in \{0, 1\} \quad \forall \rho_1, \rho_2 \in \mathcal{P}, \quad (5.21g)$$

where by (\star) we mean the conditions of (5.20).

This formulation allows for a great degree of flexibility, giving practitioners a direct handle on the maximum infrastructural period M . In fact, although fixing M to the bound proven in theorem 5.1 ensures that no PESP solution is excluded, it is entirely possible that in a practical setting one would want to limit it further, so as to bound the complexity of the infrastructural assignment. To that same purpose, the formulation also allows to forcibly forbid individual patterns if desired.

5.4 Partitionable Infrastructure Maps

In practice, the infrastructure map $\eta: \mathcal{A} \rightarrow \mathcal{E}$ is oftentimes not completely flexible, as shown in fig. 5.3a, but instead comes with additional structure. Of particular interest is the case illustrated in fig. 5.3b, where \mathcal{E} is a partition of all infrastructure elements, i.e., the infrastructure elements are all grouped and every activity can be freely assigned to all elements in one of such groups. An omnipresent example is a

station with two platforms that serves lines in two directions, with the lines in each direction having a dedicated platform. It turns out that when η is *partitionable*, the IPESPA boils down to a rather compact form.

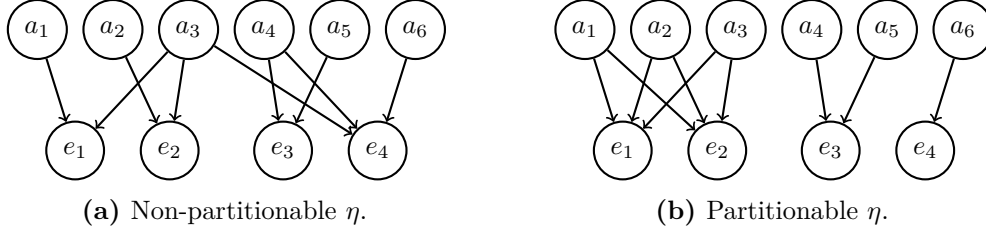


Figure 5.3: Two infrastructure maps for an instance with six activities and four infrastructure elements.

Formally, we define the following variant of the IPESPA.

Definition 5.4. — *Let (G, T, ℓ, u, w) be a PESP instance and $\eta: \mathcal{A} \rightarrow \mathcal{E}$ an infrastructure map, where \mathcal{E} is a partition of E . The Infrastructure-Aware PESP with Capacities (IPESPC) is to find a solution (π, x) to PESP on (G, T, ℓ, u, w) , together with a valid and conflict-free platform assignment ν , such that the solution is optimal, or to decide that no such solution exists.*

Alternatively, IPESPC is equivalent to IPESP with the additional feature that every $e \in E$ has now a capacity $k_e \in \mathbb{N}$. In other words, $e \in E$ no longer corresponds to a single infrastructure element, but to a group of k_e equivalent elements. For ease of exposition, we stick to this perspective going forward.

We use a matching-based approach to solve IPESPC. To this end, we expand G with auxiliary arcs between activities that can use the same group of infrastructure elements. Formally, let G' be the graph arising from G by adding for each $e \in E$ and $a_1 = (i_1, j_1), a_2 = (i_2, j_2) \in \mathcal{A}_e$ a new arc α from j_1 to i_2 . We refer to α as the auxiliary arc from a_1 to a_2 , and much like the headway arcs a^{\downarrow} used in [3] and in the Q4 butterfly constraints (5.3), we set $\ell_\alpha := h_e, u_\alpha := T - h_e, w_\alpha := 0$ if $a_1 \neq a_2$, and $\ell_\alpha := h_e, u_\alpha := T, w_\alpha := 0$ in the case $a_1 = a_2$. Let \mathcal{A}' denote the set of all auxiliary arcs, let \mathcal{A}'_e denote all auxiliary arcs associated to e , and let G'_e be the subgraph of G' on the arcs in \mathcal{A}'_e . For $S \subseteq V(G) \times V(G)$, we will use the notation $G[S]$ for the graph $(V(G), A(G) \cup S)$, so that, e.g., $G' = G[\mathcal{A}']$. The

following theorem compactly characterizes when a PESP solution admits a feasible infrastructure assignment in the context of IPESPC:

Theorem 5.3. — *Consider a PESP instance (G, T, ℓ, u, w) , the expanded graph G' , a partitionable infrastructure map $\eta: \mathcal{A} \rightarrow \mathcal{E}$, headways $h \in \mathbb{R}_{\geq 0}^E$, capacities $k \in \mathbb{N}^E$, and some PESP solution (π, x) on G . Then, there exists a valid and h -conflict-free infrastructure assignment ν if and only if for each $e \in E$ there exists a perfect matching $\mathcal{M}'_e \subseteq \mathcal{A}'_e$ of G'_e and (π, x) can be extended to a PESP solution on $G[\mathcal{M}'_e]$ such that*

$$\sum_{a \in \mathcal{A}_e} x_a + \sum_{a \in \mathcal{M}'_e} x_a \leq k_e T. \quad (5.22)$$

Proof. First, assume that there exist perfect matchings \mathcal{M}'_e of G'_e satisfying (5.22) for all $e \in E$. The matching \mathcal{M}'_e together with the arcs \mathcal{A}_e forms a set of disjoint directed cycles, where every cycle consists of arcs that alternately belong to \mathcal{A}_e and \mathcal{A}'_e .

Let then $C_1^e, C_2^e, \dots, C_{m_e}^e$ denote the cycles corresponding to $e \in E$, and define $p_j^e := \frac{1}{T} \sum_{a \in C_j^e} x_a$ for $j = 1, \dots, m_e$. Because the timetable (π, x) is feasible on $G[\mathcal{M}'_e]$, p_j^e is integer by the cycle periodicity property [42, Lemma 6.39]. Since it holds that

$$\sum_{j=1}^{m_e} p_j^e = \frac{1}{T} \sum_{j=1}^{m_e} \sum_{a \in C_j^e} x_a = \frac{1}{T} \left(\sum_{a \in \mathcal{A}_e} x_a + \sum_{a \in \mathcal{M}'_e} x_a \right) \leq k_e, \quad (5.23)$$

it suffices to show that just the activities appearing in C_j^e can be assigned on a group of capacity p_j^e . The total tension along the cycle is $p_j^e T$, so in a $p_j^e T$ -periodic schedule it is straightforward to fit all the activities in $C_j^e \cap \mathcal{A}$, simply following the order in which they appear through the cycle and with timestamps agreeing with π modulo T . Then, having an available capacity of p_j^e , that schedule can be repeated over each of the p_j^e equivalent elements, each time shifted forward by T . This construction implies adherence to the T -periodic timetable π , and a $p_j^e T$ -periodic infrastructure assignment, valid since each $\eta(C_j^e \cap \mathcal{A}) = e$, and h -conflict-free by the bounds on the auxiliary arcs.

Suppose the contrary instead, that no perfect matching in \mathcal{A}'_e respects (5.22). Notice how, given that π is fixed and the headway h_e is the same on all elements in the group e , we can without loss of generality lengthen all activities in \mathcal{A}_e by h_e and then assume that the new minimum headway is 0 instead. Now, let \mathcal{M}^* be a minimum tension perfect matching in G'_e . We then have that for some $e \in E$ there is an integer $K_e > k_e$ such that

$$\sum_{a \in \mathcal{A}_e} x_a + \sum_{a \in \mathcal{M}^*} x_a = K_e T > k_e T. \tag{5.24}$$

Now, for $t \in [0, T)$ and $S \subseteq \mathcal{A}_e \cup \mathcal{A}'_e$, let the *inventory function* $I_e(t, S)$ denote the number of h -overlapping activities in S at time t . By (5.24) we have that $I_e(t, \mathcal{A}_e \cup \mathcal{M}^*) = K_e$ for all $t \in [0, T)$. Moreover, since \mathcal{M}^* is a minimum tension perfect matching, then there is a $t^* \in [0, T)$ such that $I_e(t^*, \mathcal{M}^*) = 0$. That is because if $\mathcal{M}^* = \{(e_1, f_1), \dots, (e_m, f_m)\}$ was h -overlapping everywhere instead, then without loss of generality we can assume that the timestamps would be

$$\pi_{e_1} < \pi_{f_m} < \pi_{e_2} < \pi_{f_1} < \dots < \pi_{e_m} < \pi_{f_{m-1}}, \tag{5.25}$$

where a shorter matching is immediately apparent, negating the minimality of \mathcal{M}^* . See [35, Lemma 3] for further details. By the above, we then have that $I_e(t^*, \mathcal{A}_e) = I_e(t^*, \mathcal{A}_e \cup \mathcal{M}^*) - I_e(t^*, \mathcal{M}^*) = K_e > k_e$, implying there are more simultaneous activities than there is infrastructure capacity to host them. In particular, there is no h -conflict-free infrastructure assignment. \square



(a) Matching corresponding to Figure 5.1a. (b) Matching corresponding to Figure 5.1b.

Figure 5.4: The two matchings corresponding to Example 5.1.

Example 5.4. — We illustrate on the basis of Example 5.1 how matchings relate to infrastructure assignments in Figure 5.4. In the IPESPA situation of Figure 5.1a, we have three infrastructure elements e_1, e_2, e_3 of capacity one each. For each element e_k , the matching in Figure 5.4a creates a directed cycle of tension $1 \cdot T = 30$ containing the dwelling activity $a_k = (i_k, j_k)$ and the auxiliary arc (j_k, i_k) . Figure 5.1b is an IPESPC situation with one infrastructure element e . Here, the matching in Figure 5.4b induces a directed cycle of tension $2 \cdot T = 60$, we hence use a capacity of two. The cycle encodes that each platform is used by a_1, a_3, a_2 in this cyclic order.

Theorem 5.3 allows formulating IPESPC very compactly compared to IPESPA. Introducing matching variables as binary decision variables y_a for all $a \in \mathcal{A}'$, we have the following MIP:

$$\min \sum_{a \in A(G)} w_a x_a \quad (5.26a)$$

$$\text{s.t.} \quad (\pi, x) \text{ solves PESP on } G, \quad (5.26b)$$

$$\sum_{a \in \delta_{G'_e}^-(i)} y_a = 1 \quad \forall e \in E, \forall (i, j) \in \mathcal{A}_e, \quad (5.26c)$$

$$\sum_{a \in \delta_{G'_e}^+(j)} y_a = 1 \quad \forall e \in E, \forall (i, j) \in \mathcal{A}_e, \quad (5.26d)$$

$$\sum_{a \in \mathcal{A}_e} x_a + \sum_{a \in \mathcal{A}'_e} x_a y_a \leq k_e T \quad \forall e \in E, \quad (5.26e)$$

$$\pi_j - \pi_i + T p_a = x_a \quad \forall a = (i, j) \in \mathcal{A}', \quad (5.26f)$$

$$\ell_a y_a \leq x_a \leq u_a y_a + (T - 1)(1 - y_a) \quad \forall a \in \mathcal{A}', \quad (5.26g)$$

$$y_a \in \{0, 1\} \quad \forall a \in \mathcal{A}'. \quad (5.26h)$$

Constraints (5.26c) and (5.26d) define a unique predecessor and successor for each activity by requiring that y corresponds to a perfect matching \mathcal{M}'_e in \mathcal{A}'_e for each $e \in E$. Constraints (5.26e) ensure that the total time of the activities scheduled on a platform group and the selected auxiliary activities is at most the total available time on that platform group. These constraints can be linearized by introducing for each $a \in \mathcal{A}'$ a real variable z_a , with bounds $0 \leq z_a \leq u_a$, and constrained as $x_a - (1 - y_a)u_a \leq z_a \leq x_a$, so that it is equal to the product $x_a y_a$. The constraints

(5.26f) tie the tensions x_a on the auxiliary activities $a \in \mathcal{A}'$ to the timetable π . Finally, (5.26g) ensures that those tensions adhere to their bounds when they are part of the selected matching, and impose no restrictions otherwise.

There is a clear resemblance between (5.26) and the formulation [35] proposes for jointly optimizing a periodic timetable and vehicle circulation. This is no coincidence: as long as the corresponding mapping from activities to resources is a partition, any resource schedule associated to a periodic timetable can be described by a matching. In [35] the resources are vehicles, whereas in the present paper infrastructure elements, e.g., platforms. It immediately follows that the results established in [35] carry over to our setting. Most notably, given a feasible timetable, a greedy algorithm can actually be used to find an infrastructure assignment with the minimum number of required infrastructure elements.

For groups consisting of a single element, i.e., $k_e = 1$, the matching formulation can be enhanced using the surprising insight that in this case it is not necessary to compute the matching explicitly. We have the following theorem:

Theorem 5.4. — *Suppose $k_e = 1$ and let (π, x) be a PESP solution on G . Then \mathcal{A}_e is h -conflict-free if and only if (π, x) extends to a PESP solution on G'_e such that*

$$\sum_{a \in \mathcal{A}_e} x_a + \left(\frac{1}{|\mathcal{A}_e|} \sum_{a \in \mathcal{A}'_e} x_a \right) = \frac{|\mathcal{A}_e| + 1}{2} T. \quad (5.27)$$

Proof. Suppose that \mathcal{A}_e is h -conflict-free. This means that for any $a_s = (i_s, j_s), a_t = (i_t, j_t) \in \mathcal{A}_e$ with $a_s \neq a_t$ the Q4-constraints (5.3) must hold, namely

$$x_{i_s, j_s} + x'_{j_s, i_t} + x_{i_t, j_t} + x'_{j_t, i_s} = T, \quad (5.28)$$

where we use the notation $x'_a := [\pi_j - \pi_i - \ell_a]_t + \ell_a$ to indicate tensions on an auxiliary arc $a = (i, j)$.

Let $q(a_s, a_t)$ denote the butterfly-shaped 4-cycle given by $(i_s, j_s, i_t, j_t, i_s)$. There are $\sum_{i=1}^{|\mathcal{A}_e|-1} i = |\mathcal{A}_e|(|\mathcal{A}_e| - 1)/2$ many such butterfly cycles, each auxiliary arc $(j_s, i_t) \in \mathcal{A}'_e$ with $s \neq t$ is in exactly one such cycle, while each arc $(i_s, j_s) \in \mathcal{A}_e$ is in

$|\mathcal{A}_e| - 1$ many of them. Moreover, there are exactly $|\mathcal{A}_e|$ auxiliary arcs of the form (j_s, i_s) , for which holds that

$$h_e \leq x_{i_s, j_s} + x'_{j_s, i_s} \leq 2T - h_e, \quad (5.29)$$

since $h_e \leq x'_{j_s, i_s} \leq T$ and h -conflict freeness implies $x_{i_s, j_s} \leq T - h_e$. Due to the cycle periodicity property,

$$x_{i_s, j_s} + x'_{j_s, i_s} = T, \quad (5.30)$$

unless $h_e = 0$, but then we can subtract T from $x'_{j_s, i_s} = T$ and maintain the feasibility of (π, x) on G'_e .

Summing up over all butterfly constraints (5.28) for each pair of activities and all the self-cycles (5.30) of each activity in \mathcal{A}_e , we obtain

$$|\mathcal{A}_e| \sum_{s=1}^{|\mathcal{A}_e|} x_{i_s, j_s} + \sum_{a \in \mathcal{A}'_e} x'_a = \left(\frac{|\mathcal{A}_e|(|\mathcal{A}_e| - 1)}{2} + |\mathcal{A}_e| \right) T = \frac{|\mathcal{A}_e|(|\mathcal{A}_e| + 1)}{2} T. \quad (5.31)$$

For the other direction, suppose that \mathcal{A}'_e is not h -conflict free, but (π, x) extends to PESP solution on G'_e . Then one of (5.28) or (5.30) must be violated. Let $q(a_s, a_t)$ be a butterfly cycle with $a_s = (i_s, j_s)$ and $a_t = (i_t, j_t)$. Then

$$x_{i_s, j_s} + x'_{j_s, i_t} + x_{i_t, j_t} + x'_{j_t, i_s} \geq \ell_{i_s, j_s} + \ell_{i_t, j_t} + 2h_e > 0 \quad (5.32)$$

since we assumed that at least lower bounds or minimum headway times are positive. Due to the cycle periodicity property of periodic timetables,

$$x_{i_s, j_s} + x'_{j_s, i_t} + x_{i_t, j_t} + x'_{j_t, i_s} \geq T. \quad (5.33)$$

Moreover, considering the cycles comprised of $(i_s, j_s) \in \mathcal{A}_e$ and the auxiliary arc $(j_s, i_s) \in \mathcal{A}'_e$, we have

$$x_{i_s, j_s} + x'_{j_s, i_s} \geq \ell_{i_s, j_s} + h_e > 0, \quad (5.34)$$

so that

$$x_{i_s, j_s} + x'_{j_s, i_s} \geq T. \quad (5.35)$$

Therefore, if one of (5.28) or (5.30) is violated, we must have a strict inequality in (5.33) or (5.35). Taking the sum,

$$|\mathcal{A}_e| \sum_{s=1}^{|\mathcal{A}_e|} x_{i_s, j_s} + \sum_{a \in \mathcal{A}'_e} x'_a > \left(\frac{|\mathcal{A}_e|(|\mathcal{A}_e| - 1)}{2} + |\mathcal{A}_e| \right) T = \frac{|\mathcal{A}_e|(|\mathcal{A}_e| + 1)}{2} T, \quad (5.36)$$

so (5.27) cannot hold. \square

A direct implication of Theorem 5.4 is a new formulation for IPESP, provided that all infrastructure elements have unit capacities:

$$\min \sum_{a \in A(G)} w_a x_a \quad (5.37a)$$

$$\text{s.t.} \quad (\pi, x) \text{ solves PESP on } G', \quad (5.37b)$$

$$\sum_{a \in \mathcal{A}_e} x_a + \frac{1}{|\mathcal{A}_e|} \sum_{a \in \mathcal{A}'_e} x_a = \frac{|\mathcal{A}_e| + 1}{2} T \quad \forall e \in E. \quad (5.37c)$$

In contrast to the original formulation proposed in [3] and to the matching approach (5.26), formulation (5.37) introduces neither additional integer variables nor quadratic terms.

5.5 Experiments

In the sequel, we will evaluate different formulations for IPESP and IPESPC on a set of realistic instances. We omit the IPESPA model, as for our datasets there hardly is added value compared to IPESPC. We use Gurobi 11 as a MIP solver on an Intel Xeon E3-1270 3.80 GHz CPU with 32 GB RAM.

5.5.1 Instances

We evaluate our models on instances we constructed based on publicly available timetable information, platform usage, and track data. Additionally, some track information was provided to us by DB InfraGO AG. The instances are:

- **S-Bahn**, the full network of S-Bahn Berlin, a suburban commuter rail network with 16 lines. On several sections, there are as much as 7 trains per track and direction within the period time of 20 minutes. Our IPESP instance is based

on the annual timetable, assuming fixed driving times, but flexible dwelling and turnaround times. However, there are several places in the network where multiple platforms are available, and this builds our corresponding IPESPC instance.

- **Tram**, the full tram network of Berlin, comprising 22 lines operated with a period time of 20 minutes, with frequencies ranging between 1 and 6. The difficulty here does not lie in associating driving and dwelling times, which are fixed, but in fulfilling synchronization constraints and deciding infrastructure assignments at the terminal stations: The turnaround times are flexible and capacities in the turning loops are scarce. This is inherently an IPESPC instance, that, in fact, cannot be transformed into a feasible IPESP instance, since all T -periodic assignments are infeasible.
- **Corridor**, the central longitudinal railway corridor of regional and long-distance trains in Berlin, as well as a subsection of it, from Ostkreuz to Friedrichstraße, which we denote as **ShortCorridor**. Many stations have multiple platforms, and the trains have different stopping patterns. The period time is 60 minutes, driving, dwelling, and turnaround times are all variable. This, too, is inherently an IPESPC instance, from which we created an IPESP instance based on the annual timetable.

We use only a simple objective function for the timetabling part: Driving and dwelling activities are weighted by 2, turnarounds by 1, and all other arcs by 0. Additionally, note that we do not include any transfer arcs, in part because we have no data available regarding the flow of passengers, and in part because the scope of this work rather focuses on operational capabilities instead.

5.5.2 IPESP Experiments

For the unit capacity case of IPESP, we have now several formulations at hand: The Q4 butterfly constraints (5.3), the matching model (5.26), and the special formulation (5.37). We test these formulations and their combinations on the

S-Bahn and **Corridor** instance, with a wall time limit of one hour. We further include versions where the matching variables y are relaxed to be continuous. Our results are collected in Table 5.1.

On **S-Bahn**, not all formulations found a solution, but those that did also managed to prove optimality within the time limit. All such formulations contain the Q4 butterfly constraints (5.3), and none of them use binary matching variables. The combination of Q4 with the special sum constraint (5.37) worked best, followed by pure Q4. When initialized with the optimal solution as MIP start, almost all formulations proved its optimality within the time frame of one hour, or managed a very thin optimality gap.

On **Corridor**, no formulation found solutions, and no attempt at providing initial partial solutions was successful. On **ShortCorridor** all formulations quickly found a primal solution within seconds, and an optimal solution within minutes, but none managed to prove that optimality within one hour. Only when starting **ShortCorridor** with the best solution found so far, a proof to optimality was reached, and only by the formulation using (5.26), i.e., matching with binary variables, together with the special constraints (5.37c). This was also the fastest formulation to reach optimality to begin with.

5.5.3 IPESPC Experiments

For the instances with capacities larger than one we used formulation (5.26). To aid the solution process, all infrastructure that still had capacity one has been modelled using Q4 constraints, and used the matching variables where necessary, i.e., for all larger infrastructure. We tested this formulation on the **S-Bahn**, the **Tram**, and the **Corridor** instance, with a time limit of four hours.

On **S-Bahn** no primal solution was found, but we were able to warm start the models with solutions found in the corresponding IPESP tests instead. In that case, the more flexible IPESPC within seconds improved the optimal IPESP solution (albeit by a measly 1.1%), and within approximately 4 more minutes reached the

S-Bahn	<i>s</i> to primal	<i>s</i> to optimal	<i>s</i> to proof	<i>s</i> to proof (warm)
Q4	64	227	227	95
Q4+M	–	–	–	1221
Q4+Mc	338	1672	1672	140
Q4+S	40	156	156	150
Q4+M+S	–	–	–	– (.58%)
Q4+Mc+S	172	738	738	766
M	–	–	–	– (.61%)
S	–	–	–	595
M+S	–	–	–	2952
Mc+S	–	–	–	1402
ShortCorridor	<i>s</i> to primal	<i>s</i> to optimal	<i>s</i> to proof	<i>s</i> to proof (warm)
Q4	0	110	–	–
Q4+M	5	87	–	–
Q4+Mc	0	82	–	–
Q4+S	2	48	–	–
Q4+M+S	1	15	–	–
Q4+Mc+S	0	23	–	–
M	0	44	–	–
S	1	576	–	–
M+S	4	11	–	3268
Mc+S	6	50	–	–

Table 5.1: Timed results for IPESP tests on **S-Bahn** and **Corridor**, expressed in seconds. Tests denoted with Q4 use Q4-constraints as in (5.3). Tests denoted with M use matching constraints as in (5.26), with capacity set to 1. Tests denoted with Mc use the same constraints as M, but with relaxed continuous variables instead. Tests denoted with S use constraints as in (5.37). The first column details the time to the first primal solution that was found, the second column the time to the optimal objective value (3058 for **S-Bahn** and 10 for **Corridor**), and the third column the time to fully close the optimality gap. The last column details the time needed to prove optimality when given the optimal solution from the beginning. The time limit was 1 hour per configuration.

final primal value. In the course of the first hour, Gurobi managed to reduce the optimality gap to 0.2%, where it remained until the time limit.

On **Tram** proven optimality was reached within 10 seconds. Notably, however quick to solve this instance was, it is infeasible to formulate with simple IPESP. Only using the higher capacities enabled by IPESPC it was at all possible to generate a feasible solution.

On **Corridor**, again, no solution was found, but providing a partial starting solution on just the section of **ShortCorridor** was enough to be completed to a

full solution for the whole network, which then was improved to proven optimality in only 2 minutes and 25 seconds. On `ShortCorridor` the first primal was found in under 10 minutes and was directly proven to be optimal. Notably, its objective value was better than the IPESP case, now reaching 0 slack. Starting the same test with a solution from the IPESP case reached proven optimality in 3 seconds.

5.6 Conclusion

This work extends the Infrastructure-Aware PESP (IPESP) framework. One of our new problem formulations, Infrastructure-Aware PESP with Assignment (IPESPA), does so by integrating the choice of the infrastructure assignment within IPESP, allowing for more flexible use of the available infrastructure. In fact, this flexibility can lead to higher efficiency, as well as improved timetables. Although extremely general in its assumptions, we prove that IPESPA can be formulated as a mixed integer linear program (5.21).

Moving on to a more restricted, but highly realistic scenario, we consider the case when infrastructure elements can be effectively considered as equivalent, and formulate this special version of IPESPA as well, namely Infrastructure-Aware PESP with Capacities (IPESPC). In this case the assignment structure is of note, since it can be seen as a matching problem on a complete bipartite graph, connecting the ends of activities to the starts of the next ones. This gives us not only a compact mixed integer linear program formulation (5.26), but also novel formulations for IPESP itself, seen as a case of IPESPC with only unit capacities.

Finally, on the practical side, we tested the new matching-based IPESP formulations, as well as the IPESPC formulation. On the unit capacity side, our tests went through various combinations of approaches, and although caution is advised when drawing conclusions, it seems that on the `S-Bahn` instance the Q4-based formulations fared better, whereas on `ShortCorridor`, which has a higher density of larger infrastructure elements to deal with, matching-based formulations had more success. With instances of larger capacity, instead, our tests show that our modelling approach can be of interest in real-world scenarios.

For future work, on the theoretical side we suggest proving tighter bounds on the maximum platforming period, as well as trying to generalize the Q4-constraints much like we here generalized the Q0-constraints. On the practical side, we suggest an iterative approach that uses a separate matching solver to concurrently feed the main model with partial solutions, and to develop heuristic approaches to quickly generate initial solutions.

Authors' Contributions

Conceptualization, R.N.v.L. and E.B.; methodology, E.B. and R.N.v.L.; software, E.B.; validation, E.B., N.L., and R.N.v.L.; formal analysis, R.N.v.L., E.B., and B.M.; investigation, E.B., R.N.v.L., B.M., and N.L.; ; resources, N.L., B.M., R.N.v.L.; data curation, N.L. and E.B.; writing–original draft preparation, E.B. and R.N.v.L.; writing–review and editing, N.L., E.B., and B.M.; visualization, N.L. and E.B.; supervision, N.L.; project administration, E.B.; funding acquisition, N.L. and E.B..

All authors have read and agreed to the published version of the manuscript.

CHAPTER **6**

Conclusion

For more than three decades, the Periodic Event Scheduling Problem has been the go-to choice for periodic timetabling. Over this time, the theoretical understanding of the problem has grown increasingly capable of capturing its most essential core elements. Meanwhile, its practice has graduated from the personal and instance-specific wisdom of experienced planners, to more objective solution methods, with better speed and scalability. This work has presented clear advances in both directions.

By uncovering and studying the geometry of periodic timetables, a topic previously overlooked, not only we found particularly well-behaved polytopal objects, but their relative arrangement proved to be of interest too. This eventually led us to designing tropical neighbourhood search, a heuristic now highly relevant in practice. Our theoretical study did not end at that either, and we turned our attention to the well-known cycle offset auxiliary variables, again detailing their geometric structure, and again finding a valuable combinatorial structure behind it. With it, we also proved new tight bounds on the width of cycle bases, a parameter important for branch-and-bound trees in MIP-based PESP solving.

Wanting for a way to find good PESP solutions in concert with feasible infrastructure use, especially for realistic instances, we developed Infrastructure-Aware PESP, a novel integrated problem formulation. Instead of simply testing the performance of the naïf mixed-integer program, we refined the approach using observations about the cyclic order of critical activities, achieving faster solution times. Then, wishing to also make the infrastructure use as efficient as possible, we broadened our scope and conceived flexible infrastructure assignments. These proved capable of not only dealing with real-world instances, but also of automatically resolving infrastructural planning issues that would otherwise only be tended to by hand. Our models offer a versatile approach that aligns closely with the needs of planners.

This work serves as an example of how powerful the careful study of geometric and combinatorial structures can be, shedding light on several new aspects of periodic event scheduling and its extensions.

To conclude, a look at the horizon is now in order.

Enhanced tns heuristic A first idea is to conduct a pre-processing step to each call of `tns`, focusing on the *exploreList* (cf. Section 3.3). By pre-computing exactly which arcs (i, j) in $(\overline{G}, \kappa(p))$ are a shortest path from i to j , the exploration loop can then be limited to those arcs exclusively. This is because only such arcs can be in shortest paths at all, and only arcs that are in shortest paths can be facet-defining for the polytrope $\mathbf{R}(p)$, and neighbouring polytropes $\mathbf{R}(p')$ can only be non-empty if the support of $p - p'$ is such an arc. This pre-processing step can be done in a multitude of ways. Our preliminary trials using m calls of an ad hoc Dijkstra algorithm, have shown a notable speed-up for full-neighbourhood explorations on PESPlib instances, up to more than a factor 2 [83].

Another option to bring `tns` further would be to keep track of the explored polytropes, essentially drawing a map of the examined neighbourhoods and how they connect, together with well-designed branching and backtracking strategies, to systematically escape local optima.

Novel zns heuristic Similarly to what has been done to develop and enhance `tns`, it does not seem misguided to also consider zonotopal tiles in the cycle offset zonotope (cf. Section 2.4), and devise an exploration strategy there too, which we here refer to as *zonotopal neighbourhood search*, or `zns`. The natural idea would be to consider two tiles to be neighbours when they share a facet, and to then explore tile to tile, of course looking for their integer points.

Per our results, cells in a fine zonotopal tiling are combinatorially identified by spanning structures (cf. Lemma 2.28), and tiles by spanning tree structures. Specifically, the generators of each cell are the co-structure arcs, and the translation vector positioning the cells is given by the labelling of the arcs in the structure. Now, moving from a tile T_1 to a neighbouring T_2 , via a facet, amounts to excluding the generator of T_1 that does not generate the common facet, and then including the missing generator of T_2 , while correctly keeping track of the labels. Combinatorially, meaning at the spanning structures level, that would mean including the right

co-tree arc in the structure of T_1 , so as to reach the structure of the common facet, and then exclude the right arc to reach the structure of T_2 .

As described, this procedure sounds suspiciously similar to the modulo network simplex (**mns**), swapping arcs in and out of spanning tree structures, but thankfully, there are two key differences that separate **mns** and **zns**. First of all, **zns** keeps track of the translation vectors of the cells, while also only allowing moves consistent with the geometry of the tiling. On the contrary, **mns** tries swapping arcs much more indiscriminately. Secondly, while combinatorially the similarity is striking, the object of the search is different, since **mns** looks for spanning tree structures, whereas **zns** looks for integer points, corresponding to entire polytopes, i.e., sets of solutions. This brings us to yet another question, that of determining the difference between **zns** and **tns**. Since moving over integer points in the cycle offset zonotope is equivalent to moving over polytopes in the torus, it is highly relevant to understand if and how the neighbouring relations in Z differ from those in \mathcal{T} . These questions, together with clarifying the effects of choosing different cycle bases, strongly motivate the necessity of future inquiries.

Further problem integration To suggest one last interesting and ambitious idea, we focus specifically on [84]. See [85, Chapter 13] for a survey of the tools used. In the context of railway cargo transport at the continental scale, the authors of [84] preoccupied themselves with designing a general-purpose modelling strategy for the block-planning and fleet management problems, satisfying shipper and consignee inputs, while abiding to a given weekly periodic timetable and respecting specified infrastructure use. Given the high degree of precision needed, on the side of both the transported goods and the rolling stock, their framework is extremely fine, taking into account how individual containers and locomotives flow through the network, and how they all can be stacked, chained, and repurposed at certain special terminals. To model this they use a 4-layer approach, with timetable requirements coming into play at the top layer, and commodity inputs and outputs being dealt with at the bottom layer. Formulating the model as a MIP is possible, but only by

careful and sequential variable fixing are feasible solutions to the full continental instance found at all. In this setting, an attempt could be made to include timetable optimisation into their design, so as to optimise the inputs at the top layer instead of relying on pre-computed and suboptimal decisions.

As we well know, timetabling is not the only problem in the planning and management of transportation [13]. Advancing integrated approaches, either in modelling theory or in computational feasibility, is the obvious path towards the holy grail of completely integrated optimisation and resounding practical success.

Bibliography

- [1] E. Bortoletto, N. Lindner, and B. Masing, “The Tropical and Zonotopal Geometry of Periodic Timetables,” *Discrete & Computational Geometry*, 2024. DOI: 10.1007/s00454-024-00686-2.
- [2] E. Bortoletto, N. Lindner, and B. Masing, “Tropical Neighbourhood Search: A New Heuristic for Periodic Timetabling,” in *22nd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2022)*, vol. 106, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 3:1–3:19. DOI: 10.4230/OASIcs.ATMOS.2022.3.
- [3] E. Bortoletto, N. Lindner, and B. Masing, “Periodic Timetabling with Cyclic Order Constraints,” in *23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023)*, vol. 115, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 7:1–7:18. DOI: 10.4230/OASIcs.ATMOS.2023.7.
- [4] E. Bortoletto, R. N. van Lieshout, B. Masing, and N. Lindner, “Periodic Event Scheduling with Flexible Infrastructure Assignment,” in *24th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2024)*, vol. 123, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 4:1–4:18. DOI: 10.4230/OASIcs.ATMOS.2024.4.
- [5] P. Serafini and W. Ukovich, “A Mathematical Model for Periodic Scheduling Problems,” *SIAM Journal on Discrete Mathematics*, vol. 2, no. 4, pp. 550–581, 1989. DOI: 10.1137/0402049.
- [6] H. George, *Progress and Poverty: An Inquiry into the Cause of Industrial Depressions and of Increase of Want with Increase of Wealth; The Remedy*. D. Appleton and Company, 1879.
- [7] World Bank, “Transport,” World Bank Group, Tech. Rep., 2024. [Online]. Available: <https://www.worldbank.org/en/topic/transport/overview>.
- [8] United Nations, “Sustainable Transport, Sustainable Development: Interagency Report for Second Global Sustainable Transport Conference,” United Nations, Tech. Rep., 2021. [Online]. Available: https://sdgs.un.org/sites/default/files/2021-10/Transportation%20Report%202021_FullReport_Digital.pdf.
- [9] United Nations, “SDG Progress Report (2024),” United Nations, Tech. Rep., 2024. [Online]. Available: <https://unstats.un.org/sdgs/files/report/2024/SG-SDG-Progress-Report-2024-advanced-unedited-version.pdf>.

- [10] European Union, *Treaty on European Union (Consolidated Version)*, Treaty of Maastricht, 1992. [Online]. Available: <https://www.refworld.org/legal/agreements/eu/1992/en/16882>.
- [11] European Commission and Directorate-General for Mobility and Transport, *Transport in the European Union – Current trends and issues*. Publications Office of the European Union, 2024. DOI: 10.2832/131741.
- [12] European Commission, *Sustainable and Smart Mobility Strategy – Putting European Transport on Track for the Future*, 2021. [Online]. Available: <https://transport.ec.europa.eu/system/files/2021-04/2021-mobility-strategy-and-action-plan.pdf>.
- [13] G. Desaulniers and M. D. Hickman, “Chapter 2 – Public Transit – Handbooks in Operations Research and Management Science,” in *Transportation*, vol. 14, Elsevier, 2007, pp. 69–127. DOI: 10.1016/S0927-0507(06)14002-5.
- [14] C. Liebchen, “The First Optimized Railway Timetable in Practice,” *Transportation Science*, vol. 42, no. 4, pp. 420–435, 2008. DOI: 10.1287/trsc.1080.0240.
- [15] L. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maroti, A. Schrijver, A. Steenbeek, and R. Ybema, “The new dutch timetable: The or revolution,” *Erasmus University Rotterdam, Econometric Institute, Econometric Institute Report*, 2008.
- [16] A. Schöbel, “An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation,” *Transportation Research Part C: Emerging Technologies*, vol. 74, pp. 348–365, 2017. DOI: <https://doi.org/10.1016/j.trc.2016.11.018>.
- [17] P. Schiewe, *Integrated optimization in public transport planning*. Springer, 2020, vol. 160.
- [18] C. Liebchen and R. H. Möhring, “The Modeling Power of the Periodic Event Scheduling Problem: Railway Timetables – and Beyond,” in *Algorithmic Methods for Railway Optimization*, Springer, 2007, pp. 3–40. DOI: 10.1007/978-3-540-74247-0_1.
- [19] P. Serafini and W. Ukovich, “A periodic job shop model,” *IFAC Proceedings Volumes*, vol. 22, no. 14, pp. 89–94, 1989, IFAC/CIRP/IFIP/IFORS Workshop on Decisional Structures in Automated Manufacturing, Genova, Italy, 18-21 September. DOI: [https://doi.org/10.1016/S1474-6670\(17\)54331-1](https://doi.org/10.1016/S1474-6670(17)54331-1).
- [20] P. Serafini and W. Ukovich, “A mathematical model for the fixed-time traffic control problem,” *European Journal of Operational Research*, vol. 42, no. 2, pp. 152–165, 1989. DOI: [https://doi.org/10.1016/0377-2217\(89\)90318-4](https://doi.org/10.1016/0377-2217(89)90318-4).
- [21] R. Hassin, “A flow algorithm for network synchronization,” *Operations Research*, vol. 44, no. 4, pp. 570–579, 1996.

- [22] C. Helmberg, T. Hofmann, and D. Wenzel, “Periodic event scheduling for automated production systems,” *INFORMS Journal on Computing*, vol. 34, no. 2, pp. 1291–1304, 2022.
- [23] I. Gertsbakh and P. Serafini, “Periodic transportation schedules with flexible departure times: An interactive approach based on the periodic event scheduling problem and the deficit function approach,” *European Journal of Operational Research*, vol. 50, no. 3, pp. 298–309, 1991. DOI: [https://doi.org/10.1016/0377-2217\(91\)90262-T](https://doi.org/10.1016/0377-2217(91)90262-T).
- [24] M. A. Odijk, “Construction of periodic timetables, Part 1: A cutting plane algorithm,” TU Delft, Tech. Rep. 94–61, 1994.
- [25] N. Lindner and J. Reisch, “An analysis of the parameterized complexity of periodic timetabling,” *Journal of Scheduling*, 2022. DOI: [10.1007/s10951-021-00719-1](https://doi.org/10.1007/s10951-021-00719-1).
- [26] K. Nachtigall, “Periodic Network Optimization and Fixed Interval Timetables,” Habilitation Thesis, Universität Hildesheim, 1998.
- [27] A. Brand, L. Allen, M. Altman, M. Hlava, and J. Scott, “Beyond authorship: Attribution, contribution, collaboration, and credit,” *Learned Publishing*, vol. 28, no. 2, pp. 151–155, 2015. DOI: [10.1087/20150211](https://doi.org/10.1087/20150211).
- [28] K. Nachtigall, “Cutting Planes for a Polyhedron Associated with a Periodic Network,” Institute of Flight Guidance, Tech. Rep., 1996. [Online]. Available: <https://elib.dlr.de/29333/>.
- [29] R. Borndörfer, H. Hoppmann, M. Karbstein, and N. Lindner, “Separation of cycle inequalities in periodic timetabling,” *Discrete Optimization*, vol. 35, p. 100 552, 2020. DOI: [10.1016/j.disopt.2019.100552](https://doi.org/10.1016/j.disopt.2019.100552).
- [30] N. Lindner and C. Liebchen, “Determining All Integer Vertices of the PESP Polytope by Flipping Arcs,” in *20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020)*, vol. 85, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020, 5:1–5:18. DOI: [10.4230/OASIcs.ATMOS.2020.5](https://doi.org/10.4230/OASIcs.ATMOS.2020.5).
- [31] R. Borndörfer, N. Lindner, and S. Roth, “A concurrent approach to the periodic event scheduling problem,” *Journal of Rail Transport Planning & Management*, vol. 15, p. 100 175, 2020, Best Papers of RailNorrköping 2019. DOI: [10.1016/j.jrtpm.2019.100175](https://doi.org/10.1016/j.jrtpm.2019.100175).
- [32] M. Goerigk, *PESPlib – A benchmark library for periodic event scheduling*, 2012. [Online]. Available: <https://timpasslib.aalto.fi/pesplib.html>.
- [33] *OASIcs, Volume 106, ATMOS 2022, Complete Volume*, vol. 106, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 1–240. DOI: [10.4230/OASIcs.ATMOS.2022](https://doi.org/10.4230/OASIcs.ATMOS.2022).
- [34] B. Masing, N. Lindner, and C. Liebchen, “Periodic timetabling with integrated track choice for railway construction sites,” *Journal of Rail Transport Planning & Management*, vol. 28, p. 100 416, 2023. DOI: [10.1016/j.jrtpm.2023.100416](https://doi.org/10.1016/j.jrtpm.2023.100416).

- [35] R. N. van Lieshout, “Integrated Periodic Timetabling and Vehicle Circulation Scheduling,” *Transportation Science*, vol. 55, no. 3, pp. 768–790, 2021. DOI: 10.1287/trsc.2020.1024.
- [36] K. Nachtigall and J. Opitz, “Solving Periodic Timetable Optimisation Problems by Modulo Simplex Calculations,” in *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS’08)*, vol. 9, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008, pp. 1–15. DOI: 10.4230/OASIcs.ATMOS.2008.1588.
- [37] M. Goerigk and A. Schöbel, “Improving the modulo simplex algorithm for large-scale periodic timetabling,” *Computers and Operations Research*, vol. 40, no. 5, pp. 1363–1370, 2013. DOI: 10.1016/j.cor.2012.08.018.
- [38] N. Lindner and C. Liebchen, “New Perspectives on PESP: T-Partitions and Separators,” in *19th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2019)*, vol. 75, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 2:1–2:18. DOI: 10.4230/OASIcs.ATMOS.2019.2.
- [39] J. Pätzold and A. Schöbel, “A Matching Approach for Periodic Timetabling,” in *16th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2016)*, vol. 54, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, 1:1–1:15. DOI: 10.4230/OASIcs.ATMOS.2016.1.
- [40] P. Großmann, S. Hölldobler, N. Manthey, K. Nachtigall, J. Opitz, and P. Steinke, “Solving Periodic Event Scheduling Problems with SAT,” in *Advanced Research in Applied Artificial Intelligence*, Springer, 2012, pp. 166–175. DOI: 10.1007/978-3-642-31087-4_18.
- [41] G. P. Matos, L. M. Albino, R. L. Saldanha, and E. M. Morgado, “Solving periodic timetabling problems with SAT and machine learning,” *Public Transport*, 2020. DOI: 10.1007/s12469-020-00244-y.
- [42] C. Liebchen, “Periodic Timetable Optimization in Public Transport,” in *Operations Research Proceedings 2006*, Springer Berlin Heidelberg, 2007, pp. 29–36. DOI: 10.1007/978-3-540-69995-8_5.
- [43] M. Joswig and K. Kulas, “Tropical and ordinary convexity combined,” *Advances in Geometry*, vol. 10, no. 2, pp. 333–352, 2010. DOI: 10.1515/advgeom.2010.012.
- [44] M. Joswig and G. Loho, “Weighted digraphs and tropical cones,” *Linear Algebra and its Applications*, vol. 501, pp. 304–343, 2016. DOI: 10.1016/j.laa.2016.02.027.
- [45] C. Liebchen and L. Peeters, “Integral cycle bases for cyclic timetabling,” *Discrete Optimization*, vol. 6, no. 1, pp. 98–109, 2009. DOI: 10.1016/j.disopt.2008.09.003.

- [46] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt, and K. A. Zweig, “Cycle bases in graphs characterization, algorithms, complexity, and applications,” *Computer Science Review*, vol. 3, no. 4, pp. 199–243, 2009. DOI: 10.1016/j.cosrev.2009.08.001.
- [47] R. Bacher, P. De La Harpe, and T. Nagnibeda, “The lattice of integral flows and the lattice of integral cuts on a finite graph,” *Bulletin de la Société Mathématique de France*, vol. 125, no. 2, pp. 167–198, 1997. DOI: 10.24033/bsmf.2303.
- [48] A. Schrijver, *Combinatorial Optimization – Polyhedra and Efficiency*. Springer, 2003.
- [49] M. Develin and B. Sturmfels, “Tropical convexity,” *Documenta Mathematica*, vol. 9, pp. 1–27, 2004. [Online]. Available: <http://eudml.org/doc/51129>.
- [50] M. Joswig, *Essentials of tropical combinatorics*. American Mathematical Society, 2021, vol. 219.
- [51] G. Cohen, S. Gaubert, and J.-P. Quadrat, “Duality and separation theorems in idempotent semimodules,” *Linear Algebra and its Applications*, vol. 379, pp. 395–422, 2004. DOI: 10.1016/j.laa.2003.08.010.
- [52] K. Nachtigall and J. Opitz, “A modulo network simplex method for solving periodic timetable optimisation problems,” in *Operations Research Proceedings 2007*, Springer Berlin Heidelberg, 2008, pp. 461–466.
- [53] G. M. Ziegler, *Lectures on polytopes*. Berlin: Springer-Verlag, 1995, vol. 152. DOI: 10.1007/978-1-4613-8431-1.
- [54] M. Beck and R. Sanyal, *Combinatorial reciprocity theorems: an invitation to enumerative geometric combinatorics*. American Mathematical Society, 2018, vol. 195.
- [55] W. Tutte, “Lectures on matroids,” *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, vol. 69B, no. 1 and 2, 1965. DOI: 10.6028/jres.069B.001.
- [56] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler, *Oriented Matroids*, 2nd ed. Cambridge University Press, 1999. DOI: 10.1017/CB09780511586507.
- [57] A. Postnikov, “Permutohedra, Associahedra, and Beyond,” *International Mathematics Research Notices*, vol. 2009, no. 6, pp. 1026–1106, 2009. DOI: 10.1093/imrn/rnn153.
- [58] C. Liebchen, “Finding Short Integral Cycle Bases for Cyclic Timetabling,” in *Algorithms – ESA 2003*, Springer, 2003, pp. 715–726. DOI: 10.1007/978-3-540-39658-1_64.
- [59] R. P. Stanley, “A Zonotope Associated with Graphical Degree Sequences,” in *Applied Geometry And Discrete Mathematics*, American Mathematical Society, 1991, pp. 555–570. DOI: 10.1090/dimacs%2F004%2F42.

- [60] P. Galashin, A. Postnikov, and L. Williams, “Higher secondary polytopes and regular plabic graphs,” *Advances in Mathematics*, vol. 407, p. 108 549, 2022. DOI: 10.1016/j.aim.2022.108549.
- [61] P. Galashin and A. Postnikov, “Purity and Separation for Oriented Matroids,” *Memoirs of the American Mathematical Society*, vol. 289, no. 1439, 2023. DOI: 10.1090/memo/1439.
- [62] J. Richter-Gebert and G. M. Ziegler, “Zonotopal tilings and the Bohne-Dress theorem,” in *Contemporary Mathematics*, vol. 178, American Mathematical Society, 1994, pp. 211–232. DOI: 10.1090/conm/178/01902.
- [63] M. Goerigk and C. Liebchen, “An Improved Algorithm for the Periodic Timetabling Problem,” in *17th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2017)*, vol. 59, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 12:1–12:14. DOI: 10.4230/OASIcs.ATMOS.2017.12.
- [64] J. D. Horton, “A Polynomial-Time Algorithm to Find the Shortest Cycle Basis of a Graph,” *SIAM Journal on Computing*, vol. 16, no. 2, pp. 358–366, 1987. DOI: 10.1137/0216026.
- [65] N. Lindner and R. van Lieshout, “Benders Decomposition for the Periodic Event Scheduling Problem,” in *Operations Research Proceedings 2021*, Springer International Publishing, 2022, pp. 289–294.
- [66] C. Helmberg, T. Hofmann, and D. Wenzel, “Periodic event scheduling for automated production systems,” *INFORMS Journal on Computing*, vol. 34, no. 2, pp. 1291–1304, 2022. DOI: 10.1287/ijoc.2021.1101.
- [67] M. Goerigk, A. Schöbel, and F. Spühler, “A Phase I Simplex Method for Finding Feasible Periodic Timetables,” in *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021)*, vol. 96, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 6:1–6:13. DOI: 10.4230/OASIcs.ATMOS.2021.6.
- [68] M. Benichou, J. M. Gauthier, P. Girodet, G. Hentges, G. Ribiere, and O. Vincent, “Experiments in mixed-integer linear programming,” *Mathematical Programming*, vol. 1, no. 1, pp. 76–94, 1971. DOI: 10.1007/BF01584074.
- [69] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2022. [Online]. Available: <https://www.gurobi.com>.
- [70] E. Schede, J. Brandt, A. Tornede, M. Wever, V. Bengs, E. Hüllermeier, and K. Tierney, “A Survey of Methods for Automated Algorithm Configuration (Extended Abstract),” in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, Journal Track, International Joint Conferences on Artificial Intelligence Organization, 2023, pp. 6964–6968. DOI: 10.24963/ijcai.2023/791.
- [71] R. M. Lusby, J. Larsen, M. Ehrgott, and D. Ryan, “Railway track allocation: Models and methods,” *OR Spectrum*, vol. 33, no. 4, pp. 843–883, 2011. DOI: 10.1007/s00291-009-0189-0.

- [72] Allianz pro Schiene e.V., *Das Schienennetz in Deutschland*, 2023. [Online]. Available: <https://www.allianz-pro-schiene.de/themen/infrastruktur/schienennetz/>.
- [73] L. Peeters, “Cyclic Railway Timetable Optimization,” Ph.D. dissertation, Erasmus Universiteit Rotterdam, 2003.
- [74] 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*, vol. 142, p. 103 805, 2022. DOI: 10.1016/j.trc.2022.103805.
- [75] N. Megiddo, “Partial and complete cyclic orders,” *Bulletin of the American Mathematical Society*, vol. 82, no. 2, pp. 274–276, 1976.
- [76] G. Caimi, L. Kroon, and C. Liebchen, “Models for railway timetable optimization: Applicability and applications in practice,” *Journal of Rail Transport Planning & Management*, vol. 6, no. 4, pp. 285–312, 2017. DOI: 10.1016/j.jrtpm.2016.11.002.
- [77] A. H. S. Blainey and J. Preston, “Barriers to passenger rail use: A review of the evidence,” *Transport Reviews*, vol. 32, no. 6, pp. 675–696, 2012. DOI: 10.1080/01441647.2012.743489.
- [78] N. Lindner and B. Masing, *On the Split Closure of the Periodic Timetabling Polytope*, 2023. DOI: 10.48550/arXiv.2306.02746.
- [79] B. Masing, N. Lindner, and C. Liebchen, “Integrating Line Planning for Construction Sites into Periodic Timetabling via Track Choice,” in *23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023)*, vol. 115, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 5:1–5:15. DOI: 10.4230/OASICS.ATMOS.2023.5.
- [80] P. Schiewe and A. Schöbel, “Periodic Timetabling with Integrated Routing: Toward Applicable Approaches,” *Transportation Science*, vol. 54, no. 6, pp. 1714–1731, 2020. DOI: 10.1287/trsc.2019.0965.
- [81] *OASICS, Volume 115, ATMOS 2023, Complete Volume*, vol. 115, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, pp. 1–268. DOI: 10.4230/OASICS.ATMOS.2023.
- [82] J. Korst, E. Aarts, J. K. Lenstra, and J. Wessels, “Periodic multiprocessor scheduling,” in *Parle ’91 Parallel Architectures and Languages Europe*, Springer Berlin Heidelberg, 1991, pp. 166–178. DOI: 10.1007/978-3-662-25209-3_12.
- [83] E. Bortoletto, N. Lindner, and B. Masing, “Practical improvements for tropical neighborhood search,” Private communication, 2022.
- [84] J. Kienzle, T. G. Crainic, E. Frejinger, and S. Bisailon, “The intermodal railroad blocking and railcar fleet management planning problem,” CIRRELT – Centre interuniversitaire de recherche sur les réseaux d’entreprise, la logistique et les transports – Université de Montréal, Tech. Rep., 2021.

- [85] T. G. Crainic, M. Gendreau, and B. Gendron, *Network Design with Applications to Transportation and Logistics*. Springer Cham, 2021. DOI: 10.1007/978-3-030-64018-7.