RECOVERABLE ROBUST PERIODIC TIMETABLING IN PUBLIC TRANSPORT

VERA GRAFE

Vom Fachbereich Mathematik der Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau zur Verleihung des akademischen Grades

Doktor der Naturwissenschaften
(*Doctor rerum naturalium, Dr. rer. nat.*)

genehmigte Dissertation

Datum der Disputation: 21.06.2024

Betreuerin:	Prof. Dr. Anita Schöbel
Zweitgutachter:	Prof. Dr. Marc Goerigk
Promotionskommission:	Prof. Dr. Anita Schöbel, Prof. Dr. Sven O. Krumke, Prof. Dr. Andreas Gathmann

DE-386

ABSTRACT

An important aspect of optimising public transport is finding a good timetable. On the one hand, short travel times are important from the passengers' point of view. On the other hand, tight timetables without buffer times are prone to delays, which are inevitable in practice and highly dissatisfactory for the passengers. Hence, a good timetable should also have some degree of delay resistance. Often a periodic timetable is desirable, i.e. a timetable which repeats in a regular pattern (e.g. every hour). However, delays do in general not occur periodically, so many robust timetable models only consider aperiodic timetables.

In our work we analyse different robustness concepts in the context of periodic timetabling, where we allow aperiodic delays. The main focus lies on *recoverable robustness*, which in our context leads to the integration of the *Periodic Event Scheduling Problem* (*PESP*) and *Delay Management* (*DM*). These two problems are usually considered in two different networks: while PESP is formulated in a periodic network, DM is considered in an aperiodic network, whose construction uses the timetable as input. Hence, a first challenge when integrating these two problems is to bridge the gap between periodicity and aperiodicity by solving PESP also in the aperiodic network. For this purpose we develop a new timetabling model – *Periodic Timetabling in Aperiodic Network* (*PTTA*) – which finds a periodic timetable using the aperiodic network.

Using this preparatory work, we then develop the *Recoverable Robust Periodic Timetabling Problem* (*RRPT*), which is the first one to consider periodic timetables with aperiodic delays. Since we have multiple objective functions, namely the travel time and the worst-case delay, we consider several variants of the problem. Furthermore, we present three equivalent formulations for RRPT, compare their performance in a computational study and identify some properties of recoverable robust timetables.

Due to the high complexity of the problem, which is due to both PESP and DM being NP-complete, solving RRPT to optimality on large networks is an unrealistic goal. Hence, we develop several heuristics for finding feasible and "good" solutions. By modifying the heuristics and testing them for different parameter choices, we identify the most promising variants.

Apart from recoverable robustness, many other robustness concepts exist in the literature, among them *strict robustness*, *light robustness* and *adjustable robustness*. We apply them to the problem at hand and analyse the relations between the resulting models in special cases. Furthermore, we compare them with respect to the *real travel time*, which is the sum of the travel time in the undisturbed setting and the worst-case delay, and discuss for which of these concepts considering aperiodic delays is actually equivalent to only considering periodic delays, and for which it does indeed make a difference. We conclude that recoverable robust timetables are superior, but come with the disadvantage of being the most difficult to compute.

ZUSAMMENFASSUNG

Ein wichtiger Aspekt bei der Optimierung von öffentlichen Verkehrssystemen ist die Suche nach einem guten Fahrplan. Einerseits sind kurze Fahrzeiten aus Sicht der Fahrgäste wichtig. Andererseits sind knapp geplante Fahrpläne ohne Pufferzeiten anfällig für Verspätungen, die in der Praxis unvermeidlich und für die Fahrgäste höchst unbefriedigend sind. Daher sollte ein guter Fahrplan auch eine gewisse Verspätungsresistenz aufweisen. Oft ist ein periodischer Fahrplan wünschenswert, d. h. ein Fahrplan, der sich regelmäßig wiederholt (z. B. jede Stunde). Verspätungen treten jedoch im Allgemeinen nicht periodisch auf, sodass viele robuste Fahrplanmodelle nur aperiodische Fahrpläne betrachten.

In unserer Arbeit analysieren wir verschiedene Robustheitskonzepte im Kontext periodischer Fahrplanung, wobei wir aperiodische Verspätungen zulassen. Das Hauptaugenmerk liegt dabei auf der *wiederherstellenden Robustheit*, was in unserem Kontext zur Integration des *Periodic Event Scheduling Problems* (*PESP*) und *Delay Managements* (*DM*) führt. Diese beiden Probleme werden normalerweise in zwei verschiedenen Netzwerken betrachtet: Während das PESP in einem periodischen Netzwerk formuliert wird, wird DM in einem aperiodischen Netz betrachtet, dessen Konstruktion den Fahrplan als Input verwendet. Daher besteht eine erste Herausforderung bei der Integration dieser beiden Probleme darin, die Lücke zwischen Periodizität und Aperiodizität zu schließen, indem das PESP auch im aperiodischen Netz gelöst wird. Zu diesem Zweck entwickeln wir ein neues Modell für die Fahrplanerstellung – *Periodic Timetabling in Aperiodic Network* (*PTTA*) – das einen periodischen Fahrplan unter Verwendung des aperiodischen Netzwerks findet.

Anhand dieser Vorarbeiten entwickeln wir dann das *Recoverable Robust Periodic Timetabling Problem* (*RRPT*), das erstmals periodische Fahrpläne mit aperiodischen Verspätungen betrachtet. Da wir mehrere Zielfunktionen haben, nämlich die Reisezeit und die Worst-Case-Verspätung, betrachten wir mehrere Varianten des Problems. Außerdem präsentieren wir drei äquivalente Formulierungen für RRPT, vergleichen ihre Performance und identifizieren einige Eigenschaften von wiederherstellend robusten Fahrplänen.

Aufgrund der hohen Komplexität des Problems, die daraus resultiert, dass sowohl PESP als auch DM NP-vollständig sind, ist die optimale Lösung von RRPT auf großen Netzen ein unrealistisches Ziel. Aus diesem Grund entwickeln wir mehrere Heuristiken, um zulässige und „gute" Lösungen zu finden. Durch Modifizierung der Heuristiken und Testen verschiedener Parameter identifizieren wir die vielversprechendsten Varianten.

Neben der wiederherstellbaren Robustheit gibt es in der Literatur noch viele andere Robustheitskonzepte, darunter *strenge Robustheit*, *leichte Robustheit* und *anpassbare Robustheit*. Wir wenden diese Konzepte auf das vorliegende Problem an und analysieren die Beziehungen zwischen den resultierenden Modellen in Spezialfällen. Außerdem vergleichen wir sie im Hinblick auf die reale Reisezeit, die sich aus der Summe der Reisezeit im ungestörten Zustand und der Worst-Case-Verspätung zusammensetzt, und diskutieren, für welche dieser Konzepte die Berücksichtigung aperiodischer Verspätungen tatsächlich äquivalent dazu ist, nur periodische Verspätungen zu berücksichtigen. Wir kommen zu dem Schluss, dass RRPT qualitativ überlegene Fahrpläne liefert, allerdings den Nachteil der schwierigen Berechenbarkeit mit sich bringt.

ACKNOWLEDGMENTS

First and foremost, I want to thank my supervisor Prof. Dr. Anita Schöbel for her support and helpful advice. She has always been a source of inspiration for me and she has the ability to make seemingly unsurmountable problems suddenly look much less threatening. I also thank Prof. Dr. Marc Goerigk for agreeing to read and evaluate this thesis.

Special thanks go to the proofreaders of this thesis, Sarah Roth and Reena Urban, for their helpful suggestions, which contributed significantly to its improvement. I also thank Reena Urban for her valuable input during one of our group meetings, which helped me solving a problem I had.

I cannot imagine a more pleasant work environment than in the optimisation group. I am grateful to all my colleagues, current and former, for shaping this wonderful experience by interesting discussions, numerous board game nights, exciting foosball matches and lots of cake!

Grazie, Fabian Chlumsky-Harttmann, per imparare l'italiano con me e per portarmi litri di tè in ufficio.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

MOTIVATION

Public transport plays an essential role in today's mobility. In particular under consideration of the omnipresent impact of climate change and the resulting necessity to reduce greenhouse emissions, the need for good public transport systems becomes evident. Therefore, it is vital to design public transport systems as attractively as possible for the passengers.

An important task when pursuing this goal is finding a good timetable. In European transport systems, periodic timetables are widely used, i.e. timetables which repeat in a regular pattern (for example every hour). One advantage of such timetables is the better memorability for the passengers. Since passengers want to reach their destination as fast as possible, travel times should be short. However, in practice disturbances occur during operation of the timetable. Hence, it is often not possible to stick to the plan, which – due to numerous interdependencies in the public transport system – can lead to a lot of delays in the entire network. This is very unsatisfactory for the passengers, who, as a consequence, might refrain from using public transport in the future.

To prevent this, a good timetable should also have some degree of delay resistance – it should be *robust*. Hence, *robust optimisation* might be the right tool to tackle the problem. But what exactly does robust even mean? A lot of different robustness concepts exist in the literature, but not all of them are suitable for every problem. For timetabling, *recoverable robustness* is a promising concept: we have to be able to recover the nominal feasible timetable in every delay scenario, i.e. make it feasible for the realised scenario. The recovery is achieved by doing *delay management* – a topic which itself is well-researched and an important task in public transport. However, usually timetabling and delay management are two sequential planning steps and the timetable is already given as input when doing delay management. Hence, our goal is to integrate these two problems – periodic timetabling, which is usually modelled as *Periodic Event Scheduling Problem* (*PESP*), and *Delay Management* (*DM*). The difficulty when doing so lies in the fact that – unlike PESP – delay management is an aperiodic problem since delays do, in general, not occur periodically. Hence, an interesting aspect of robust periodic timetabling is bridging the gap between periodic timetabling and aperiodic delay management.

But is recoverable robustness really the right choice for timetabling? To answer this question, we compare it to other robustness concepts from the literature, namely *strict robustness*, *light robustness* and *adjustable robustness*.

LITERATURE REVIEW

Periodic Timetabling

Periodic timetabling is well studied in the literature. The problem is usually modelled as a *Periodic Event Scheduling Problem*, which has first been introduced in [SU89], where it was shown to be NP-complete. It aims at finding a feasible periodic timetable. Instead of only considering the feasibility problem, one can also consider different objective functions. In [Nac96] this is done by minimising the waiting times of the transferring passengers. A common objective is to minimise the total travel time of the passengers. An extension with variable trip time is presented in [KP03]. Apart from the standard formulation one can also use alternative formulations using cycle bases, see e.g. [Nac98; Gov99; PK01; Pee03; Lie03; LP02; LP09].

The practical use of PESP was demonstrated by the introduction of a new timetable for the Berlin underground in 2005 [Lie08], which was the first mathematically optimised railway timetable in practice. Also *Nederlandse Spoorwegen*, the largest railway company in the Netherlands, introduced a completely new timetable in 2006 by solving PESP with constraint programming techniques [Kro+09].

A constraint generation approach has been applied in [Odi96]. Branch-and-bound [Nac96] and branch-and-cut [Lie06] have been used to solve PESP. SAT solvers have proved to be particular useful [Gro11; Gro+12; Küm+15].

However, due to the high complexity of the problem, for many practical applications one has to resort to heuristic approaches. Heuristics for PESP include genetic algorithms [NV96], the modulo network simplex [NO08; GS13; GL17] and a matching approach [PS16]. A concurrent approach running several solution methods in parallel is presented in [BLR20]. It combines mixed integer programming techniques, the modulo network simplex, SAT approaches and a new max cut heuristic.

Naturally, one can consider a lot of possible extensions for PESP, including more modelling details, such as operational costs [Lin00], passenger routing [SS15; SS20] or track choice [MLL23] to name just a few. In [LM07] it is shown how different planning aspects can be modelled as PESP.

Surveys on periodic timetabling can be found in [Lus+11; CT12; CKL17].

Robust Optimisation

The roots of robust optimisation go back to [Soy73]. However, it started to emerge as an own field of research much later with the pioneer work by Ben-Tal and Nemirovski [BN98; BN99; BN00] and El Ghaoui et al. [EL97; EOL98]. Early approaches on robust optimisation tend to be quite conservative, but many other concepts were introduced over the last decades, e.g. *light robustness* [FM09; Sch14], *adjustable robustness* [Ben+04], *recoverable robustness* [Lie+09] and many more. For a thorough introduction to robust optimisation we refer to [BEN09]. A survey is given in [GMT14] and an overview on many different robustness concepts can be found in [GS16].

Robust Timetabling

The literature on robust timetabling mainly treats aperiodic timetabling, i.e. the timetables are not required to follow a regular pattern. In [FM09] the concept of light robustness has been introduced and applied to aperiodic timetabling. Recoverable robustness has been introduced and used for the same application in [Lie+09]. Also [Cic+09; DAn+11] use recoverable robustness for aperiodic timetabling and present pseudo-polynomial algorithms for simple graph types (paths and trees) for a single delay. A version for arbitrarily many delays is considered in [Cic+08]. A similar concept, recovery-to-optimality, is studied in [GS10; GS14]. Here, the idea is to recover the timetable not only to a feasible but to an optimal solution for the realised scenario such that the recovery costs are minimal. The authors compare this new concept to several other robustness concepts in a numerical study. A bicriteria approach considering the travel time and the robustness of the timetable as objective functions is studied in [SK09]. Three different robustness measures based on waiting time rules are used. In [CCF12] a Lagrangian relaxation heuristic is used for finding approximately Pareto optimal aperiodic timetables, where both the nominal efficiency and the estimated robustness of the timetable are used as objective functions.

Improving a given timetable with regards to robustness is considered in [FSZ09] using stochastic programming and light robustness methods. In [Mül+21; Mül+22] the robustness of a transportation system – not only a timetable – is estimated with the help of machine learning. This oracle is then used in a local search heuristic and a genetic algorithm to improve robustness. Increasing the robustness of a given timetable – aperiodic or periodic – is also studied in [Jov+17] by reallocating buffer on headway activities. The authors show that this problem can be modelled as a multidimensional knapsack problem.

In the literature on robust *periodic* timetabling, usually also the disturbances are assumed to be periodic. This is the case in [Goe15] for the concept recovery-to-optimality. While small instances can be solved exactly, the author presents heuristic approaches which are suitable for different instance sizes. Similar to our idea, periodic timetabling and delay management are integrated in [Pät21]. However, only periodic disturbances are considered, while we do not require any periodicity in the delays. Two simplified models, using a no-wait policy (i.e. trains do not wait for delayed feeder trains) or shrinking the scenario set, are presented. A similar approach is also presented in [Pol+19]. Here, the timetable is required to be adjustable such that it is still feasible in every scenario, which can be interpreted as integrating delay management with an always-wait strategy.

[Lie+10] is one of the few papers where periodic timetabling with possibly aperiodic source delays is considered. Similar to our approach, the objective when computing a periodic timetable consists of a timetabling part and a delay management part. However, there are some key differences to our model. It is a stochastic model, using the approximate expected delay as part of the objective function. Several simplifying assumptions, like a strict no-wait policy and fixed driving times, are used. Furthermore, the objective function is approximated by a convex piecewise-linear function. The authors describe their approach as an extension of light robustness. After computing the timetable, it is then evaluated using a more detailed delay management model. Another stochastic programming approach is pursued in [Kro+06; KDV07]. The authors consider both the computation of a timetable for a single line and the improvement of a given timetable for a network. A branch-and-bound algorithm for this model is given in [Mar17].

Instead of taking specific source delays into account, another approach is to use the structure of the network. This is done in [Goe12], where a local search algorithm for robust periodic timetabling is presented. The robustness of the timetable is evaluated with a linear function depending on the position of each activity within its trip.

For surveys on robust timetabling we refer to [CT12], [CKL17] and [LLB18].

In total, we can say that so far most robust optimisation models in the literature either consider aperiodic timetabling, i.e. timetables which are not required to repeat in a regular pattern, or periodic timetabling where also the source delays are assumed to be periodic. Robust periodic timetabling with aperiodic source delays has hardly been treated in the literature so far.

CONTRIBUTION AND OUTLINE

In this thesis, we investigate robust periodic timetabling problems with possibly aperiodic source delays, where we consider several robustness concepts from the literature: strict robustness, light robustness, adjustable robustness and recoverable robustness. The main focus lies on the latter, which leads to an integration of two well-researched problems, the Periodic Event Scheduling Problem and the Delay Management Problem. The novelty of our approach lies in considering periodic timetables with aperiodic disturbances. Apart from developing models for robust periodic timetabling, we also present heuristic methods and show the qualitative superiority of recoverable robust timetables over timetables obtained using other robustness concepts.

The remainder of this thesis is structured as follows. In Chapter 2 we summarise the basic notations, definitions and models from public transport optimisation and robust optimisation needed in the course of this thesis.

In Chapter 3 we briefly explain how we model uncertainties in periodic timetabling.

Since our goal is to integrate the Periodic Event Scheduling Problem (PESP) with the aperiodic Delay Management Problem (DM), Chapter 4 is dedicated to the question how PESP can be solved in an aperiodic network, such that both problems of interest can be considered in the same network. We develop several equivalent formulations for the problem and compare them in a computational study.

These insights are then used in Chapter 5 to formulate the Recoverable Robust Periodic Timetabling Problem (RRPT). Since the problem has multiple objectives, we consider different variants of the problem, putting a different emphasis on nominal travel time and worst-case delay. Furthermore, we develop several equivalent formulations for RRPT and compare them in a numerical study.

In Chapter 6 we have a look at some properties of recoverable robust periodic timetables. First of all, we identify several intuitive characteristics of timetables which do not hold any more when considering the recoverable robust version. Furthermore, we analyse the influence of the parameters in one of the problem variants from the previous chapter, namely the one minimising the nominal travel time while bounding the delay, and identify a special case in which the uncertainty set can be reduced.

Due to the high complexity of the problem RRPT, solving it exactly for large instances is a too ambitious goal. Hence, in Chapter 7 we develop several heuristic approaches and test their performance compared to the standard approach of solving PESP and DM sequentially.

Finally, Chapter 8 is concerned with applying some of the numerous other robustness concepts, namely strict robustness, light robustness and adjustable robustness, to the periodic timetabling problem. We present formulations for all of these concepts, analyse their relations to each other in special cases, and put a focus on the question to which extend they are able to incorporate the aperiodicity of the delays. Furthermore, we compare them with respect to their real travel time – both theoretically and in a computational study.

We conclude this thesis in Chapter 9 and give directions for further research.

PUBLICATIONS

Parts of this thesis have already been published. Large parts of Chapter 4 have been published in [GS21]. The results of Chapter 5 have been previously published in [GS23].

2 PRELIMINARIES

In this chapter we introduce the basic notions and concepts used in this thesis. We assume basic knowledge on graph theoretical concepts and linear and integer programming. For an introduction to these topics we refer to [HK00] and [NW88].

Public transport planning consists of several planning steps, e.g. stop location, line planning, timetabling, vehicle scheduling, delay management and crew scheduling. In this thesis, we focus on the steps timetabling and delay management, which are introduced in the following sections. Furthermore, we consider robust optimisation. A lot of different robustness concepts exist in the literature. The ones relevant for this thesis are explained in Section 2.5.

For simplicity we will sometimes use the term train when talking about a vehicle in a public transport system. However, everything also applies to buses, metros and similar modes of transport.

2.1 PUBLIC TRANSPORT NETWORKS AND LINE PLANS

Since timetabling is part of a chain of planning steps, we first establish some basic notions from the previous planning stages, from which the input of the actual timetabling step is derived. One central structure in the early planning stages is the *public transport network*.

Definition 2.1 (Public Transport Network (PTN))**.** A *public transport network* is an undirected graph (V, E) consisting of a set V of stations and a set E of direct connections between them.

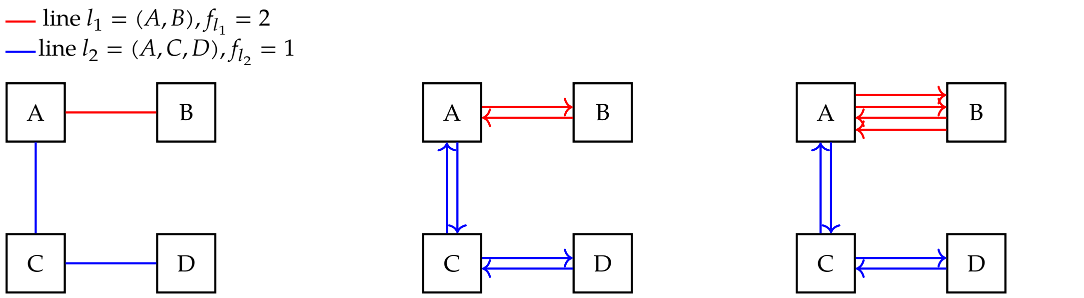
Given the PTN, we can describe how vehicles move through the system along *lines*.

Definition 2.2 (Line concept)**.** A *line* is an (undirected) path in the PTN. The *line plan* $\mathcal{L}$ is the set of all lines. For every line $l \in \mathcal{L}$ we have a frequency $f_l \in \mathbb{N}$, describing how often the line l is operated in a given planning period I. Together we obtain a *line concept* $(\mathcal{L}, f)$.

The lines are served in both directions and each line yields several *trips.*

Definition 2.3 (Directed line, trip)**.** Given a PTN (V, E) and a line concept $(\mathcal{L}, f)$, we obtain two *directed lines* l^+, l^- for every line $l \in \mathcal{L}$. The set of directed lines is denoted by $\mathcal{L}^{dir}$. We define a *trip* t as the journey of a vehicle from the beginning of a line to its end. It is given as a directed path in the PTN. Note that every line l yields $2f_l$ trips, namely f_l trips in each direction.

These definitions are illustrated in Figure 2.1. For further details on line planning, including models and solution methods, see [Sch12].



(a) PTN with two lines.

(b) PTN with directed lines.

(c) The line concept yields $2(f_{l_1} + f_{l_2}) = 6$ trips.

Figure 2.1: Example for a PTN and a line concept.

2.2 PERIODIC TIMETABLING

The goal of timetabling is to assign arrival and departure times to the services in a public transport system. In the classic planning process, timetabling is situated after stop location and line planning, i.e. the public transport network is already given and different lines have already been established. In this thesis we consider *periodic* timetabling, i.e. the timetable is required to repeat in a regular pattern, namely every T minutes for some $T \in \mathbb{N}_{>0}$. In particular, for periodic timetabling it is not necessary to consider every single trip, but because of the repetitious behaviour it suffices to look at the (directed) lines (and their repetitions within the period in case of line frequencies higher than one). Based on the PTN and the line concepts we construct the *event-activity network*.

Definition 2.4 (Event-Activity Network (EAN))**.** Let a PTN (V,E) and a line concept $(\mathcal{L}, f)$ be given. We denote an *arrival event* as a triple (v,l,arr), meaning that the line $l \in \mathcal{L}^{dir}$ arrives at station $v \in V$. Analogously, we have *departure events* of the form (v,l,dep). We define the set of *events* as $\mathcal{E} := \mathcal{E}^{\text{arr}} \cup \mathcal{E}^{\text{dep}}$, where

$$\mathcal{E}^{\text{arr}} := \{(v,l,arr) : \text{line } l \in \mathcal{L}^{dir} \text{ arrives at station } v \in V\}$$

is the set of arrival events and

$$\mathcal{E}^{\text{dep}} := \{(v,l,dep) : \text{line } l \in \mathcal{L}^{dir} \text{ departs from station } v \in V\}$$

the set of departure events.

We also need to model processes between the events. Distinguishing between different process types we define a set of *activities* $\mathcal{A} = \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{transfer}} \cup \mathcal{A}_{\text{head}}$, where:

$$\begin{aligned}
\mathcal{A}_{\text{drive}} &:= \{((v_1,l,dep),(v_2,l,arr)) \in \mathcal{E}^{\text{dep}} \times \mathcal{E}^{\text{arr}} : \text{line } l \text{ directly goes from} \\
&\qquad \text{station } v_1 \text{ to station } v_2\} \\
\mathcal{A}_{\text{wait}} &:= \{((v,l,arr),(v,l,dep)) \in \mathcal{E}^{\text{arr}} \times \mathcal{E}^{\text{dep}}\} \\
\mathcal{A}_{\text{transfer}} &:= \{((v,l_1,arr),(v,l_2,dep)) \in \mathcal{E}^{\text{arr}} \times \mathcal{E}^{\text{dep}} : \text{there is a required transfer} \\
&\qquad \text{from } l_1 \text{ to } l_2 \text{ at station } v\} \\
\mathcal{A}_{\text{head}} &:= \{(i,j) \in \mathcal{E} \times \mathcal{E} : \text{ there has to be at least distance } h_{ij} \in \mathbb{N} \text{ between } i \text{ and } j\}.
\end{aligned}$$

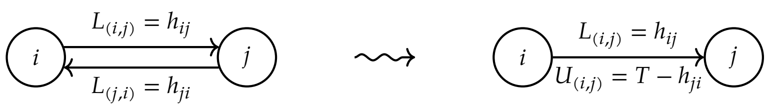


Figure 2.2: In periodic timetabling a pair of headway activities can be replaced by a single activity.

Together, the events and activities form the *event-activity network* $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$.

Driving activities $\underline{\mathcal{A}}_{\text{drive}}$ model a train line driving from one station to another, while *waiting activities* $\underline{\mathcal{A}}_{\text{wait}}$ represent a line waiting at a station. Passengers have the possibility to transfer between different lines, which is included by the *transfer activities* $\underline{\mathcal{A}}_{\text{transfer}}$. *Headway activities* $\underline{\mathcal{A}}_{\text{head}}$ are used to model safety regulations requiring a minimal distance between two consecutive departures or arrivals, or the safety restriction on single-track lines. They usually come in pairs: if we require security distances between i and j, we need two arcs (i,j) and (j,i).

For every activity $\underline{a} \in \underline{\mathcal{A}}$ a lower bound $L_{\underline{a}} \in \mathbb{N}$ and an upper bound $U_{\underline{a}} \in \mathbb{N}$ are given. The lower bound $L_{\underline{a}}$ is the minimal time necessary to perform the activity $\underline{a}$, e.g. for a driving activity this is the time needed for driving from one station to the next. Waiting activities have to be long enough such that the passengers can board or alight the vehicle and transfer activities have to be sufficiently long to allow passengers to switch trains. The upper bound $U_{\underline{a}}$ is the maximal time allowed for $\underline{a}$. A motivation for the upper bounds is that from the passengers' view, activities should not take overly long, e.g. passengers do not want to have a very long stop at every station. The purpose of the headway activities is to ensure that the distance between two trains is sufficiently large by imposing lower bounds. Since we consider periodic timetables, the events alternate, so we can also formulate this using lower as well as upper bounds: if for $\underline{a} = (i,j)$ we have $L_{\underline{a}} = h_{ij}$ and $U_{\underline{a}} = T - h_{ji}$, then the lower bound for the reversed arc (j,i) is automatically fulfilled, see Figure 2.2.

In the case that not all lines have the same frequency, we have to make copies of the events which take place more often. We want to spread these equally over the period, so in this case we have an additional type of activities, namely *synchronisation activities* $\underline{\mathcal{A}}_{\text{sync}}$.

Now, we can formally define a timetable.

Definition 2.5 (Periodic Timetable)**.** Given an EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$, a *timetable with period* T, $T \in \mathbb{N}_{>0}$, is a mapping $\pi\colon \underline{\mathcal{E}} \to \{0, \dots, T-1\}$ assigning a time to every event. To simplify notation, we set $\pi_i := \pi(i)$ for $i \in \underline{\mathcal{E}}$. A timetable is *feasible* if it respects the bounds on the activities, i.e. for an activity $\underline{a} = (i,j) \in \underline{\mathcal{A}}$ it has to hold

$$\pi_j - \pi_i + z_{\underline{a}} T \in [L_{\underline{a}}, U_{\underline{a}}]$$

for some $z_{\underline{a}} \in \mathbb{Z}$. We call the values $z_{\underline{a}}$ *modulo parameters*.

Note that due to the periodicity of the timetable we can assume without loss of generality that

$$0 \le L_{\underline{a}} \le T - 1 \text{ and } L_{\underline{a}} \le U_{\underline{a}} \le L_{\underline{a}} + T - 1 \text{ for all } \underline{a} \in \underline{\mathcal{A}} \tag{2.1}$$

since an interval of length greater than T does not pose any restriction.

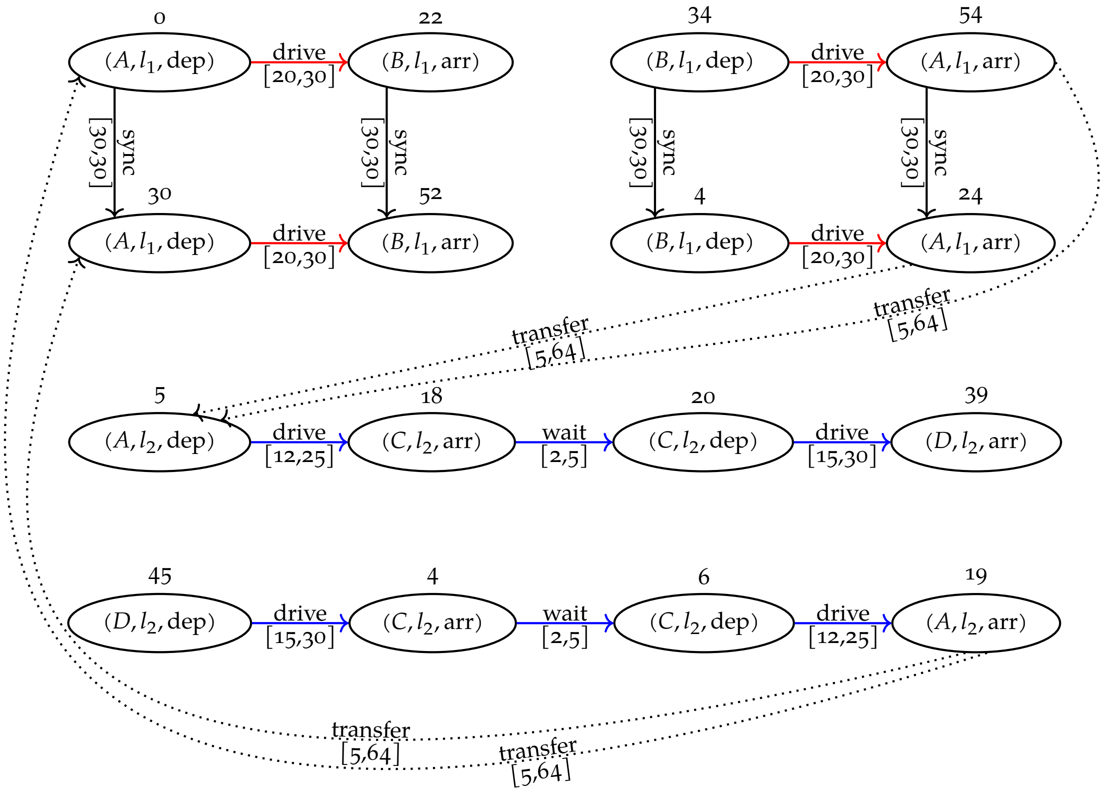


Figure 2.3: A possible EAN and a timetable with period $T = 60$ for the PTN and line concept from Figure 2.1.

Example 2.6. In Figure 2.3 we see a possible EAN for the PTN and line concept from Figure 2.1. Since line l_1 has frequency $f_{l_1} = 2$, we have two copies of all events corresponding to this line and synchronisation activities between them. At station A it is possible to transfer between both lines. For every activity $\underline{a}$ the interval $[L_{\underline{a}}, U_{\underline{a}}]$ is written next to the arc. The numbers above the nodes give a feasible timetable with period $T = 60$.

The *Periodic Event Scheduling Problem* (*PESP*), which was introduced in [SU89], asks for a feasible periodic timetable.

Periodic Event Scheduling Problem (PESP)

Input: EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with lower and upper bounds $[L_{\underline{a}}, U_{\underline{a}}]$ for $\underline{a} \in \underline{\mathcal{A}}$, period length T.

Task: Find a feasible periodic timetable π.

In the original version, PESP only asks for a feasible timetable. Additionally, we want to minimise the total travel time summed over all passengers. For $\underline{a} \in \underline{\mathcal{A}}$ let $w_{\underline{a}} \in \mathbb{N}$ be the number of passengers using activity $\underline{a}$. This yields the following IP formulation:

$$\begin{aligned}
\min \quad & f^{\text{PESP}} = \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}} \cdot (\pi_j - \pi_i + z_{\underline{a}} T) & & \text{(PESP)} \\
\text{s.t.} \quad & \pi_j - \pi_i + z_{\underline{a}} T \leq U_{\underline{a}} & \underline{a} = (i,j) \in \underline{\mathcal{A}} \qquad & (2.2) \\
& \pi_j - \pi_i + z_{\underline{a}} T \geq L_{\underline{a}} & \underline{a} = (i,j) \in \underline{\mathcal{A}} \qquad & (2.3) \\
& \pi_i \in \{0, \dots, T-1\} & i \in \underline{\mathcal{E}} \qquad & (2.4) \\
& z_{\underline{a}} \in \mathbb{Z} & \underline{a} \in \underline{\mathcal{A}}. \qquad & (2.5)
\end{aligned}$$

Instead of using node potentials, another approach, see [Nac98], is to use tensions, i.e. instead of assigning a time π_i to every event $i \in \underline{\mathcal{E}}$ we assign a duration $\xi_{\underline{a}}$ to every activity $\underline{a} \in \underline{\mathcal{A}}$.

Definition 2.7. Let $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ and $T \in \mathbb{N}$. We call $\xi \in \mathbb{Z}^{|\underline{\mathcal{A}}|}$ a *periodic tension* if there is a periodic timetable π with modulo parameters $z \in \mathbb{Z}^{|\underline{\mathcal{A}}|}$ such that

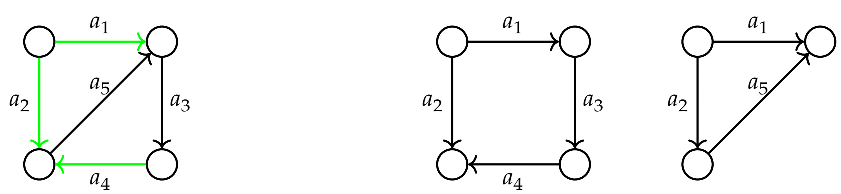
$$\pi_j - \pi_i + z_{\underline{a}} T = \xi_{\underline{a}} \quad \text{for all } \underline{a} = (i,j) \in \underline{\mathcal{A}}.$$

In order to find a periodic tension, we can use a cycle basis. For this purpose, we choose an arbitrary spanning tree $\underline{\mathcal{T}}$ in $\underline{\mathcal{N}}$. For every $\underline{a} \in \underline{\mathcal{A}} \setminus \underline{\mathcal{T}}$ there is a unique circuit $C_{\underline{a}}$ in $\underline{\mathcal{T}} \cup \{\underline{a}\}$, called *elementary circuit*. Denoting the arcs in forward and backward direction by $C_{\underline{a}}^+$ respectively $C_{\underline{a}}^-$, we can define the *network matrix* Γ by

$$\Gamma_{\underline{a},\underline{a}'} = \begin{cases} 1 & \underline{a}' \in C_{\underline{a}}^+, \\ -1 & \underline{a}' \in C_{\underline{a}}^-, \\ 0 & \underline{a}' \notin C_{\underline{a}} \end{cases}$$

for $\underline{a} \in \underline{\mathcal{A}} \setminus \underline{\mathcal{T}}, \underline{a}' \in \underline{\mathcal{A}}$. An example is given in Figure 2.4.

The network matrix Γ can be used to find a periodic tension by using the following theorem.



(a) A graph with a spanning tree marked in green. (b) The elementary circuits for the given spanning tree.

$$\Gamma = \left(\begin{array}{c|ccccc} & a_1 & a_2 & a_3 & a_4 & a_5 \\ \hline a_3 & 1 & -1 & 1 & 1 & 0 \\ a_5 & -1 & 1 & 0 & 0 & 1 \end{array} \right)$$

(c) The network matrix.

Figure 2.4: Example for a network matrix.

Theorem 2.8 ([Pee03]). *Let $\underline{\mathcal{N}}$ be an EAN and Γ the network matrix w.r.t. some spanning tree. $\xi \in \mathbb{Z}^{|\underline{\mathcal{A}}|}$ is a periodic tension in $\underline{\mathcal{N}}$ if and only if there is some $q \in \mathbb{Z}^{|\underline{\mathcal{A}}|-|\underline{\mathcal{E}}|-1}$ with $\Gamma\xi = Tq$.*

Using this result yields the cycle basis formulation of PESP, which is equivalent to the standard formulation, but needs significantly less computing time [PK01]:

$$
\begin{aligned}
\min \quad & w^T\xi && && \text{(PESP-cb)}\\
\text{s.t.} \quad & \Gamma\xi = Tq && && (2.6)\\
& L \leq \xi \leq U && && (2.7)\\
& \xi_{\underline{a}} \in \mathbb{Z} && \underline{a} \in \underline{\mathcal{A}} && (2.8)\\
& q_{\underline{a}} \in \mathbb{Z} && \underline{a} \in \underline{\mathcal{A}} \setminus \mathcal{T}. && (2.9)
\end{aligned}
$$

The objective function minimises the weighted tension, i.e. the passengers' travel time. Constraint (2.6) ensures that ξ is indeed a periodic tension with period T. The lower and upper bounds are respected due to (2.7).

Details about periodic timetabling can be found in the literature on PESP, a good introduction is given in [Lie06; Nac98].

2.3 ROLLING OUT THE NETWORK

For later planning stages taking place when a timetable is already given, such as vehicle scheduling or delay management, it is not sufficient to consider the periodic network $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$, because we have to be able to distinguish between the physical trains serving the same line. While in a periodic EAN the events represent the arrivals or departures of a *line* at some station, in an aperiodic EAN they model the arrival or departure of a single *trip*. Hence, instead of only considering the lines, we consider all trips of the lines separately. This means we have to "roll out" the periodic EAN to an aperiodic one in a time interval $I = [t_{\min}, t_{\max}]$. Let a feasible periodic timetable π for the network $\underline{\mathcal{N}}$ be given. For every $i \in \underline{\mathcal{E}}$ set

$$\pi_{\text{first}}(i) := \min\{\pi_i + kT : \pi_i + kT \geq t_{\min}, k \in \mathbb{Z}\},$$

$$\pi_{\text{last}}(i) := \max\{\pi_i + kT : \pi_i + kT \leq t_{\max}, k \in \mathbb{Z}\}.$$

These are the first respectively last times the event i occurs in the considered time horizon. The roll-out process then works as follows:

- For every $i \in \underline{\mathcal{E}}$ and $1 \leq s \leq K_i := \left\lfloor \frac{\pi_{\text{last}}(i) - \pi_{\text{first}}(i)}{T} \right\rfloor + 1$ construct an aperiodic event i_s with $\pi_{i_s} = \pi_{\text{first}}(i) + (s-1)T$. Let $\mathcal{E}(i) := \{i_s : 1 \leq s \leq K_i\}$ be the set of aperiodic events corresponding to the periodic event i and $\mathcal{E} := \cup_{i \in \underline{\mathcal{E}}} \mathcal{E}(i)$.
- For every $\underline{a} = (i,j) \in \underline{\mathcal{A}} \setminus \underline{\mathcal{A}}_{\text{head}}$ and $i_s \in \mathcal{E}(i)$ determine $j_t \in \mathcal{E}(j)$ (if it exists) such that $L_{\underline{a}} \leq \pi_{j_t} - \pi_{i_s} \leq U_{\underline{a}}$. We create an aperiodic activity $a_{st} = (i_s, j_t)$ and set $L_{a_{st}} = L_{\underline{a}}$, $U_{a_{st}} = U_{\underline{a}}$ and $w_{a_{st}} = w_{\underline{a}}$. For each pair $\underline{a} = (i,j), \underline{a}' = (j,i) \in \underline{\mathcal{A}}_{\text{head}}$ of headway activities and $i_s \in \mathcal{E}(i), j_t \in \mathcal{E}(j)$ create two aperiodic activities $a_{st} = (i_s, j_t), a_{ts} = (j_t, i_s)$ with $L_{a_{st}} = L_{\underline{a}}$ and $L_{a_{ts}} = T - U_{\underline{a}}$. If j_t does not exist, we are at the end of $[t_{\min}, t_{\max}]$ and nothing has to be done. All the activities constructed in this manner form the set $\mathcal{A}$.

Together, we obtain the *rolled out* network $\mathcal{N} = (\mathcal{E}, \mathcal{A})$. Note that using (2.1) the j_t in the roll-out process is uniquely determined, if it exists.

A particularity are the headway activities which ensure a security distance between two consecutive departures. For every pair of these headway arcs a_{st}, a_{ts} the lower bound has to be respected for exactly one of them, i.e. the pair $\underline{a} = (i, j), \underline{a}' = (j, i) \in \underline{\mathcal{A}}_{\text{head}}$ yields the following constraints:

$$\text{For all } 1 \leq s \leq K_i, 1 \leq t \leq K_j \text{ either } \pi_{j_t} - \pi_{i_s} \geq h_{ij} \text{ or } \pi_{i_s} - \pi_{j_t} \geq h_{ji},$$

where $h_{ij} = L_{\underline{a}}, h_{ji} = L_{\underline{a}'}$.

For further details on the roll-out procedure, we refer to [Lie+10].

2.4 DELAY MANAGEMENT

When the timetable is put into practice, we can often observe that not everything goes as planned: technical failures, accidents on the tracks or bad weather conditions are just a few examples for the occurrence of *source delays*. As a consequence, the original timetable π is not feasible any more and has to be adapted to the current situation. This problem is called *delay management* and has been introduced in [Sch01] and developed further in [Sch07]. It is important to note that delay management has to be done in the rolled out network $\mathcal{N} = (\mathcal{E}, \mathcal{A})$, because we have to be able to distinguish between the physical trains serving the same line. We distinguish between two different kinds of source delays: if an activity $a \in \mathcal{A}_{\text{train}} := \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}}$ has a source delay of $d_a \in \mathbb{N}$, this means the lower bound for a increases from L_a to $L_a + d_a$. If an event $i \in \mathcal{E}$ has a source delay of $d_i \in \mathbb{N}$, the event cannot happen earlier than $\pi_i + d_i$. The source delays propagate through the network, causing follow-up delays at later events. Furthermore, we have to make an important decision for each transfer activity that has become impossible due to a delay of the feeder train: we can either wait for a delayed train, allowing passengers to make their transfer but causing even more delays, or we have to cancel the transfer, in which case the passengers who wanted to take this transfer have to wait for a later connection. Hence, delay management consists of two tasks: deciding which transfers need to be maintained and which should be cancelled, and assigning a new time to every event. We call the resulting new timetable *disposition timetable*. Note that the upper bounds for the activities are ignored in delay management since it is often impossible to respect them in case of delays.

DELAY MANAGEMENT (DM)

Input: EAN rolled out for timetable π, source delays d.

Task: Find wait/no-wait decisions for every $a \in \mathcal{A}_{\text{transfer}}$ and a disposition timetable such that the sum of delays of all passengers is minimised.

Example 2.9. We are given (parts of) an EAN in Figure 2.5. Note that this is the rolled out version, i.e. we do not only consider lines but the single trips. There are two source delays: the departure of trip t_1 at station A is delayed by 5 minutes and the driving activity of

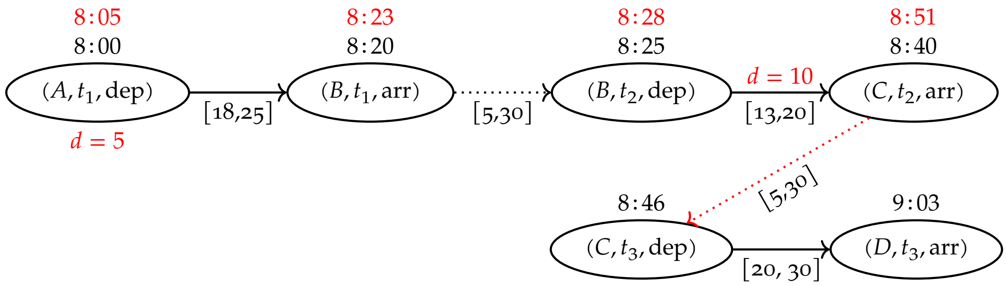


Figure 2.5: An example for a disposition timetable.

trip t_2 from station B to station C needs additional 10 minutes. The delays propagate through the network, where buffer times in the timetable can be used to absorb some of the delay. Note that the dotted arcs are transfer activities. In this example, t_2 waits for the delayed trip t_1, which causes further delay. On the other hand, the transfer from t_2 to t_3 is cancelled and t_3 departs on time.

For the decisions on the transfers we use binary variables

$$y_a = \begin{cases} 1 & \text{if } a \text{ is cancelled,} \\ 0 & \text{if } a \text{ is maintained.} \end{cases}$$

Variables x give the new time of the events. Furthermore, we have to take care of the security distances: if we have a pair $(i, j), (j, i) \in \mathcal{A}_{\text{head}}$ of headway activities, only for one of them the constraint imposed by the lower bound is active, depending on which of the events takes place first. Hence, we have further binary variables

$$p_{ij} = \begin{cases} 1 & \text{if } i \text{ takes place before } j, \\ 0 & \text{otherwise,} \end{cases}$$

see [SS10]. The objective is to minimise the delay the passengers have at their destination. If passengers miss a transfer, we use the common assumption that they take the next trip T minutes later and that this trip does not have a delay. Let $w_i \in \mathbb{N}$ be the number of passengers leaving the transport system at event $i \in \mathcal{E}$. For a fixed timetable π and source delays d this yields the following IP formulation (for an appropriately large constant M):

$$\min \sum_{i \in \mathcal{E}} w_i(x_i - \pi_i) + T \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a \tag{DM}$$

$$\begin{aligned}
\text{s.t.} \quad & x_i \geq \pi_i + d_i && i \in \mathcal{E} && (2.10)\\
& x_j - x_i \geq L_a + d_a && a = (i,j) \in \mathcal{A}_{\text{train}} && (2.11)\\
& M y_a + x_j - x_i \geq L_a && a = (i,j) \in \mathcal{A}_{\text{transfer}} && (2.12)\\
& M(1 - p_{ij}) + x_j - x_i \geq L_a && a = (i,j) \in \mathcal{A}_{\text{head}} && (2.13)\\
& p_{ij} + p_{ji} = 1 && (i,j),(j,i) \in \mathcal{A}_{\text{head}} && (2.14)\\
& x_i \in \mathbb{N} && i \in \mathcal{E} && (2.15)\\
& y_a \in \{0,1\} && a \in \mathcal{A}_{\text{transfer}} && (2.16)\\
& p_{ij} \in \{0,1\} && (i,j) \in \mathcal{A}_{\text{head}}. && (2.17)
\end{aligned}$$

The objective function minimises the sum of the deviation from the original timetable (weighted with the number of passengers leaving the public transport system at every event) and the number of missed transfers (weighted with the number of passengers using the transfers and a penalty T). The event source delays are respected due to (2.10). Note that we set $d_i = 0$ if $i \in \mathcal{E}$ does not have a source delay, and analogously for the activity delays we can also have $d_a = 0$ for $a \in \mathcal{A}_{\text{train}}$. Constraints (2.11) ensure that the delay is propagated correctly along driving and waiting activities and the source delays are respected. Similarly, Constraints (2.12) make sure that the delay propagation is handled correctly along transfer activities which are maintained. For each pair of headway activities, we have to respect the lower bound for exactly one of them, which is ensured by (2.13) and (2.14).

For the large constant, in [Sch09] it is shown that

$$M := \max_{i \in \mathcal{E}} d_i + \sum_{a \in \mathcal{A}_{\text{train}}} d_a + \sum_{\substack{(i,j) \in \mathcal{A}_{\text{head}}: \\ \pi_i > \pi_j}} (\pi_i - \pi_j + L_{ij})$$

is sufficiently large.

Note that the objective function, which is commonly used in the literature on delay management, can overestimate the actual delay. However, if the so-called *never-meet property* is fulfilled, which roughly speaking means that the paths of two delayed customers never meet, it is exact [Sch07].

Delay management was shown to be NP-complete in [Gat+05]. Heuristic approaches determining which trains should wait for delayed feeder trains are presented in [SS10; KS11; BS14; Rüc+17]. Numerous extensions of the model have been considered, for example integrating passenger re-routing [Dol+12], station capacities [Dol+15] and vehicle circulations [Fli+08; GSS22]. Overviews on delay management can be found in [Dol+18; Kön20].

2.5 ROBUST OPTIMISATION

When solving optimisation problems, we usually assume that all data is known exactly. However, when handling real-world problems, we often face some form of uncertainty. The parameters in the model might only be determined approximately. Or they might not be known yet, so we can only use a forecast. Maybe some disturbances occur and change the parameters. Hence, we want to take these uncertainties into account when solving the optimisation problem: we want to have a solution which is "robust". But what does robust even mean? The literature has a lot of different answers to this question in store. In this section we introduce those robustness concepts which are used in the course of this thesis.

A standard optimisation problem without any uncertainty might have the following form:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & x \in \mathbb{R}^n, \end{aligned} \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ for some $n, m \in \mathbb{N}$. If we take uncertainty into account, the objective function as well as the constraints can depend on some parameter from a given *uncertainty set* $\mathcal{U}$. Hence, we do not have a single problem any more, but a whole family of problems $(\mathrm{P}(r))_{r\in\mathcal{U}}$ of the following form:

$$\begin{aligned} \min \quad & f(x,r) \\ \text{s.t.} \quad & g(x,r) \leq 0 \\ & x \in \mathbb{R}^n, \end{aligned} \tag{P(r)}$$

where $f(\cdot, r) : \mathbb{R}^n \to \mathbb{R}$ and $g(\cdot, r) : \mathbb{R}^n \to \mathbb{R}^m$. In this thesis, we assume a finite uncertainty set, i.e. the scenarios are given explicitly. For an overview on different types of uncertainty see [GS16].

2.5.1 *Strict Robustness*

Robust optimisation has first been introduced by [Soy73] and [BN98]. To distinguish their concept from others, we call it *strict robustness*. Here, a feasible solution has to fulfil the constraints for every $r \in \mathcal{U}$. An optimal solution minimises the objective function in the worst case, i.e. the problem becomes

$$\begin{aligned} \min \quad & \sup_{r\in\mathcal{U}} f(x,r) \\ \text{s.t.} \quad & g(x,r) \leq 0 \qquad\qquad r \in \mathcal{U} \\ & x \in \mathbb{R}^n. \end{aligned} \tag{SRC}$$

This problem is called the *strictly robust counterpart* (*SRC*).

In many applications this strict approach is necessary, e.g. in matters of security. If we build a bridge, we need to be sure that it will not collapse when there is a lot of traffic or a heavy storm. However, for a lot of settings this high level of caution leads to overly conservative solutions, which is why different robustness concepts were developed.

2.5.2 *Light Robustness*

For the application to timetabling, strict robustness is too conservative. To allow more flexibility, [FM09] and [Sch14] have hence introduced the concept of *light robustness*. In this concept, we consider a fixed nominal scenario $\hat{r}$. This could e.g. be the most likely scenario or the one without any disturbances. While the lightly robust solution should be feasible for $\hat{r}$, for other scenarios we allow infeasibility to some extend. However, the objective value should not deviate too much from the optimal objective value in the nominal case.

To formalise this, we define the *grade of infeasibility of x w.r.t. constraint i*, $i \in \{1, \dots, m\}$, as

$$\mathrm{goi}_i(x) := \max\{0, \sup_{r\in\mathcal{U}} g_i(x,r)\}$$

and the *grade of infeasibility of* x as

$$\mathrm{goi}(x) := \|(\mathrm{goi}_i(x))_{i \in \{1,\dots,m\}}\|$$

for some norm $\|\cdot\|$. Hence, the grade of infeasibility describes, in some sense, how close the solution is to the strictly robust solution (which has a goi of zero). Our goal is to find a solution x whose objective value $f(x, \hat{r})$ in the nominal scenario differs at most by $\rho \geq 0$ from the optimal objective value $\hat{f}$ of the nominal problem $P(\hat{r})$. The objective is to minimise $\mathrm{goi}(x)$.

Hence, the *lightly robust counterpart* (*LRC*) is given by

$$\begin{aligned}
\min \quad & \|\gamma\| && && (\mathrm{LRC}(\rho)) \\
\text{s.t.} \quad & f(x, \hat{r}) \leq \hat{f} + \rho \\
& g(x, \hat{r}) \leq 0 \\
& g(x, r) \leq \gamma && r \in \mathcal{U} \\
& x \in \mathbb{R}^n \\
& \gamma \in \mathbb{R}^m.
\end{aligned}$$

2.5.3 *Adjustable Robustness*

Ben-Tal et al. [Ben+04] argue that in real-world problems often not all decisions have to be made before the realisation of the uncertain data becomes known. Rather, only a subset of the variables has to be fixed "here and now", while for the others we can "wait and see" until the data becomes certain. For example, when producing some goods, in the beginning we have to decide on the quantity before we know the actual demand. However, for later production periods we can use the knowledge on the previous demand.

Formally, this means the set of variables is partitioned into two parts (μ, ν), where μ is the non-adjustable part (i.e. the "here-and-now variables") and ν the adjustable part (the "wait-and-see variables"). Let us again consider our uncertain problem $(\mathrm{P}(r))_{r \in \mathcal{U}}$. Using the variable partition $x = (\mu, \nu)$ we can rewrite this as

$$\begin{aligned}
\min \quad & c \\
\text{s.t.} \quad & f(\mu, \nu, r) \leq c \\
& g(\mu, \nu, r) \leq 0 \\
& \mu \in \mathbb{R}^{n_1}, \nu \in \mathbb{R}^{n_2}
\end{aligned}$$

and can hence assume without loss of generality that the adjustable part ν as well as the uncertain data r only show up in the constraints, but not in the objective function.

The *adjustable robust counterpart* (*ARC*) asks for values for the "here-and-now variables" μ such that for every scenario $r \in \mathcal{U}$ there are values for the "wait-and-see variables" ν^r such that together (μ, ν^r) is feasible for scenario r. The objective function stays the same. We hence write:

$$
\begin{aligned}
\min \quad & c & & \qquad\qquad \text{(ARC)}\\
\text{s.t.} \quad & f(\mu, \nu^r, r) \le c & & r \in \mathcal{U} \\
& g(\mu, \nu^r, r) \le 0 & & r \in \mathcal{U} \\
& \mu \in \mathbb{R}^{n_1} & & \\
& \nu^r \in \mathbb{R}^{n_2} & & r \in \mathcal{U}.
\end{aligned}
$$

2.5.4 *Recoverable Robustness*

The main focus in this thesis lies on recoverable robustness, which has been introduced in [Lie+09], also with the aim to apply it to timetabling. The idea is to find a solution for the nominal scenario such that for every scenario $r \in \mathcal{U}$ it is possible to *recover* this solution, i.e. make it feasible for the scenario r with *limited effort*. This recovery is achieved by applying a recovery algorithm A from a set $\mathcal{A}$ of admissible algorithms. By "limited effort" we mean that we have the possibility to restrict the actions the recovery algorithm is allowed to take by choosing the set $\mathcal{A}$. For example, we could require that the recovered solution does not deviate too much from the nominal feasible solution.

Given a feasible solution x to the nominal problem and a scenario $r \in \mathcal{U}$, the algorithm A returns a solution $A(x, r)$. The goal is that this solution is feasible for the scenario r. A schematic illustration of the concept is shown in Figure 2.6.

Denoting the set of feasible solutions for the undisturbed setting by $\mathcal{X}$, we can formulate the problem as follows:

$$\min_{(x,A)\in\mathcal{X}\times\mathcal{A}} \{f(x) : \forall r \in \mathcal{U} : g(A(x,r), r) \le 0\}. \tag{2.18}$$

In this version of the problem, no costs for the recovery are considered in the objective function. Hence, the solution $A(x, r)$ only needs to be feasible. Alternatively, we can incorporate the recovery costs (which in some way measure the difference between the solution x and the new solution $A(x, r)$) in the objective function. For this purpose, let c be some function measuring the recovery costs. We require $c(A(x, r)) \le \lambda$, where $\lambda \in \Lambda$

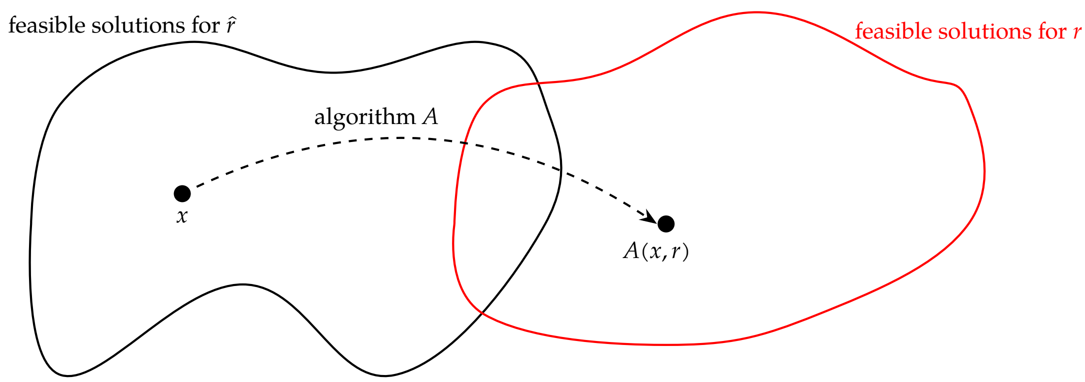


Figure 2.6: Schematic illustration of the concept of recoverable robustness.

is a variable (for an appropriately chosen set Λ) which is incorporated in the objective by some additional objective function f'. Then our problem can be formulated as

$$\min_{(x,\mathcal{A},\lambda)\in\mathcal{X}\times\mathcal{A}\times\Lambda} \{f(x) + f'(\lambda) : \forall r \in \mathcal{U} : g(A(x,r),r) \leq 0 \wedge c(A(x,r)) \leq \lambda\} \tag{2.19}$$

or even

$$\min_{(x,\mathcal{A},\lambda)\in\mathcal{X}\times\mathcal{A}\times\Lambda} \left\{ \begin{pmatrix} f(x) \\ f'(\lambda) \end{pmatrix} : \forall r \in \mathcal{U} : g(A(x,r),r) \leq 0 \wedge c(A(x,r)) \leq \lambda \right\} \tag{2.20}$$

if we consider a multi-objective version.

In this thesis, we consider several of these variants.

3 MODELLING UNCERTAINTIES IN TIMETABLING

In this chapter we explain which kind of uncertainty we have in timetabling and how the uncertain PESP looks like.

First, we re-write the standard PESP formulation such that it has the same form as the generic optimisation problem P from Section 2.5. For this purpose, note that the optimal modulo variables $z_{\underline{a}}$ can be easily determined if π is fixed, namely

$$z_{\underline{a}}(\pi) := \min\{z \in \mathbb{Z} : \pi_j - \pi_i + zT \geq L_{\underline{a}}\}.$$

Hence, by defining

$$\begin{aligned} f^{\text{PESP}}(\pi) &:= \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\pi_j - \pi_i + z_{\underline{a}}(\pi)T) \\ g^U_{\underline{a}}(\pi) &:= \pi_j - \pi_i + z_{\underline{a}}(\pi)T - U_{\underline{a}} \\ g^L_{\underline{a}}(\pi) &:= L_{\underline{a}} - \pi_j + \pi_i - z_{\underline{a}}(\pi)T \end{aligned}$$

we can then also write PESP as

$$\begin{aligned} \min \quad & f^{\text{PESP}}(\pi) \\ \text{s.t.} \quad & g^U(\pi) \leq 0 \\ & g^L(\pi) \leq 0 \\ & \pi \in \{0, \dots, T-1\}^{\underline{\mathcal{E}}}, \end{aligned}$$

where $g^U(\pi) := (g^U_{\underline{a}}(\pi))_{\underline{a}\in\underline{\mathcal{A}}}$, $g^L(\pi) := (g^L_{\underline{a}}(\pi))_{\underline{a}\in\underline{\mathcal{A}}}$.

In the context of timetabling, we assume that the uncertainty of the data lies in the lower bounds for the duration of the driving and waiting activities, i.e. in scenario $r \in \mathcal{U}$ there might occur some source delay $d^r_{\underline{a}}$ on an activity $\underline{a} \in \underline{\mathcal{A}}_{\text{train}}$, which increases the minimal time necessary to perform this activity from $L_{\underline{a}}$ to $L_{\underline{a}} + d^r_{\underline{a}}$. Such a delay $d^r_{\underline{a}} \geq 0$ is not known in advance. Note that although we only have source delays for activities from the set $\underline{\mathcal{A}}_{\text{train}}$, for simpler notation we sometimes do not distinguish between the different activity types by simply setting $d^r_{\underline{a}} := 0$ for $\underline{a} \in \underline{\mathcal{A}} \backslash \underline{\mathcal{A}}_{\text{train}}$.

There are applications where it makes sense to consider periodic delays, e.g. in case of construction sites which cause delays in a regular pattern. However, in general delays are not periodic. Hence, in this thesis we consider aperiodic delays.

To be able to take the aperiodicity of the delays into account, it is not sufficient to consider the periodic network $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$. Like for the problem of delay management we need an aperiodic network $\mathcal{N} = (\mathcal{E}, \mathcal{A})$, which treats every repetition of an event or activity (within a certain planning interval I of K periods) separately. This means that actually we do not have a source delay for some arc $\underline{a} \in \underline{\mathcal{A}}$ in the periodic network, but for every period $s \in \{1, \dots, K\}$ the corresponding aperiodic activity $\underline{a}_s$ can have a different delay. Hence, for every scenario $r \in \mathcal{U}$ we have a vector $d^r \in \mathbb{N}^{K|\underline{\mathcal{A}}|}$, containing a source delay for every repetition of every activity. Therefore, when considering the uncertain PESP, it does not suffice to have a single constraint for every constraint from the original PESP, but every constraint $g^L_{\underline{a}}(\pi) \leq 0$ induces K constraints $g^L_{\underline{a}}(\pi) + d^r_{\underline{a}_s} \leq 0$ for $s \in \{1, \dots, K\}$. We define

$$\mathcal{U}^r := \{(d^r_{\underline{a}_s})_{\underline{a} \in \underline{\mathcal{A}}_{\text{train}}} : s \in \{1, \dots, K\}\} \tag{3.1}$$

and

$$g^L(\pi, r) := (g^L(\pi) + d)_{d \in \mathcal{U}^r}.$$

The uncertain PESP for some $r \in \mathcal{U}$ is then given by

$$\begin{aligned} \min \quad & f^{\text{PESP}}(\pi) \\ \text{s.t.} \quad & g^U(\pi) \leq 0 \\ & g^L(\pi, r) \leq 0 \\ & \pi \in \{0, \dots, T-1\}^{|\underline{\mathcal{E}}|}, \end{aligned}$$

and by plugging in the definitions we get

$$\min \quad \sum_{\underline{a}=(i,j) \in \underline{\mathcal{A}}} w_{\underline{a}}(\pi_j - \pi_i + z_{\underline{a}} T) \tag{PESP(r)}$$

$$\text{s.t.} \quad \pi_j - \pi_i + z_{\underline{a}} T \leq U_{\underline{a}} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}} \tag{3.2}$$

$$\pi_j - \pi_i + z_{\underline{a}} T \geq L_{\underline{a}} + d^r_{\underline{a}_s} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}}, s \in \{1, \dots, K\} \tag{3.3}$$

$$z_{\underline{a}} \in \mathbb{Z} \qquad \underline{a} \in \underline{\mathcal{A}} \tag{3.4}$$

$$\pi_i \in \{0, \dots, T-1\} \qquad i \in \underline{\mathcal{E}}. \tag{3.5}$$

Note that we only have uncertainty in the constraints given by g^L, but neither in the constraints given by g^U nor in the objective function. Furthermore, note that some of the constraints (3.3) in the formulation above are redundant. We could instead write

$$\pi_j - \pi_i + z_{\underline{a}} T \geq L_{\underline{a}} + \max_{s \in \{1, \dots, K\}} d^r_{\underline{a}_s} \quad \underline{a} = (i,j) \in \underline{\mathcal{A}}.$$

However, for the recoverable robust approach it is essential that we keep the formulation PESP(r) as given since only then we can incorporate the aperiodicity of the source delays. For other robustness concepts it does not matter which of the two formulations we choose. We will look into this issue in more detail in Chapter 8.

Recall that in delay management we have two different types of source delays: in addition to the activity delays, which increase the lower bound for an activity $a \in \mathcal{A}$, we can also have event delays d_i for an event $i \in \mathcal{E}$, which means that the event i cannot start on time. Since our aim is to integrate timetabling and delay management, we also want to incorporate this type of delay. However, since the event delays model a deviation from the planned timetable, and hence imply that the timetable is infeasible, we cannot include this when writing down the uncertain PESP. However, note that event delays can be re-written as activity delays by considering a modified network as shown in [Sch09]. An example is given in Figure 3.1. If we have an event i with a source delay $d_i = \delta > 0$,

(a) The original EAN with a source delay on an event. (b) The modified EAN without source delays on events.

Figure 3.1: Event delays can be transformed into activity delays.

we add a virtual event i^0 and an activity (i^0, i) with $[L_{(i^0,i)}, U_{(i^0,i)}] = [0, 0], w_{i^0} = 1$ and $w_{(i^0,i)} = 0$. We can then simply replace the event delay d_i by an activity delay $\tilde{d}_{(i^0,i)}$. As an illustration, imagine that the event delay d_i is caused by a crew member being late. The activity (i^0, i) can then be interpreted as the crew member going to work and the activity delay as some disturbance on the work way. Using this transformation we can assume that we only have activity delays. However, since this comes at the cost of a larger network, we usually treat both types of delays separately when doing delay management – and will also do so in the integrated problem.

4

A MODEL FOR PERIODIC TIMETABLING IN AN APERIODIC NETWORK

Our aim is to develop a recoverable robust model for periodic timetabling. To this end, we have to integrate periodic timetabling and delay management. As pointed out in Sections 2.2 to 2.4, timetables are determined in a periodic network, but delay management is done in an aperiodic network since in general delays do not occur periodically. In order to integrate delay management into timetabling, we hence have to find a way to solve both problems in the same network.

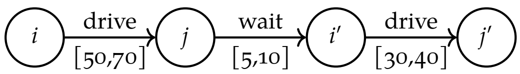
One way for such an integration is to develop a timetabling model which computes a periodic timetable in an aperiodic network, which is the goal of this chapter. We call the new model ***P****eriodic* ***T****ime****t****abling in* ***A****periodic Network* (*PTTA*).

Computing a periodic timetable in an aperiodic network was already considered in [VV06]. As opposed to our model, in [VV06] the decision on which transfer activities are needed is not part of the optimisation process but is fixed before by a simple heuristic rule. In [BL08] the problem is considered only for a single train line between two stations. A model putting an emphasis on passenger satisfaction and including the passenger routing is proposed in [Rob+16]. It uses the assumption that all drive and dwell times are fixed and it does not consider track safety constraints.

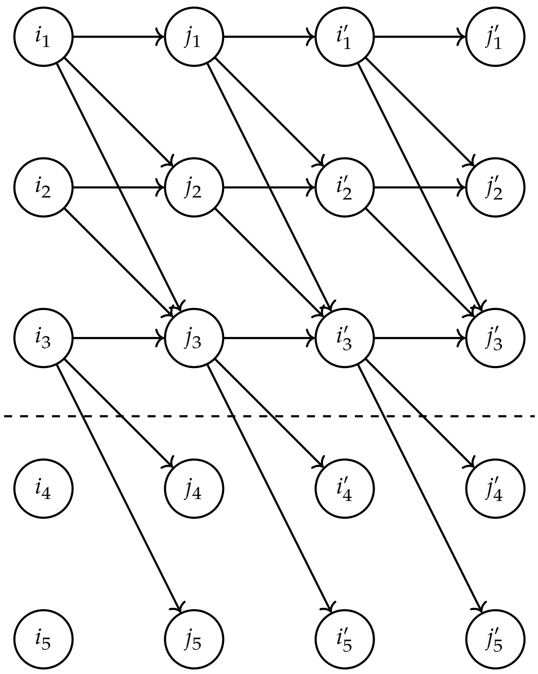
OUTLINE In Section 4.1 we introduce the new timetabling model and make several modifications to the model such that it better meets our needs. In Section 4.2 we compare the new model PTTA to the established model PESP and show that they are equivalent. We present some computational results in Section 4.3 and conclude the chapter with some final remarks and suggestions for further research.

4.1 A NEW TIMETABLING MODEL

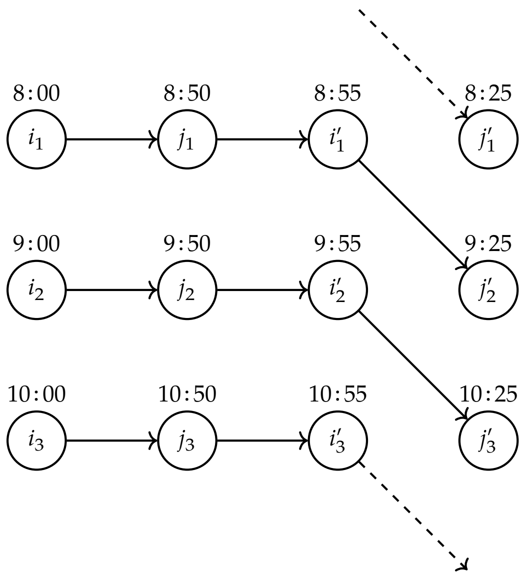
In Section 2.3 we saw how the periodic EAN can be rolled out for some planning horizon $[t_{\min}, t_{\max}]$, which is a necessary step before doing delay management. The goal of this chapter is to compute the timetable in the rolled out EAN. Hence, we cannot use the timetable when rolling out. However, the timetable information is important for determining the activities between the correct arrival and departure events. This is shown in Figures 4.1c and 4.1d, where two different timetables are used for the roll-out of the periodic EAN shown in Figure 4.1a, leading to two different aperiodic networks. Since we do not know beforehand which activities will be needed for the optimal timetable, we allow all possibilities (see Figure 4.1b) and leave it to the optimisation to choose the correct activities together with the optimal timetable. Note that every periodic event has several, say K, repetitions within the planning horizon I. Since it is possible that some activities start within I but end outside of it, we add a certain number b of periods at the end of the planning horizon, which we define formally in the following.



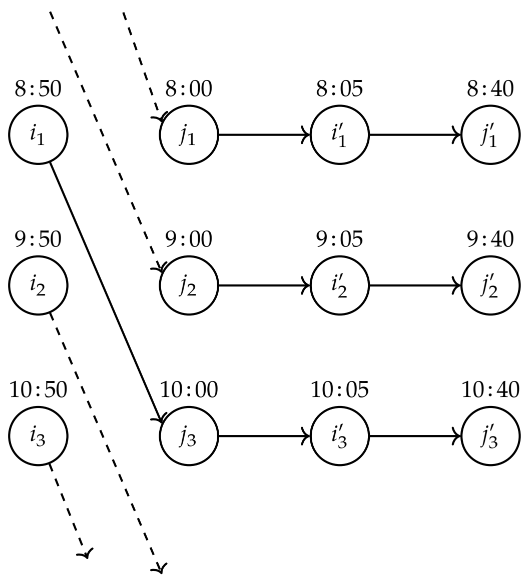
(a) Periodic EAN with $[L_{\underline{a}}, U_{\underline{a}}]$ given below the arcs.



(b) EAN rolled out with all possible activities ($b = 2$).



(c) Rolled out EAN after choosing a feasible timetable and the corresponding activities. The dashed arcs indicate the connections entering or leaving the planning horizon.



(d) Rolled out EAN with another feasible timetable, which results in different activities.

Figure 4.1: Rolling out a periodic EAN without knowing the timetable for the time interval [8:00,10:59] with $T = 60$, i.e. $K = 3$.

We adapt the roll-out procedure in the following way.

- We set $K := \left\lfloor \frac{t_{\max}-t_{\min}}{T} \right\rfloor + 1$ and $b_{\underline{a}} := \left\lceil \frac{U_{\underline{a}}}{T} \right\rceil$ for $\underline{a} \in \underline{\mathcal{A}}$, $b := \max_{\underline{a}\in\underline{\mathcal{A}}} b_{\underline{a}}$.
- For every periodic event $i \in \underline{\mathcal{E}}$ and $1 \leq s \leq K + b$ create an aperiodic event i_s. Let $\mathcal{E}(i) := \{i_s : 1 \leq s \leq K + b\}$ be the set of all aperiodic events corresponding to i. The set of all events is $\mathcal{E} := \cup_{i\in\underline{\mathcal{E}}}\mathcal{E}(i)$.
- For every periodic activity $\underline{a} = (i, j) \in \underline{\mathcal{A}}\backslash\underline{\mathcal{A}}_{\text{head}}$, for exactly one arc $\underline{a} = (i, j)$ of every pair of headway activities and for every $1 \leq s \leq K, s \leq t \leq K + b_{\underline{a}}$ create a *possible (aperiodic) activity* a_{st} with $L_{a_{st}} = L_{\underline{a}}$, $U_{a_{st}} = U_{\underline{a}}$ and $w_{a_{st}} = w_{\underline{a}}$. Let $\mathcal{A}(\underline{a}) := \{a_{st} = (i_s, j_t) : 1 \leq s \leq K, s \leq t \leq s + b_{\underline{a}}\}$ be the set of possible activities corresponding to $\underline{a}$. The set of all possible activities is

$$\mathcal{A} := \bigcup_{\underline{a}\in\underline{\mathcal{A}}} \mathcal{A}(\underline{a}). \tag{4.1}$$

The final network $(\mathcal{E}, \mathcal{A})$ is called the *rolled out network*. From now on, if we refer to the EAN rolled out with a given timetable π as constructed in Section 2.3, we use the notation $(\mathcal{E}(\pi), \mathcal{A}(\pi))$.

If we have a timetable π and an activity $\underline{a} = (i, j)$, for some s there may be no t such that $\pi_{j_t} - \pi_{i_s} \in [L_{\underline{a}}, U_{\underline{a}}]$, since the time of the event we would theoretically have to choose exceeds the planning horizon. Hence, we add b periods at the end, to ensure that we can define these activities. We will show in Lemma 4.4 that this is a reasonable choice.

We remark that when rolling out with a timetable, the number K_i of aperiodic events corresponding to a periodic event i depends on i. This is not the case when rolling out without knowing the timetable, where we have a constant K. However, this only makes a difference if our planning horizon $[t_{\min}, t_{\max}]$ covers a fractional number of periods. For example, if we consider 3.5 periods, some events will take place three times and some four times. Since this depends on the timetable, we cannot make this distinction when rolling out without knowing the timetable, so we have to consider each event four times. If we assume that we only consider whole periods, K_i is constant for all $i \in \underline{\mathcal{E}}$ and thus both procedures yield the same number of events. Additionally, we assume $K > 1$, since for $K = 1$ rolling out the network becomes obsolete.

The rolled out network contains not only the actual activities, but all possibilities for the activities. Thus, when fixing the timetable, we have to simultaneously solve an assignment problem: for each periodic activity we have to choose exactly one of the corresponding arcs in every considered period.

We can formulate our problem as follows:

PERIODIC TIMETABLING IN APERIODIC NETWORK (PTTA)

Input: Periodic EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, interval $I = [t_{\min}, t_{\max}]$.

Task: Find a timetable π with corresponding assignments in the network $\mathcal{N}$ rolled out for I such that the travel time summed over all passengers is minimal.

In order to formulate this as an MIP, we introduce binary variables

$$u_a = \begin{cases} 1 & \text{if } a \text{ is chosen,} \\ 0 & \text{otherwise} \end{cases}$$

for the assignment problem. Below we give our first formulation for finding a periodic timetable in the rolled out network:

$$\min \sum_{a=(i_1,j_t)\in\mathcal{A}} w_a \cdot u_a(\pi_{j_t} - \pi_{i_1}) \cdot K \tag{PTTA1}$$

$$\text{s.t. } \pi_{j_t} - \pi_{i_s} + M(u_a - 1) \leq U_a \qquad a = (i_s, j_t) \in \mathcal{A} \tag{4.2}$$

$$\pi_{j_t} - \pi_{i_s} + M(1 - u_a) \geq L_a \qquad a = (i_s, j_t) \in \mathcal{A} \tag{4.3}$$

$$\pi_{i_s} - \pi_{i_{s-1}} = T \qquad i_s \in \mathcal{E}, 2 \leq s \leq K + b \tag{4.4}$$

$$\sum_{a=(i_s,j_t)\in\mathcal{A}} u_a = 1 \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}}, 1 \leq s \leq K \tag{4.5}$$

$$\pi_i \geq t_{\min} \qquad i \in \mathcal{E} \tag{4.6}$$

$$\pi_{i_1} \leq t_{\min} + T - 1 \qquad i \in \underline{\mathcal{E}} \tag{4.7}$$

$$\pi_i \in \mathbb{N} \qquad i \in \mathcal{E} \tag{4.8}$$

$$u_a \in \{0, 1\} \qquad a \in \mathcal{A}. \tag{4.9}$$

The objective function minimises the total travel time over all passengers. Note that due to the periodicity of input data and timetable it is sufficient to consider only the first period here and multiply by K. In the case that an activity a is chosen, i.e. $u_a = 1$, Constraints (4.2) and (4.3) ensure that the upper and lower bounds for this activity are respected. If a is not selected, the constraints become redundant for appropriately chosen M. These constraints are the *timetabling constraints*. Constraints (4.4) are called *periodicity constraints* and ensure that the timetable has period T. For every periodic activity the *assignment constraints* (4.5) choose exactly one of the corresponding aperiodic activities in every period in such a way that it fits to the timetable constraints (4.2) and (4.3). Constraints (4.6) and (4.7) enforce that no event is scheduled earlier than $t_{\min}$ and that the first event takes place in the first period we consider.

Note that when rolling out with a timetable, we handled the headway activities differently than when rolling out without knowing a timetable. However, both ways of handling the headways are actually equivalent. Recall that we assume $K > 1$ since in the case $K = 1$ no roll-out is necessary.

Lemma 4.1. *Let* $\underline{a} = (i,j), \underline{a}' = (j,i) \in \underline{\mathcal{A}}_{\text{head}}$. *The following statements are equivalent:*

(a) *For all* $1 \leq s, t \leq K$ *we have either* $\pi_{j_t} - \pi_{i_s} \geq L_{\underline{a}} = h_{ij}$ *or* $\pi_{i_s} - \pi_{j_t} \geq L_{\underline{a}'} = h_{ji}$.

(b) *For all* $1 \leq s \leq K$ *there is some* $s \leq t \leq K + b_{\underline{a}}$ *such that* $\pi_{j_t} - \pi_{i_s} \in [L_{\underline{a}}, U_{\underline{a}}] = [h_{ij}, T - h_{ji}]$.

Proof. First, note that $U_{\underline{a}} = T - h_{ji} \leq T$ for $\underline{a} = (i,j) \in \underline{\mathcal{A}}_{\text{head}}$, i.e. $b_{\underline{a}} = 1$.
"(a)⇒(b)" Let $1 \leq s \leq K$. We consider the event j_{K+2}. Since the event i_s takes place in the s-th period, we have $\pi_{j_{K+2}} - \pi_{i_s} > T > h_{ij}$. Let now t be minimal such that $\pi_{j_t} - \pi_{i_s} \geq h_{ij} = L_{\underline{a}}$. It remains to show that $\pi_{j_t} - \pi_{i_s} \leq T - h_{ji} = U_{\underline{a}}$.

First case: $t > 1$. By minimality of t we have $\pi_{j_{t-1}} - \pi_{i_s} < h_{ij}$ and hence, $\pi_{i_s} - \pi_{j_{t-1}} \geq h_{ji}$. This yields $\pi_{j_t} - \pi_{i_s} = \pi_{j_{t-1}} + T - \pi_{i_s} \leq T - h_{ji} = U_{\underline{a}}$.
Second case: $t = 1$. Since $t \geq s$, this implies $s = 1 < K$, i.e. $s+1 = 2 \leq K$. Hence, we can use that (a) holds for the event i_2. Assume $\pi_{j_1} - \pi_{i_1} > T - h_{ji}$. Then $\pi_{j_1} - \pi_{i_2} = \pi_{j_1} - \pi_{i_1} - T > -h_{ji}$, i.e. $\pi_{i_2} - \pi_{j_1} < h_{ji}$. Hence, we must have $\pi_{j_1} - \pi_{i_2} \geq h_{ij}$, which in particular means that $\pi_{j_1} \geq \pi_{i_2}$. Since j_1 takes place in the first period and i_2 in the second period, this is a contradiction. Thus, our assumption was false and we have $\pi_{j_1} - \pi_{i_1} \leq T - h_{ji} = U_{\underline{a}}$.
The bounds on t will follow from Lemma 4.4.
"(b)⇒(a)" Let $1 \leq s \leq K$. By assumption there is some t' such that $\pi_{j_{t'}} - \pi_{i_s} \in [h_{ij}, T - h_{ji}]$. For $t \geq t'$ we have $\pi_{j_t} - \pi_{i_s} \geq \pi_{j_{t'}} - \pi_{i_s} \geq h_{ij}$. On the other hand, for $t < t'$ we have $\pi_{j_t} \leq \pi_{j_{t'}} - T$ and hence $\pi_{i_s} - \pi_{j_t} \geq \pi_{i_s} - \pi_{j_{t'}} + T \geq h_{ji}$. Thus, for every t one of the conditions is fulfilled. □

For simplicity, we will always handle the headways as given by the constraints in (b), regardless whether we roll out with or without using a given timetable. In the following we analyse and strengthen PTTA1.

The objective function of PTTA1 is quadratic since the π variables are multiplied with the u variables. However, we can easily linearise this using standard techniques. For this purpose we introduce a new variable F_a for $a = (i_1, j_t) \in \mathcal{A}$ to obtain the following equivalent formulation:

$$\begin{aligned} \min \quad & \sum_{a=(i_1,j_t)\in\mathcal{A}} w_a F_a \cdot K && \text{(PTTA2)} \\ \text{s.t.} \quad & (4.2) - (4.9) \\ & F_a \geq M(u_a - 1) + \pi_{j_t} - \pi_{i_1} \qquad a = (i_1, j_t) \in \mathcal{A} && (4.10) \\ & F_a \in \mathbb{N} \qquad a = (i_1, j_t) \in \mathcal{A}. && (4.11) \end{aligned}$$

Lemma 4.2. *PTTA1 and PTTA2 are equivalent.*

Proof. "⇒" Let (π, u) be a feasible solution to PTTA1. For $a = (i_s, j_t)$ set

$$F_a = \begin{cases} \pi_{j_t} - \pi_{i_s} & \text{if } u_a = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Then constraints (4.10) are fulfilled. Note that $F_a \geq 0$, and hence $F_a \in \mathbb{N}$, since $u_a = 1$ implies that $\pi_{j_t} - \pi_{i_s} \geq L_a \geq 0$. Since all other constraints are the same as before, (π, u, F) is feasible for PTTA2. Furthermore, the objective values coincide:

- If $u_a = 1$: $u_a(\pi_{j_t} - \pi_{i_s}) = \pi_{j_t} - \pi_{i_s} = F_a$
- If $u_a = 0$: $u_a(\pi_{j_t} - \pi_{i_s}) = 0 = F_a$.

"⇐" Let (π, u, F) be a feasible solution to PTTA2 with objective value f'. Then obviously (π, u) is feasible for PTTA1. Let f be the corresponding objective value. We have:

- If $u_a = 1$: $F_a \geq u_a(\pi_{j_t} - \pi_{i_s}) = \pi_{j_t} - \pi_{i_s}$
- If $u_a = 0$: $F_a \geq u_a(\pi_{j_t} - \pi_{i_s}) = 0$.

Plugging this into the objective function yields $f \leq f'$. □

We still need to specify how to choose the large constant M.

Lemma 4.3. *The constant $M := 2T - 1 + \max_{a \in \mathcal{A}} U_a$ is sufficiently large.*

Proof. We have to show that for every $a = (i_s, j_t) \in \mathcal{A}$ the following inequalities hold:

- $M \geq \pi_{i_s} - \pi_{j_t} + L_a$
- $M \geq \pi_{j_t} - \pi_{i_s} - U_a$
- $M \geq \pi_{j_t} - \pi_{i_s} - F_a$.

In order to see this, we use the following observations. First, using Constraints (4.4) inductively yields $\pi_{i_s} = \pi_{i_1} + (s-1)T$. Second, by definition of $\mathcal{A}$, we have $s - t \leq 0$ and $t - s \leq \left\lceil \frac{U_a}{T} \right\rceil \leq \frac{U_a}{T} + 1$. Finally, we have $t_{\min} \leq \pi_{i_1}, \pi_{j_1} \leq t_{\min} + T - 1$, which implies $\pi_{j_1} - \pi_{i_1} \leq T - 1$. Putting all this together we obtain

$$\pi_{i_s} - \pi_{j_t} + L_a = \pi_{i_1} - \pi_{j_1} + (s-t)T + L_a \leq T - 1 + L_a < M,$$

which shows the first inequality. Furthermore, it follows

$$\pi_{j_t} - \pi_{i_s} = \pi_{j_1} - \pi_{i_1} + (t-s)T \leq T - 1 + \left(\frac{U_a}{T} + 1\right) T = 2T - 1 + U_a \leq M.$$

Since $U_a, F_a \geq 0$, this implies the second and third inequality. □

When rolling out the network we only consider arcs of the form (i_s, j_t) for $s \leq t \leq s + b_{\underline{a}}$. The following lemma shows that these are indeed the only possible arcs in a feasible solution.

Lemma 4.4. *Let (4.4), (4.6) and (4.7) be fulfilled for some π. Let $\underline{a} = (i,j) \in \underline{\mathcal{A}}$ and $1 \leq s \leq K$. Then for $t \geq s + 1 + b_{\underline{a}}$ or $t \leq s - 1$ we have $\pi_{j_t} - \pi_{i_s} \notin [L_{\underline{a}}, U_{\underline{a}}]$.*

Proof. By periodicity we obtain for $t \geq s + 1 + b_{\underline{a}}$:

$$\begin{aligned} \pi_{j_t} - \pi_{i_s} &= (\pi_{j_1} + (t-1)T) - (\pi_{i_1} + (s-1)T) \geq 1 - T + (t-s)T \\ &\geq 1 - T + (1 + b_{\underline{a}})T \geq 1 + U_{\underline{a}} > U_{\underline{a}}. \end{aligned}$$

Similarly, for $t \leq s - 1$ we have:

$$\begin{aligned} \pi_{j_t} - \pi_{i_s} &= (\pi_{j_1} + (t-1)T) - (\pi_{i_1} + (s-1)T) \\ &\leq T - 1 + (t-s)T \leq T - 1 - T = -1 < L_{\underline{a}}. \end{aligned}$$ □

Hence, we do not have to consider arcs for other choices of s and t. Note that this may be a significant reduction, e.g. under the assumption (2.1) we have $U_{\underline{a}} \leq L_{\underline{a}} + T - 1 \leq 2(T-1)$ and hence $b_{\underline{a}} \leq 2$.

We can reduce the activities we have to consider even further with the following reasoning: because of the periodicity of the timetable, the choice of $u_{(i_1, j_t)}$ already determines the value of u for later periods. Hence, we only need to consider variables $u_{(i_1, j_t)} \in \mathcal{A}$

with i_1 being the event in the first period instead of $u_{(i_s,j_t)} \in \mathcal{A}$ for all i_s with $(i_s, i_t) \in \mathcal{A}$. This affects constraints (4.2), (4.3), (4.5), and (4.9) in PTTA2 and reduces the number of variables and constraints in our formulation considerably leading to the following IP:

$$\min \sum_{a=(i_1,j_t)\in\mathcal{A}} w_a F_a \cdot K \tag{PTTA3}$$

$$\pi_{j_t} - \pi_{i_1} + M(u_a - 1) \leq U_a \qquad a = (i_1, j_t) \in \mathcal{A} \tag{4.12}$$

$$\pi_{j_t} - \pi_{i_1} + M(1 - u_a) \geq L_a \qquad a = (i_1, j_t) \in \mathcal{A} \tag{4.13}$$

$$\pi_{i_s} - \pi_{i_{s-1}} = T \qquad i_s \in \mathcal{E}, 2 \leq s \leq K + b \tag{4.14}$$

$$\sum_{a=(i_1,j_t)\in\mathcal{A}} u_a = 1 \qquad (i,j) \in \underline{\mathcal{A}} \tag{4.15}$$

$$F_a \geq M(u_a - 1) + \pi_{j_t} - \pi_{i_1} \qquad a = (i_1, j_t) \in \mathcal{A} \tag{4.16}$$

$$\pi_i \geq t_{\min} \qquad i \in \mathcal{E} \tag{4.17}$$

$$\pi_{i_1} \leq t_{\min} + T - 1 \qquad i \in \underline{\mathcal{E}} \tag{4.18}$$

$$\pi_i \in \mathbb{N} \qquad i \in \mathcal{E} \tag{4.19}$$

$$u_a \in \{0, 1\} \qquad a = (i_1, j_t) \in \mathcal{A} \tag{4.20}$$

$$F_a \in \mathbb{N} \qquad a = (i_1, j_t) \in \mathcal{A}. \tag{4.21}$$

Lemma 4.5. *PTTA2 and PTTA3 are equivalent.*

Proof. "⇒" Let (π, u, F) be a solution to PTTA2. For $a = (i_1, j_t)$ set $u'_a := u_a$. Clearly, (π, u', F) is a feasible solution to PTTA3 and the objective values coincide.
"⇐" Let (π, u', F) be a solution to PTTA3. For $a = (i_s, j_t) \in \mathcal{A}$ set $u_a := u'_{(i_1, j_{t-s+1})}$. Note that since $a \in \mathcal{A}$, we have $s \leq t \leq s + b_{\underline{a}}$ with $a \in \mathcal{A}(\underline{a})$ and therefore $1 \leq t - s + 1 \leq 1 + b_{\underline{a}}$, which implies that also $(i_1, j_{t-s+1}) \in \mathcal{A}$. We show that (π, u, F) is a feasible solution to PTTA2:

- Let $a = (i_s, j_t) \in \mathcal{A}$. We have

$$\begin{aligned}&\pi_{j_t} - \pi_{i_s} + M(u_a - 1) = (\pi_{j_{t-s+1}} + (s-1)T) - (\pi_{i_1} + (s-1)T) + M(u_a - 1)\\ =&\pi_{j_{t-s+1}} - \pi_{i_1} + M(u'_{(i_1, j_{t-s+1})} - 1) \leq U_{(i_1, j_{t-s+1})} = U_a,\end{aligned}$$

  which shows constraints (4.2). Analogously we obtain (4.3).

- Let $(i,j) \in \underline{\mathcal{A}}, 1 \leq s \leq K$. We have

$$\sum_{a=(i_s,j_t)\in\mathcal{A}} u_a = \sum_{a=(i_1,j_{t-s+1})\in\mathcal{A}} u'_a = 1$$

  and hence, (4.5) holds.

- Constraints (4.4) and (4.6) to (4.11) follow immediately.

Consequently, (π, u, F) is a feasible solution to PTTA2 with the same objective value as (π, u', F). □

4.2 COMPARISON OF PTTA AND PESP

We now want to compare the new assignment-based model with the established model PESP. There are several questions we want to answer:

(a) Let an instance of PESP be given. We roll out the EAN without a timetable.

1. Let $(\tilde{\pi}, z)$ be a feasible (optimal) solution to PESP. Can we construct a feasible (optimal) solution to PTTA?
2. Let (π, u, F) be a feasible (optimal) solution to PTTA. Can we construct a feasible (optimal) solution to PESP?

(b) Let $(\tilde{\pi}, z)$ be a feasible (optimal) solution to PESP. We define PTTA$(\tilde{\pi})$ as the corresponding instance of PTTA when rolling out with the solution $(\tilde{\pi}, z)$, i.e. the activities are determined by $\tilde{\pi}$ and the assignment problem becomes trivial.

1. Is the solution which is constructed in the roll-out process feasible (optimal) for PTTA$(\tilde{\pi})$?
2. Let (π, u, F) be a feasible (optimal) solution to PTTA$(\tilde{\pi})$. Can we construct a feasible (optimal) solution to PESP?

While the questions in (b) are more of theoretical interest, the questions in (a) are vital to achieve our goal: if we can answer them positively, PESP and PTTA are equivalent, which will help us when integrating timetabling and delay management.

Let a periodic timetable be given. As an intermediate step we consider the roll-out w.r.t. this timetable as described in Section 2.3. The following lemma ensures that for any realisation i_s of event i (except for those at the end of the planning horizon) there exists a corresponding realisation j_t such that the timetabling constraints for the rolled out activity (i_s, j_t) are fulfilled.

Lemma 4.6. *Let $(\tilde{\pi}, z)$ be a feasible solution to PESP and π the solution constructed in the roll-out process w.r.t. $\tilde{\pi}$. Let $\underline{a} = (i, j) \in \underline{\mathcal{A}}$ and $k, l \in \mathbb{Z}$ such that $\pi_{\text{first}}(i) = \tilde{\pi}_i + kT$ and $\pi_{\text{first}}(j) = \tilde{\pi}_j + lT$. For any choice of $1 \leq s \leq K$ and $t := z_{\underline{a}} + k - l + s$, the bounds on activity (i_s, j_t) are fulfilled, i.e. $\pi_{j_t} - \pi_{i_s} \in [L_{\underline{a}}, U_{\underline{a}}]$.*

Proof. By definition of π we have $\pi_{i_s} = \pi_{\text{first}}(i) + (s-1)T = \tilde{\pi}_i + (k+s-1)T$ and $\pi_{j_t} = \pi_{\text{first}}(j) + (t-1)T = \tilde{\pi}_j + (l+t-1)T$. Hence, it follows

$$\pi_{j_t} - \pi_{i_s} = \tilde{\pi}_j - \tilde{\pi}_i + (l - k - s + t)T = \tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T \in [L_{\underline{a}}, U_{\underline{a}}]. \qquad \square$$

Corollary 4.7. *In the situation of Lemma 4.6 for $1 \leq s \leq K$ there exists an $s \leq t \leq s + b_{\underline{a}}$ with $\pi_{j_t} - \pi_{i_s} \in [L_{\underline{a}}, U_{\underline{a}}]$.*

Proof. We remark that by Lemma 4.4 it follows that for t as chosen in Lemma 4.6 we have $s \leq t \leq s + b_{\underline{a}}$. □

We can use these results to construct a solution in the network which was rolled out without a solution.

4.2.1 *Rolling out without a timetable*

Let an instance of PESP $(\underline{\mathcal{E}}, \underline{\mathcal{A}})$ be given and $(\mathcal{E}, \mathcal{A})$ be the EAN received by rolling out without knowing a timetable. Let $(\tilde{\pi}, z)$ be a solution to PESP. We define π as in the roll-out process with the timetable given, i.e. $\pi_{i_s} = \pi_{\text{first}}(i) + (s-1)T$. Furthermore, for $a = (i_s, j_t) \in \mathcal{A}$ we choose k, l as in Lemma 4.6 and set

$$u_a = \begin{cases} 1 & \text{if } t = z_{\underline{a}} + k - l + s, \\ 0 & \text{otherwise,} \end{cases}$$

and for $a = (i_1, j_t)$ we set

$$F_a = \begin{cases} \pi_{j_t} - \pi_{i_1} & \text{if } u_a = 1, \\ 0 & \text{otherwise.} \end{cases}$$

This construction gives us a feasible solution to PTTA2 in the rolled out network as the following lemma shows.

Lemma 4.8. *Let $(\tilde{\pi}, z)$ be a feasible solution to PESP with objective value $\tilde{f}$. Then (π, u, F) as defined above is a feasible solution to PTTA2 and the corresponding objective value is $f = K\tilde{f}$.*

Proof. We check that (π, u, F) fulfils all constraints:

- (4.2) and (4.3) are fulfilled by choice of u and Lemma 4.6.
- Let $i_s \in \mathcal{E}$, $2 \leq s \leq K + b$. By definition of π it follows

  $$\pi_{i_s} - \pi_{i_{s-1}} = (\pi_{\text{first}}(i) + (s-1)T) - (\pi_{\text{first}}(i) + (s-2)T) = T,$$

  which proves (4.4).
- Constraints (4.6) to (4.9) are obviously fulfilled.
- Let $\underline{a} = (i, j) \in \underline{\mathcal{A}}, 1 \leq s \leq K$. By Lemma 4.6 we have $\pi_{j_t} - \pi_{i_s} \in [L_{\underline{a}}, U_{\underline{a}}]$ for $t = z_{\underline{a}} + k - l + s$, which by Lemma 4.4 implies $t \leq s + b_{\underline{a}}$. In particular, $(i_s, j_t) \in \mathcal{A}$. By choice of u it follows $\sum_{a=(i_s,j_t)\in\mathcal{A}} u_a = 1$, i.e. constraints (4.5) are fulfilled.
- Let $a = (i_1, j_t) \in \mathcal{A}$.
  First case: $u_a = 1$. Then $F_a = \pi_{j_t} - \pi_{i_1} = M(u_a - 1) + \pi_{j_t} - \pi_{i_1}$.
  Second case: $u_a = 0$. Then $F_a = 0 > -M + \pi_{j_t} - \pi_{i_1} = M(u_a - 1) + \pi_{j_t} - \pi_{i_1}$.
  Hence, constraints (4.10) are fulfilled.
- For (4.11), $F_a \in \mathbb{Z}$ is clear. Note that by (4.3) $u_a = 1$ is only possible if $\pi_{j_t} \geq \pi_{i_1}$, which in particular means that $F_a \geq 0$ and therefore $F_a \in \mathbb{N}$.

Hence, (π, u, F) is indeed a feasible solution. For the objective value we obtain:

$$\begin{aligned} f = K \cdot \sum_{a=(i_1,j_t)\in\mathcal{A}} w_a F_a &= K \cdot \sum_{a=(i_1,j_t)\in\mathcal{A}: u_a = 1} w_a(\pi_{j_t} - \pi_{i_1}) \\ &\overset{(*)}{=} K \cdot \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}} T) = K \cdot \tilde{f}, \end{aligned}$$

where $(*)$ follows from the proof of Lemma 4.6. □

Again, let an instance of PESP $(\underline{\mathcal{E}}, \underline{\mathcal{A}})$ be given and $(\mathcal{E}, \mathcal{A})$ be the EAN received by rolling out without knowing a timetable. Let (π, u, F) be a feasible solution to PTTA2. For $i \in \underline{\mathcal{E}}$ we set

$$\tilde{\pi}_i := \pi_{i_1} \mod T,$$

i.e. there is some $r_i \in \mathbb{Z}$ such that $\pi_{i_1} = \tilde{\pi}_i + r_i T$. For $\underline{a} = (i,j) \in \underline{\mathcal{A}}$ there is some t such that $u_{(i_1,j_t)} = 1$. Set

$$z_{\underline{a}} := r_j - r_i + t - 1.$$

Also this construction works, i.e. we get a feasible solution to PESP with bounded objective function value.

Lemma 4.9. *Let (π, u, F) be a feasible solution to PTTA2 with objective value f. Then $(\tilde{\pi}, z)$ as defined above is a feasible solution to PESP and for its objective value $\tilde{f}$ we have $\tilde{f} \leq f \cdot \frac{1}{K}$.*

Proof. Let $\underline{a} = (i,j) \in \underline{\mathcal{A}}$. The following holds:

$$\begin{aligned}\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}} T &= (\pi_{j_1} - r_j T) - (\pi_{i_1} - r_i T) + z_{\underline{a}} T \\ &= (\pi_{j_t} - (t-1)T - r_j T) - (\pi_{i_1} - r_i T) + z_{\underline{a}} T \\ &= \pi_{j_t} - \pi_{i_1} - (r_j - r_i + t - 1)T + z_{\underline{a}} T \\ &= \pi_{j_t} - \pi_{i_1} \in [L_{\underline{a}}, U_{\underline{a}}].\end{aligned}$$

Hence, $(\tilde{\pi}, z)$ is a feasible solution to PESP. For the objective value we have:

$$\begin{aligned}\tilde{f} &= \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}} T) \\ &= \sum_{a=(i_1,j_t)\in\mathcal{A}:u_a=1} w_a(\pi_{j_t} - \pi_{i_1}) \\ &\overset{(*)}{\leq} \sum_{a=(i_1,j_t)\in\mathcal{A}:u_a=1} w_a F_a \\ &\overset{(**)}{\leq} \sum_{a=(i_1,j_t)\in\mathcal{A}} w_a F_a = f \cdot \frac{1}{K}.\end{aligned}$$

Here, $(*)$ follows from (4.10) and $(**)$ from $F_a \geq 0$. □

Putting the two constructions together, we finally conclude that we can in fact construct an optimal solution to PESP if we know an optimal solution to PTTA2 and vice versa. In particular, it makes no difference whether one computes a solution with PTTA2 or rolls out a solution obtained with PESP, i.e. PTTA2 and PESP are equivalent. The proof directly follows from Lemma 4.8 and Lemma 4.9.

Corollary 4.10. *If $(\tilde{\pi}, z)$ is an optimal solution to PESP, the solution (π, u, F) constructed in Lemma 4.8 is optimal for PTTA2. On the other hand, if (π, u, F) is an optimal solution to PTTA2, the solution $(\tilde{\pi}, z)$ constructed in Lemma 4.9 is optimal for PESP.*

Proof. Let $(\tilde{\pi}, z)$ be an optimal solution to PESP with objective value $\tilde{f}$. By Lemma 4.8 we obtain a feasible solution (π, u, F) to PTTA with objective value $f = K\tilde{f}$. Assume this is not optimal, i.e. there is a solution (π', u', F') with objective value $f' < f$. By Lemma 4.9 we get a solution $(\bar{\pi}, \bar{z})$ to PESP with objective value $\bar{f} \leq f' \cdot \frac{1}{K} < f \cdot \frac{1}{K} = \tilde{f}$, which is a contradiction to $(\tilde{\pi}, z)$ being an optimal solution.
On the other hand, let (π, u, F) be an optimal solution to PTTA with objective value f. Lemma 4.9 yields a feasible solution $(\tilde{\pi}, z)$ to PESP with objective value $\tilde{f} \leq f \cdot \frac{1}{K}$. Assume $(\tilde{\pi}, z)$ is not optimal, i.e. there is a solution $(\bar{\pi}, \bar{z})$ with objective value $\bar{f} < \tilde{f}$. By Lemma 4.8 we receive a solution (π', u', F') to PTTA with objective value $f' = K\bar{f} < K\tilde{f} \leq f$, which is a contradiction. □

4.2.2 *Rolling out with a timetable*

Let $(\underline{\mathcal{E}}, \underline{\mathcal{A}})$ be an instance of PESP and $(\tilde{\pi}, z)$ a feasible solution with objective value $\tilde{f}$. Let $(\mathcal{E}(\tilde{\pi}), \mathcal{A}(\tilde{\pi}))$ be the EAN received by rolling out with this solution and π the solution constructed in the roll-out process. Recall that in this case we do not have to solve an assignment problem, since the activities are already fixed.

Lemma 4.11. *Set $u_a = 1$ for all $a = (i_s, j_t) \in \mathcal{A}(\tilde{\pi})$ and $F_a = \pi_{j_t} - \pi_{i_1}$ for all $a = (i_1, j_t) \in \mathcal{A}(\tilde{\pi})$. Then (π, u, F) is a feasible solution to PTTA2$(\tilde{\pi})$ with objective value $f = K\tilde{f}$.*

Proof. Analogously to Lemma 4.8 we check the single constraints:

- In the roll-out process $\mathcal{A}(\tilde{\pi})$ is chosen such that (4.2) and (4.3) are fulfilled.
- Constraints (4.4) follow exactly as in the proof of Lemma 4.8.
- Let $\underline{a} = (i, j) \in \underline{\mathcal{A}}$ and $1 \leq s \leq K$. By construction of $\mathcal{A}(\tilde{\pi})$ there is exactly one t with $(i_s, j_t) \in \mathcal{A}(\tilde{\pi})$, namely the t as chosen in Lemma 4.6. By choice of u this implies $\sum_{a=(i_s,j_t)\in\mathcal{A}(\tilde{\pi})} u_a = 1$, which shows constraints (4.5).
- Constraints (4.6) to (4.11) are obviously fulfilled by choice of π, u and F.

Consequently, (π, u, F) is a feasible solution to PTTA2$(\tilde{\pi})$. For the objective value f we obtain

$$\begin{aligned} f = K \cdot \sum_{a=(i_1,j_t)\in\mathcal{A}(\tilde{\pi})} w_a F_a &= K \cdot \sum_{a=(i_1,j_t)\in\mathcal{A}(\tilde{\pi})} w_a(\pi_{j_t} - \pi_{i_1}) \\ &\overset{(*)}{=} K \cdot \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T) = K \cdot \tilde{f}, \end{aligned}$$

where $(*)$ follows from the proof of Lemma 4.6. □

Lemma 4.12. *Let (π', u', F') be a feasible solution to PTTA2$(\tilde{\pi})$ with objective value f'. For $i \in \underline{\mathcal{E}}$ we set $\bar{\pi}_i := \pi_{i_1} \mod T$, i.e. there is some $r_i \in \mathbb{Z}$ such that $\pi'_{i_1} = \bar{\pi}_i + r_i T$. For $\underline{a} = (i, j) \in \underline{\mathcal{A}}$ there is some t such that $u'_{(i_1,j_t)} = 1$. Set $\bar{z}_{\underline{a}} := r_j - r_i + t - 1$. Then $(\bar{\pi}, \bar{z})$ is a feasible solution to PESP and for its objective value $\bar{f}$ we have $\bar{f} \leq f' \cdot \frac{1}{K}$.*

Proof. The proof is analogous to the one of Lemma 4.9. □

Corollary 4.13. *If* $(\tilde{\pi}, z)$ *is an optimal solution to PESP, the solution* (π, u, F) *constructed in Lemma 4.11 is optimal for PTTA2*$(\tilde{\pi})$*.*

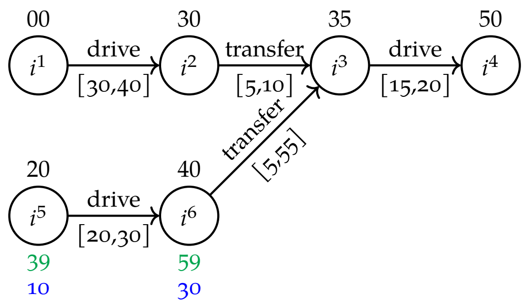
Proof. The proof is analogous to the first part in the proof of Corollary 4.10. □

Remark 4.14. While Corollary 4.13 is the analogon to the first part of Corollary 4.10, we do not get the same for the second part. In Lemma 4.11 we have only shown that the solution $(\tilde{\pi}, z)$ to PESP we used for rolling out yields a solution to PTTA2$(\tilde{\pi})$. However, to do the proof like in Corollary 4.10, we would need that an arbitrary solution $(\bar{\pi}, \bar{z})$ to PESP yields a solution to PTTA2$(\tilde{\pi})$. However, since the arcs present in the rolled out EAN depend on $(\tilde{\pi}, z)$, this cannot be done in the same way for a different solution. Indeed, if we start with a suboptimal solution $(\tilde{\pi}, z)$ when rolling out, then an optimal solution to PTTA2$(\tilde{\pi})$ will in general not yield an optimal solution to PESP. This can be seen in Example 4.15.

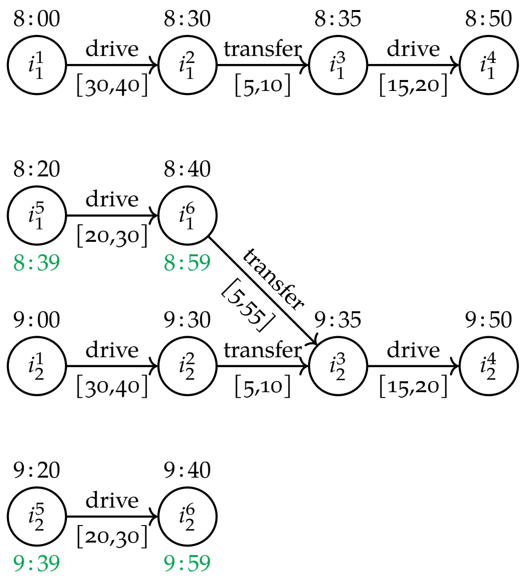
Example 4.15. We consider the PESP instance given in Figure 4.2a with the timetable $\tilde{\pi}$ given above the nodes. If we roll out the EAN with this solution for $I = [8:00, 9:59]$ with $T = 60$, i.e. $K = 2$, we obtain the instance PTTA2$(\tilde{\pi})$ together with a timetable π given in Figure 4.2b. However, this is not optimal, since we can improve the solution by replacing some of the times with the ones given in green. Let π' be this solution. Indeed, π' is optimal: we cannot schedule i_1^6 later than 8:59 because of Constraints (4.7). On the other hand, by (4.6) and periodicity we cannot schedule i_2^1 earlier than 9:00, which together with the lower bounds of the activities implies that the earliest possible time for i_2^3 is 9:35. Hence, we cannot reduce the transfer time from i^6 to i^3 any further. For all other activities the timetable is tight. As shown in Lemma 4.9, we can convert π' to a solution $\bar{\pi}$ to PESP, which is again given by replacing some of the times by the ones given in the green labels in the upper figure. We can easily see that this is not an optimal solution to PESP: we could reschedule the events i^5 and i^6 as given by the blue labels to obtain a solution for which the duration of every activity is equal to the lower bound, and hence optimal.

Remark 4.16. The previous example showed that the solution used for rolling out the EAN can have an enormous impact on the PTTA instance we obtain. The decision to match i_1^6 with i_2^3 instead of i_1^3 made us loose a lot of flexibility when choosing a timetable in the rolled out network. If we had chosen the blue timetable for rolling out, we would have obtained the EAN given in Figure 4.2c together with an optimal solution. This emphasises that for different PESP solutions we obtain different PTTA instances with different optimal objective values.

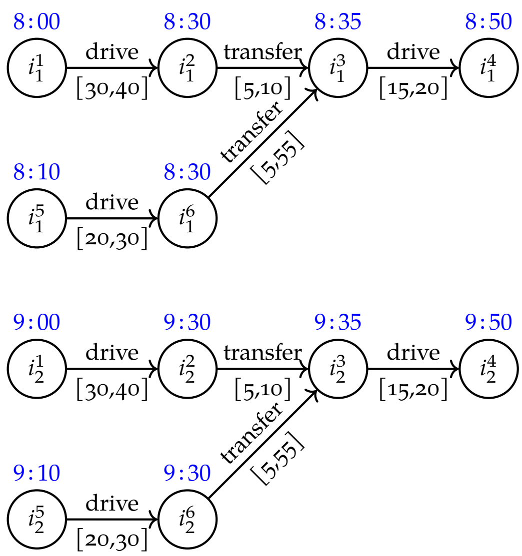
To summarise, we get back to the questions we posed at the beginning of the section. As we have seen in Example 4.15, an optimal solution (π, u, F) to PTTA$(\tilde{\pi})$ does not necessarily yield an optimal solution to PESP, if $\tilde{\pi}$ was not an optimal PESP solution in the first place. However, we could answer all the other questions positively. In particular, this means we can always solve PTTA instead of solving PESP – which will prove useful when integrating timetabling and delay management.



(a) PESP instance with timetable for $T = 60$.



(b) PTTA instance for the fixed timetable, $I = [8{:}00, 9{:}59]$, $(K = 2)$.



(c) PTTA instance for another timetable.

Figure 4.2: Rolling out a suboptimal timetable $\tilde{\pi}$ can yield a suboptimal timetable for PTTA($\tilde{\pi}$).

Line concept	$\vert\mathcal{L}\vert$	$\vert\mathcal{E}\vert$	$\vert\mathcal{A}\vert$	$\vert\mathcal{A}_{\text{drive}}\vert$	$\vert\mathcal{A}_{\text{wait}}\vert$	$\vert\mathcal{A}_{\text{transfer}}\vert$	$\vert\mathcal{A}_{\text{head}}\vert$
`line concept 1`	5	180	284	90	80	70	44
`line concept 2`	6	196	348	98	86	96	68
`line concept 3`	6	212	414	106	94	130	84

Table 4.1: Size of the periodic EAN for the used line concepts.

4.3 COMPUTATIONAL EXPERIMENTS

In this section, we test the performance of the new models when solving the IP formulations with Gurobi [Gur23] and compare them to solving PESP. We used the dataset `lowersaxony`, which contains data of the regional railway network in a region of Lower Saxony in northern Germany and has a size for which our integer programs can still be solved in reasonable time. The dataset is part of the open-source software framework LinTim, see [Sch+23; Sch+]. An overview of the PTN of this dataset is given in Figure A.1 in the appendix. We used LinTim to generate different line concepts and the resulting EANs. An overview of the number of lines $|\mathcal{L}|$ and the size of the EANs is given in Table 4.1. We solved PTTA2 and PTTA3 for different time horizons, varying the number of periods $K \in \{3, 4, \dots, 15\}$, observed the computing time and compared it to the computing time when solving PESP.

We implemented the IP models in Python and ran them on an Lenovo laptop with Intel(R) Core(TM) i5-10310U CPU @ 1.70 GHz, 2.21 GHz and 16 GB RAM using the solver Gurobi 9.1.1 [Gur23]. The results are shown in Figure 4.3.

We first note that, as expected due to the higher number of variables and the additional assignment constraints, for both versions of PTTA the solver takes much longer than for PESP. However, recall that our motivation was to integrate delay management – a task PESP is not suited for – so we do not have the aspiration to beat PESP when doing pure timetabling. Since PTTA3 only solves the assignment for the first period, while PTTA2 does this for all periods, one would expect it to be faster solvable than PTTA2. Indeed, we can see this behaviour in the instance `line concept 3`. For `line concept 2` both models perform quite similar. In the instance `line concept 1` we can observe that for larger K the computing time of PTTA2 increases more than for PTTA3, which can again be explained with PTTA3 only solving the assignment in the first period. The peak of PTTA3 at $K = 11$ is an exception. However, inspecting the progress of the solver shows that the optimal solution was actually found much earlier and the most part of the computing time was dedicated to proving optimality, so we treat this as a random outlier. The instance `line concept 3`, which is the largest one, shows the largest variance. Investigating the solving process shows that also here for both formulations the solver often has difficulties to determine that the incumbent solution is indeed optimal, a well known phenomenon for many integer problems. Thus, providing dual bounds has the potential to speed up the solving process significantly.

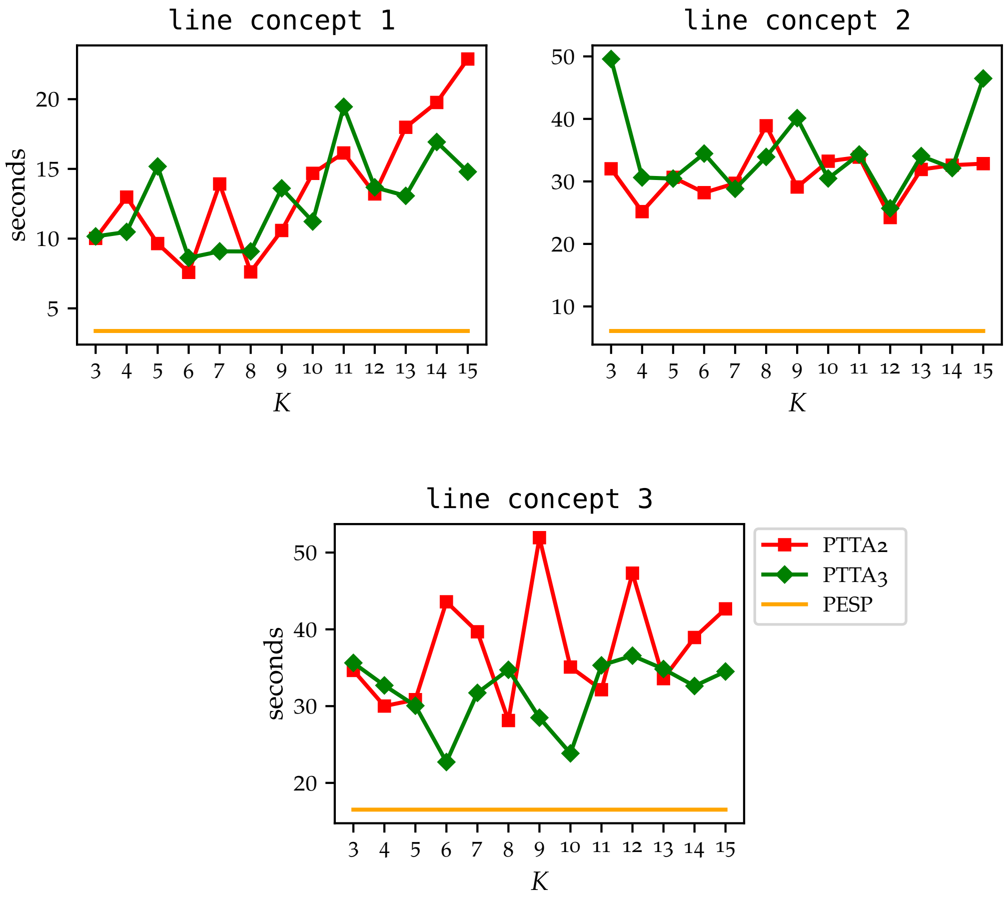


Figure 4.3: Average computing time (over four runs) for different line concepts with varying K.

CONCLUSION

We have developed a new model for periodic timetabling which uses a non-periodic network as a basis. We have shown that the new model is equivalent to PESP and that – although this was not our main focus – the achieved computing times are acceptable. We also derived a streamlined version which uses significantly less variables and constraints.

The new model opens many possibilities for future research. An obvious line of research is to strengthen its IP formulation, e.g. by using dual bounds, to speed up the solving process. A possible extension of our model could be to allow more flexibility in the periodicity constraints, e.g. to allow that the differences between repetitions of events are not exactly T but in some interval $[T - \epsilon, T + \epsilon]$. Our main interest, however, is to use the model for integration purposes. In the next chapter, we use the new aperiodic model for integrating timetabling and delay management in a two-stage model to achieve recoverable robust timetables.

5 MODELS FOR RECOVERABLE ROBUST PERIODIC TIMETABLING

After having established how to model periodic timetables in an aperiodic network by introducing PTTA in the previous chapter, we now formulate the *Recoverable Robust Periodic Timetabling Problem* (*RRPT*). The goal is to find a periodic timetable which can be recovered in every delay scenario $r \in \mathcal{U}$. Note that we have two contradictory objectives: nominal travel time and delay. Hence, we actually have a multi-objective problem. We consider different single-objective versions, concentrating on the different objectives. Furthermore, we develop several equivalent formulations for RRPT and look into their computational efficiency.

OUTLINE In Section 5.1 we introduce different variants of the Recoverable Robust Periodic Timetabling model, differing in their objective functions. Furthermore, we have a look at how to overcome some modelling difficulties. In Section 5.2 we give a first MIP formulation for the problem, which is based on the model PTTA from the previous chapter. Two other equivalent formulations are presented in Section 5.3. Finally, we compare the three formulations in Section 5.4.

5.1 APPLYING RECOVERABLE ROBUSTNESS TO TIMETABLING

As mentioned before, we use delay management as recovery algorithm. In particular, for the set $\mathcal{A}$ of admissible algorithms we actually choose only a single algorithm solving the delay management problem. Hence, for a nominal feasible timetable π and a scenario $r \in \mathcal{U}$ with source delays d^r (as defined in Chapter 3) the recovery algorithm yields a disposition timetable x^r with wait/no-wait decisions y^r. By doing so, we actually do not strictly stick to the definition of recoverable robustness given in Section 2.5: according to (2.18), we would require that the output $A(\pi, r)$ of the recovery algorithm is feasible for PESP(r). In particular, the disposition timetable would need to be periodic – which is not reasonable for our application. Furthermore, the constraints for cancelled transfers should be omitted. Hence, to be able to fully use the power of delay management, we have to relax the definition a bit. We hence formulate the constraints in the rolled out network (as done in PTTA), drop the periodicity constraints, the upper bounds and those constraints for cancelled transfers. For $a = (i, j) \in \mathcal{A}_{\text{train}}$ we define

$$\hat{g}_a^L(\pi, x^r, y^r) := \begin{cases} L_a + x_i^r - x_j^r & \text{if } u(\pi)_a = 1, \\ 0 & \text{otherwise,} \end{cases}$$

and for $a = (i, j) \in \mathcal{A}_{\text{transfer}}$

$$\hat{g}_a^L(\pi, x^r, y^r) := \begin{cases} L_a + x_i^r - x_j^r - M' y_a^r & \text{if } u(\pi)_a = 1, \\ 0 & \text{otherwise,} \end{cases}$$

where $u(\pi)$ contains the assignments corresponding to the timetable π and M' is a large constant. Furthermore, for $a = (i, j) \in \mathcal{A}_{\text{head}}$ we define

$$\hat{g}_a^L(\pi, x^r, y^r) := \begin{cases} L_a + x_i^r - x_j^r & \text{if } x_j^r - x_i^r \geq L_a, \\ 0 & \text{otherwise.} \end{cases}$$

For $r \in \mathcal{U}$ we then set

$$\hat{g}^L(\pi, x^r, y^r, r) := \begin{pmatrix} \left(\hat{g}_a^L(\pi, x^r, y^r) + d_a^r\right)_{a \in \mathcal{A}_{\text{train}}} \\ \left(\hat{g}_a^L(\pi, x^r, y^r)\right)_{a \in \mathcal{A}_{\text{transfer}} \cup \mathcal{A}_{\text{head}}} \end{pmatrix},$$

where $d_a^r := d_{\underline{a}_s}^r$ for $a = (i_s, j_t) \in \mathcal{A}(\underline{a}), \underline{a} \in \underline{\mathcal{A}}$, i.e. the activity a is a repetition of the periodic activity $\underline{a}$ in the s-th period. For the output $(x^r, y^r) = A(\pi, r)$ of the recovery algorithm we then require

$$\forall r \in \mathcal{U} : \hat{g}^L(\pi, x^r, y^r, r) \leq 0. \tag{5.1}$$

Note that these constraints are indeed fulfilled for a solution of the Delay Management Problem.

Recall from (2.20) one variant of the recoverable robust problem:

$$\min_{(x, A, \lambda) \in \mathcal{X} \times \mathbb{A} \times \Lambda} \left\{ \begin{pmatrix} f(x) \\ f'(\lambda) \end{pmatrix} : \forall r \in \mathcal{U} : g(A(x, r), r) \leq 0 \wedge c(A(x, r)) \leq \lambda \right\}.$$

As explained above, we replace g by $\hat{g}^L$. The function f is the objective function of the nominal problem. We still have to specify c and f'. For this purpose, we first introduce different evaluation functions of a timetable and its resulting delay.

Definition 5.1. Let a timetable π be given. Every $r \in \mathcal{U}$ yields an instance of DM, which we denote by DM(r), to which we can find a solution (x^r, y^r). Let

$$Z_1^r(\pi, x) := \sum_{i \in \mathcal{E}(\pi)} w_i (x_i^r - \pi_i)$$

be the *weighted event delay* and

$$Z_2^r(\pi, y) := \sum_{a \in \mathcal{A}_{\text{transfer}}(\pi)} w_a y_a^r$$

the *number of missed transfers* in scenario r. Furthermore, denote $Z_1(\pi, x) := \max_{r \in \mathcal{U}} Z_1^r(\pi, x)$ and $Z_2(\pi, y) := \max_{r \in \mathcal{U}} Z_2^r(\pi, y)$.

The *nominal travel time* of π, i.e. the travel time of all passengers in the undisturbed setting, is denoted by

$$f^{\text{nom}}(\pi) := \sum_{a=(i_1,j_t)\in\mathcal{A}_{\text{train}}(\pi)\cup\mathcal{A}_{\text{transfer}}(\pi)} w_a(\pi_{j_t} - \pi_{i_1}) \cdot K.$$

Note that $f^{\text{nom}}(\pi) = K \cdot f^{\text{PESP}}(\pi)$. Let (x^r, y^r) be an optimal solution to DM(r) for every $r \in \mathcal{U}$. We denote by

$$f^{\text{del}}(\pi) := \max_{r\in\mathcal{U}}(Z_1^r(\pi,x) + TZ_2^r(\pi,y))$$

the *worst-case delay* of π and by

$$f^{\text{real}}(\pi) := f^{\text{nom}}(\pi) + f^{\text{del}}(\pi)$$

the *real travel time* of π with respect to $\mathcal{U}$. The problem of minimising $f^{\text{del}}(\pi)$ for fixed π, i.e. solving DM(r) for all $r \in \mathcal{U}$, is called DM$(\mathcal{U})$.

As function c measuring the recovery costs we choose

$$c(\pi, x^r, y^r) = Z_1^r(\pi,x) + TZ_2^r(\pi,y),$$

i.e. the recovery is measured by a weighted sum of the weighted event delay and the number of missed transfers – which is also the objective function of DM. As the second objective f' we simply choose $f'(\lambda) = \lambda$. In particular, for an optimal solution we have

$$f'(\lambda) = \max_{r\in\mathcal{U}}(Z_1^r(\pi,x) + TZ_2^r(\pi,y)) = f^{\text{del}}(\pi).$$

Our goal can hence be formulated as follows:

MULTI-OBJECTIVE RECOVERABLE ROBUST PERIODIC TIMETABLING (MR-RPT)

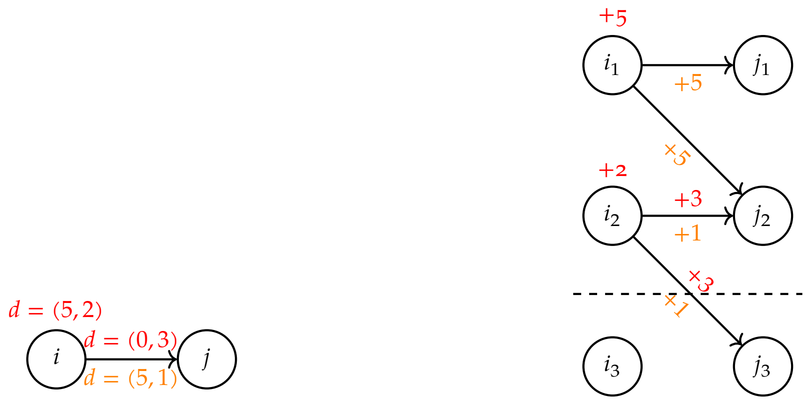
Input: Periodic EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$.

Task: Find a feasible periodic timetable π and disposition timetables x^r with wait/no-wait decisions y^r for every $r \in \mathcal{U}$ such that the nominal travel time $f^{\text{nom}}(\pi)$ and the worst-case delay $f^{\text{del}}(\pi)$ are minimal.

Recall from Chapter 3 that we can re-write event delays as activity delays. Hence, using this transformation we can also consider event delays in the model described above. While the delay transformation is shown for delay management in [Sch09], we will also prove it for the integrated model in Lemma 5.8.

The following example illustrates how the source delays in an instance of the recoverable robust periodic timetabling problem are given.

Example 5.2. We look at Figure 5.1, where the periodic EAN is given on the left. The events and activities have labels d indicating their source delays for every single period, in this case for $K = 2$ periods. We have two different delay scenarios, r_1 (in red) and r_2 (in orange). If a node or edge is without a label, it does not have any source delays. In



(a) Periodic EAN with aperiodic source delays.

(b) Rolled out EAN (without a timetable) together with the source delays.

Figure 5.1: Example for source delays.

scenario r_1, the event i has a source delay of 5 minutes in the first period and of 2 minutes in the second period, i.e. $d^{r_1}_{i_1} = 5$ and $d^{r_1}_{i_2} = 2$ in the rolled out EAN on the right. The activity (i, j) has no delay in the first period, but 3 minutes delay in the second period, i.e. $d^{r_1}_{(i_2,j_2)} = d^{r_1}_{(i_2,j_3)} = 3$. For the other scenario r_2 we have $d^{r_2}_{(i_1,j_1)} = d^{r_2}_{(i_1,j_2)} = 5$ and $d^{r_2}_{(i_2,j_2)} = d^{r_2}_{(i_2,j_3)} = 1$.

The problem has several objective functions: the nominal travel time and the worst-case delay, where the latter can be split into the weighted event delay and the number of missed transfers. However, since delving deeply into multi-objective optimisation is out of the scope of this thesis, we will mostly be using a weighted sum scalarisation:

RECOVERABLE ROBUST PERIODIC TIMETABLING (RRPT)

Input: Periodic EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$.

Task: Find a feasible periodic timetable π and disposition timetables x^r with wait/no-wait decisions y^r for every $r \in \mathcal{U}$ such that the real travel time f^{real} is minimal.

We denote an optimal timetable by π^{RR}. Concerning the computational complexity, it is not surprising that RRPT is hard.

Theorem 5.3. *RRPT is NP-hard.*

Proof. This follows easily from PESP being NP-hard. Let an instance of PESP with EAN $\underline{\mathcal{N}}$ and period length T be given. We construct an instance of RRPT without any source delays, leaving $\underline{\mathcal{N}}$ and T untouched. Then an optimal solution $(\tilde{\pi}, z)$ to PESP yields a feasible solution to RRPT by taking the rolled out timetable π and setting $x \equiv \pi$ and $y \equiv 0$. Since this solution does not have any delay, it is indeed optimal. Vice versa, let (π, x, y) be an optimal solution to RRPT. Then the corresponding timetable $(\tilde{\pi}, z)$ in the periodic network is an optimal solution to PESP since any better solution could be used to construct a better solution for RRPT without any delays. □

Instead of using a weighted sum scalarisation, we can also minimise only one objective function while bounding the others. Hence, we will also the consider the following versions of the problem:

- RRPT(α, β): For given $\alpha, \beta \geq 0$, minimise the nominal travel time f^{nom} such that the weighted event delay and the number of missed transfers are bounded by α and β, respectively, i.e. $Z_1(\pi, x) \leq \alpha, Z_2(\pi, y) \leq \beta$. We denote an optimal timetable by $\pi^{\text{RR},\alpha,\beta}$.
- RRPT$(\bar{f})$: For given $\bar{f}$, minimise the worst-case delay f^{del} such that the nominal travel time is bounded by $\bar{f}$, i.e. $f^{\text{nom}}(\pi) \leq \bar{f}$. We denote an optimal timetable by $\pi^{\text{RR},\bar{f}}$.

The problem definition above can easily be adapted for these different versions.

Note that there are different possibilities to handle the delay: in RRPT and RRPT$(\bar{f})$ we consider f^{del} to appropriately account for the delays the passengers face. Alternatively, one can also consider the weighted event delay and the number of missed transfers separately, which is also done in [Lie+09]. However, this can overestimate the real delay. Since both types of delay should not be too high, in RRPT(α, β) we have a bound for each of them.

Observation 5.4. Per definition we get the following inequalities:

(a) $f^{\text{real}}(\pi^{\text{RR}}) \leq f^{\text{real}}(\pi)$ for every feasible timetable π.

(b) $f^{\text{nom}}(\pi^{\text{RR},\alpha,\beta}) \leq f^{\text{nom}}(\pi)$ for every feasible timetable π with DM(r)-solutions $(x^r, y^r)_{r \in \mathcal{U}}$ such that $Z_1(\pi, x) \leq \alpha$ and $Z_2(\pi, y) \leq \beta$.

(c) $f^{\text{del}}(\pi^{\text{RR},\bar{f}}) \leq f^{\text{del}}(\pi)$ for every feasible timetable π with $f^{\text{nom}}(\pi) \leq \bar{f}$.

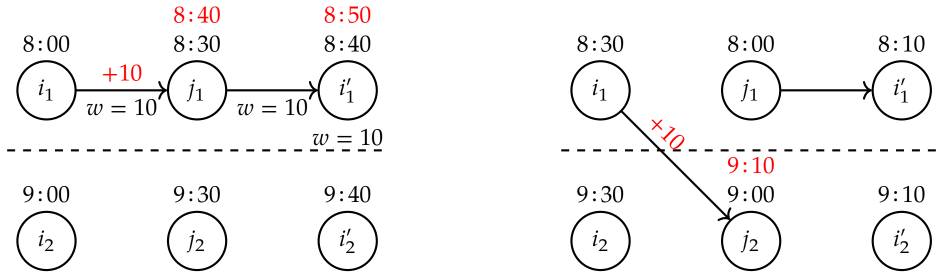
To derive MIP formulations for these problems, we can now use the preparatory work we did when introducing PTTA: since we have formulated the timetabling problem, which is a subproblem of RRPT, already in the aperiodic network, we can now simply add the delay management constraints for every scenario $r \in \mathcal{U}$. Of course we only have constraints for those arcs a which are actually chosen in the assignment subproblem of PTTA, i.e. those with $u_a = 1$. Hence, we have to add the delay propagation constraints as big-M-constraints. We choose

$$M' := \max_{r \in \mathcal{U}} \left(\max_{a \in \mathcal{A}} (L_a + d_a^r) + \max_{i \in \mathcal{E}} d_i^r + \sum_{a \in \mathcal{A}} d_a^r \right) + t_{\max} + \frac{T \cdot |\mathcal{A}_{\text{head}}|}{2}$$

and will see later in Lemma 5.7 that this constant is sufficiently large.

Another problem we have to deal with is the passengers leaving our planning horizon I, as the following example demonstrates.

Example 5.5. We consider a part of a rolled out EAN as depicted in Figure 5.2 for two different timetables and only a single delay scenario. In Figure 5.2a there are 10 passengers arriving at i'_1 with 10 minutes delay, so we have $Z_1(\pi, x) = 10$. However, if we shift the timetable by 30 minutes as seen in Figure 5.2b, different arcs are chosen, so we have the arc (i_1, j_2) leaving the planning horizon. In this case there is no delay at the event i'_1. Since the event j_2 which is delayed in this case has weight zero, the weighted event delay is



(a) All passengers arrive with a delay. (b) The passengers' delays are not counted.

Figure 5.2: The delay of passengers leaving the planning horizon is not counted correctly.

$Z_1(\pi, x) = 0$. The reason for this is that the passengers' delays are counted when they arrive at their final destination. With the shifted timetable, the arrival is outside of our planning horizon, so no delay is recognised by our objective function. To prevent this, we count the last known delay of those passengers leaving the planning horizon: in this case these are the 10 minutes delay at the event j_2, which we weight with the number of passengers using the arc (i_1, j_2).

Notation 5.6. We denote the rolled out driving, waiting and transfer activities leaving the planning horizon I by $\mathcal{A}_{\text{out}} := \{a = (i_s, j_t) \in \mathcal{A}_{\text{train}} \cup \mathcal{A}_{\text{transfer}} : t > K\}$ and adapt the definition of the weighted event delay:

$$Z_1^r(\pi, x) := \sum_{i \in \mathcal{E}(\pi)} w_i(x_i^r - \pi_i) + \sum_{a=(i,j) \in \mathcal{A}_{\text{out}}(\pi)} w_a(x_j^r - \pi_j).$$

We will also need big-M-constraints for the activities $\mathcal{A}_{\text{out}}$, so we set

$$M'' := \max_{r \in \mathcal{U}} \left(\max_{i \in \mathcal{E}} d_i^r + \sum_{a \in \mathcal{A}} d_a^r \right) + \frac{T \cdot |\mathcal{A}_{\text{head}}|}{2},$$

which is sufficiently large as we will see in Lemma 5.7.

Recall that while in periodic timetabling headway activities can be treated in the same way as driving, waiting and transfer activities, this is not the case for aperiodic timetabling and delay management. To enable us to change the order of trains in case of delays, we need precedence constraints between all pairs of events using the same piece of infrastructure. Additionally to the headways $\underline{\mathcal{A}}_{\text{head}}$ we now also need to respect headways between repetitions of the same periodic event: if the event i_s has a big delay, there can be a conflict with the next event i_{s+1}. Therefore, we define

$$\mathcal{A}'_{\text{head}} := \{(i_s, j_t) : (i,j) \in \underline{\mathcal{A}}_{\text{head}}, 1 \le s, t \le K\} \cup \{(i_s, i_t) : i \in \underline{\mathcal{E}}, 1 \le s, t \le K\}.$$

For an example see Figure 5.3. We will need these activities for the delay propagation constraints. Note that $\mathcal{A}'_{\text{head}}$ is not a subset of $\mathcal{A}$, since it also contains arcs of the form (i_s, j_t) for $t < s$ and $t > s + b$. This is due to the fact that delays can change the order of the events.

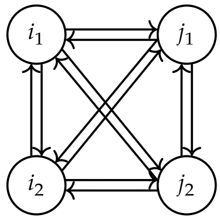


Figure 5.3: $\mathcal{A}'_{\text{head}}$ induced by $\underline{\mathcal{A}}_{\text{head}} = \{(i,j),(j,i)\}$.

5.2 FORMULATIONS USING PTTA

Now we can formulate the different recoverable robust problems from the previous section as MIPs. We use PTTA2 for the timetabling subproblem to simplify notation. To indicate that these formulations use the aperiodic network and to distinguish them from other formulations we will introduce later, we mark them with "-a". We start with the multi-objective version.

$$\min \sum_{a=(i_1,j_t)\in\mathcal{A}_{\text{train}}\cup\mathcal{A}_{\text{transfer}}} w_a F_a \cdot K \tag{MRRPT-a}$$

$$\min \quad Z$$

$$\text{s.t.} \quad \pi_j - \pi_i + M(u_a - 1) \le U_a \qquad a=(i,j)\in\mathcal{A} \tag{5.2}$$

$$\pi_j - \pi_i + M(1-u_a) \ge L_a \qquad a=(i,j)\in\mathcal{A} \tag{5.3}$$

$$\pi_{i_s} - \pi_{i_{s-1}} = T \qquad i_s\in\mathcal{E}, 2\le s\le K+b \tag{5.4}$$

$$\sum_{a'=(i_s,j_t)\in\mathcal{A}} u_{a'} = 1 \qquad (i,j)\in\underline{\mathcal{A}}, 1\le s\le K \tag{5.5}$$

$$F_a \ge M(u_a-1) + \pi_{j_t} - \pi_{i_1} \qquad a=(i_1,j_t)\in\mathcal{A}_{\text{train}}\cup\mathcal{A}_{\text{transfer}} \tag{5.6}$$

$$\pi_i \ge t_{\min} \qquad i\in\mathcal{E} \tag{5.7}$$

$$\pi_{i_1} \le t_{\min} + T - 1 \qquad i\in\underline{\mathcal{E}} \tag{5.8}$$

$$x_i^r \ge \pi_i + d_i^r \qquad i\in\mathcal{E}, r\in\mathcal{U} \tag{5.9}$$

$$M'(1-u_a) + x_j^r - x_i^r \ge L_a + d_a^r \qquad a=(i,j)\in\mathcal{A}_{\text{train}}, r\in\mathcal{U} \tag{5.10}$$

$$M'(1-u_a) + M'y_a^r + x_j^r - x_i^r \ge L_a \qquad a=(i,j)\in\mathcal{A}_{\text{transfer}}, r\in\mathcal{U} \tag{5.11}$$

$$M'(1-p_{ij}^r) + x_j^r - x_i^r \ge L_a \qquad a=(i,j)\in\mathcal{A}'_{\text{head}}, r\in\mathcal{U} \tag{5.12}$$

$$p_{ij}^r + p_{ji}^r = 1 \qquad (i,j),(j,i)\in\mathcal{A}'_{\text{head}}, r\in\mathcal{U} \tag{5.13}$$

$$\sum_{a\in\mathcal{A}_{\text{transfer}}} w_a y_a^r \le Z_2^r \qquad r\in\mathcal{U} \tag{5.14}$$

$$\sum_{i_s\in\mathcal{E}:s\le K} w_{i_s}(x_{i_s}^r - \pi_{i_s}) + \sum_{a\in\mathcal{A}_{\text{out}}} w_a H_a^r \le Z_1^r \qquad r\in\mathcal{U} \tag{5.15}$$

$$Z_1^r + TZ_2^r \le Z \qquad r\in\mathcal{U} \tag{5.16}$$

$$H_a^r \ge M''(u_a-1) + x_j^r - \pi_j \qquad a=(i,j)\in\mathcal{A}_{\text{out}}, r\in\mathcal{U} \tag{5.17}$$

$$\pi_i\in\mathbb{N} \qquad i\in\mathcal{E} \tag{5.18}$$

$$x_i^r\in\mathbb{N} \qquad i\in\mathcal{E}, r\in\mathcal{U} \tag{5.19}$$

$$y_a^r \in \{0,1\} \qquad a \in \mathcal{A}_{\text{transfer}}, r \in \mathcal{U} \tag{5.20}$$

$$u_a \in \{0,1\} \qquad a \in \mathcal{A} \tag{5.21}$$

$$F_a \geq 0 \qquad a = (i_1, j_t) \in \mathcal{A}_{\text{train}} \cup \mathcal{A}_{\text{transfer}} \tag{5.22}$$

$$p_{ij}^r \in \{0,1\} \qquad (i,j) \in \mathcal{A}'_{\text{head}}, r \in \mathcal{U} \tag{5.23}$$

$$H_a^r \geq 0 \qquad a \in \mathcal{A}_{\text{out}}, r \in \mathcal{U} \tag{5.24}$$

$$Z_1^r, Z_2^r \geq 0 \qquad r \in \mathcal{U} \tag{5.25}$$

$$Z \geq 0. \tag{5.26}$$

As mentioned before, the objective functions are the nominal travel time (i.e. the objective function of PTTA), and the worst-case delay. Constraints (5.2) to (5.8) are the same as in PTTA2. The subsequent constraints are the constraints from DM adapted to our needs: (5.9) ensure that for every delay scenario and every event the time in the disposition timetable is not earlier than in the original timetable and the event delays are respected. Constraints (5.10) make sure that the delays are propagated along the driving and waiting activities for those arcs a fulfilling $u_a = 1$. Similarly, the delay propagation along maintained transfers is ensured by (5.11). The delay propagation along headway constraints is handled by (5.12), while (5.13) makes sure that for each pair of headways exactly one is chosen. The number of missed transfers and the weighted event delay are counted by (5.14) and (5.15), respectively, and the worst-case delay is determined in Constraints (5.16). Note that for the weighted event delay we count the weighted delay of every event within the planning horizon (i.e. those i_s with $s \leq K$) as well as the weighted delay of the arcs $\mathcal{A}_{\text{out}}$ leaving the planning horizon. To ensure that only those arcs in $\mathcal{A}_{\text{out}}$ with $u_a = 1$ are respected here, we need another set of big-M-constraints given in (5.17).

Lemma 5.7. *M' and M'' are sufficiently large.*

Proof. Let (π, u, F) be a feasible solution to the subproblem PTTA . We have to show for every $r \in \mathcal{U}$ there is an optimal solution (x^r, y^r) to DM(r) fulfilling the following constraints:

- $L_a + d_a^r - x_j^r + x_i^r \leq M'$ for $a = (i,j) \in \mathcal{A}_{\text{train}}, r \in \mathcal{U}$
- $L_a - x_j^r + x_i^r \leq M'$ for $a = (i,j) \in \mathcal{A}_{\text{transfer}} \cup \mathcal{A}'_{\text{head}}, r \in \mathcal{U}$
- $x_j^r - \pi_j - H_a^r \leq M''$ for $a = (i,j) \in \mathcal{A}_{\text{out}}, r \in \mathcal{U}$.

For DM(r) it is known (see [Sch09]) that there is an optimal solution (x^r, y^r) fulfilling

$$x_i^r - \pi_i \leq \max_{l \in \mathcal{E}(\pi)} d_l^r + \sum_{a \in \mathcal{A}(\pi)} d_a^r + \sum_{\substack{a=(l,j) \in \mathcal{A}_{\text{head}}(\pi): \\ \pi_l > \pi_j}} (\pi_l - \pi_j + L_a) \tag{5.27}$$

for all $i \in \mathcal{E}(\pi)$. We consider the last term in (5.27):

$$\sum_{\substack{a=(l,j) \in \mathcal{A}_{\text{head}}(\pi): \\ \pi_l > \pi_j}} (\underbrace{\pi_l - \pi_j}_{\leq U_{(l,j)} = T - h_{lj}} + \underbrace{L_a}_{= h_{lj}}) \leq \sum_{\substack{a=(l,j) \in \mathcal{A}_{\text{head}}(\pi): \\ \pi_l > \pi_j}} T \leq \frac{T \cdot |\mathcal{A}_{\text{head}}|}{2}.$$

Hence, for $(i, j) \in \mathcal{A}_{\text{out}}$ we obtain

$$x_j^r - \pi_j - H_a^r \overset{H_a^r \geq 0}{\leq} x_j^r - \pi_j \leq \max_{l \in \mathcal{E}(\pi)} d_l^r + \sum_{a \in \mathcal{A}(\pi)} d_a^r + \frac{T \cdot |\mathcal{A}_{\text{head}}|}{2} \leq M'', \tag{5.28}$$

which shows the last inequality. Furthermore, for $a = (i, j) \in \mathcal{A}_{\text{train}}$ we have

$$\begin{aligned} & L_a + d_a^r - x_j^r + x_i^r \\ \leq\ & L_a + d_a^r + x_i^r \\ \overset{(5.28)}{\leq}\ & L_a + d_a^r + \pi_i + \max_{i \in \mathcal{E}(\pi)} d_i^r + \sum_{a' \in \mathcal{A}(\pi)} d_{a'}^r + \frac{T \cdot |\mathcal{A}_{\text{head}}|}{2} \\ \leq\ & L_a + d_a^r + t_{\max} + \max_{i \in \mathcal{E}(\pi)} d_i^r + \sum_{a' \in \mathcal{A}(\pi)} d_{a'}^r + \frac{T \cdot |\mathcal{A}_{\text{head}}|}{2} \\ \leq\ & M', \end{aligned}$$

showing the first inequality. The second one follows in the same manner. □

Adapting MRRPT-a to obtain formulations for the other versions is straightforward. For the weighted sum scalarisation we only have to adjust the objective function:

$$\begin{aligned} \min \quad & f^{\text{real}} = \sum_{a \in \mathcal{A}_{\text{train}} \cup \mathcal{A}_{\text{transfer}}} w_a F_a \cdot K + Z \\ \text{s.t.} \quad & (5.2) - (5.26). \end{aligned} \tag{RRPT-a}$$

If we bound the delays and only minimise the nominal travel time we get:

$$\begin{aligned} \min \quad & f^{\text{nom}} = \sum_{a \in \mathcal{A}_{\text{train}} \cup \mathcal{A}_{\text{transfer}}} w_a F_a \cdot K && && \text{(RRPT-a}(\alpha, \beta)) \\ \text{s.t.} \quad & (5.2) - (5.13) \\ & (5.17) - (5.24) \\ & \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a^r \leq \beta && r \in \mathcal{U} && (5.29) \\ & \sum_{i_s \in \mathcal{E}: s \leq K} w_{i_s} (x_{i_s}^r - \pi_{i_s}) + \sum_{a \in \mathcal{A}_{\text{out}}} w_a H_a^r \leq \alpha && r \in \mathcal{U}. && (5.30) \end{aligned}$$

Finally, we can also bound the nominal travel time and minimise the delays:

$$\begin{aligned} \min \quad & f^{\text{del}} = Z && \text{(RRPT-a}(\bar{f})) \\ \text{s.t.} \quad & (5.2) - (5.26) \\ & \sum_{a \in \mathcal{A}_{\text{train}} \cup \mathcal{A}_{\text{transfer}}} w_a F_a \cdot K \leq \bar{f}. && (5.31) \end{aligned}$$

It remains to show that the delay transformation, transforming an event delay into an activity delay by considering an extended network, does not change the problem, so we can indeed handle the event delays.

Lemma 5.8. *Using the transformation described in Chapter 3, every optimal solution for an instance of RRPT is still optimal for the transformed instance and vice versa.*

Proof. First, note that for the new activity $\underline{a}^0 = (i^0, i)$ we have $U_{\underline{a}^0} = 0$ and hence also $b_{\underline{a}^0} = 0$, meaning that we only have one possible rolled out activity in every period and thus can set $u_{(i_s^0, i_s)} = 1$ for all $s \in \{1, \dots, K\}$.

Let $(\pi, u, F, x, y, Z_1, Z_2, Z, H, p)$ be a feasible solution to RRPT-a (with event delays). We set $x^r_{i_s^0} = \pi_{i_s^0} = \pi_{i_s}$ for all $s \in \{1, \dots, K\}$ and $F_{(i_s^0, i_s)} = 0$. Since $\pi_{i_s^0} = \pi_{i_s}$ and $U_{\underline{a}_0} = L_{\underline{a}_0} = 0$, the timetabling constraints (5.2) to (5.8) follow immediately. Due to $\tilde{d}^r_{i_s} = 0$ also (5.9) follows. We treat the virtual activities as those in $\mathcal{A}_{\text{train}}$. We have $x^r_{i_s} - x^r_{i_s^0} = x^r_{i_s} - \pi_{i_s} \geq d^r_{i_s} = \tilde{d}^r_{\underline{a}_s^0}$, so also (5.10) is fulfilled. Since $x^r_{i_s^0} = \pi_{i_s^0}$, also the worst-case delay does not change and the other constraints are not affected by the transformation, so the constructed solution is feasible for the transformed instance with the same objective value.

Vice versa, if we have a solution for the larger instance without event delays, restricting it to the smaller instance with event delays gives a feasible solution with the same objective value.

This holds for all considered variants of the problem. □

5.3 FORMULATIONS USING PESP

In this section we will derive two formulations equivalent to RRPT-a. All results can be transferred to the other problem variants.

We assume $\frac{t_{\min}}{T} \in \mathbb{Z}$. If the period length is, for example, one hour, this means the planning horizon starts at the full hour.

5.3.1 *Event-based formulation*

So far we have used the PTTA constraints and added delay management constraints to obtain a formulation for RRPT. An alternative approach is to use the regular PESP constraints and use our knowledge from the development of the model PTTA to retrieve the assignment variables u from the modulo variables z. Indeed, as we have seen in Lemma 4.8, if we have a feasible solution $(\tilde{\pi}, z)$ to PESP, setting

$$u_a = \begin{cases} 1 & \text{if } t = z_{\underline{a}} + k - l + s, \\ 0 & \text{otherwise,} \end{cases}$$

for $\underline{a} = (i, j) \in \underline{\mathcal{A}}, a = (i_s, j_t)$, where k, l such that $\tilde{\pi}_i + kT := \min\{\tilde{\pi}_i + cT : \tilde{\pi}_i + cT \geq t_{\min}, c \in \mathbb{Z}\}$ and $\tilde{\pi}_j + lT := \min\{\tilde{\pi}_j + cT : \tilde{\pi}_j + cT \geq t_{\min}, c \in \mathbb{Z}\}$, yields a feasible PTTA solution. Since we assume $\frac{t_{\min}}{T} \in \mathbb{Z}$, it follows $k = l$, so we set $u_a = 1$ if and only if $t = z_{\underline{a}} + s$. To formulate this as linear constraints, we again need big-M-constraints. Fortunately, for the big-M we can choose b, which is usually quite small (≈ 2).

This yields the following formulation, which we denote by RRPT-pe, since, in contrast to the previous formulation, it uses only the periodic events.:

$$
\begin{aligned}
\min \quad & f^{\text{real}} = \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T)\cdot K + Z && && \text{(RRPT-pe)}\\
\text{s.t.} \quad & \tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T \le U_{\underline{a}} && \underline{a}=(i,j)\in\underline{\mathcal{A}} && (5.32)\\
& \tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T \ge L_{\underline{a}} && \underline{a}=(i,j)\in\underline{\mathcal{A}} && (5.33)\\
& \pi_{i_s} - \tilde{\pi}_i = t_{\min} + (s-1)T && i\in\underline{\mathcal{E}}, 1\le s\le K+b && (5.34)\\
& b(1-u_a) + t - s - z_{\underline{a}} \ge 0 && \underline{a}\in\underline{\mathcal{A}}, a=(i_s,j_t)\in\mathcal{A}(\underline{a}) && (5.35)\\
& b(u_a - 1) + t - s - z_{\underline{a}} \le 0 && \underline{a}\in\underline{\mathcal{A}}, a=(i_s,j_t)\in\mathcal{A}(\underline{a}) && (5.36)\\
& (5.5) && &&\\
& (5.9) - (5.21) && &&\\
& (5.23) - (5.26) && &&\\
& \tilde{\pi}_i \in \mathbb{N} && i\in\underline{\mathcal{E}} && (5.37)\\
& 0 \le \tilde{\pi}_i \le T-1 && i\in\underline{\mathcal{E}} && (5.38)\\
& z_{\underline{a}} \in \mathbb{Z} && \underline{a}\in\underline{\mathcal{A}}. && (5.39)
\end{aligned}
$$

The objective function minimises the weighted sum of the nominal travel time and the worst-case delay. Constraints (5.32) and (5.33) are the constraints from PESP. Constraints (5.34) ensure that the times for the rolled out events are set correctly. As explained above, the values of the modulo variables already determine the values of the assignment variables. This relation is accounted for in (5.35) and (5.36). The other constraints are taken from our previous formulation RRPT-a.

We will now show that this formulation is actually equivalent to RRPT-a. First we need the following lemma.

Lemma 5.9. *Let (π, z) be a feasible solution to PESP. Then $0 \le z_{\underline{a}} \le b$ for all $\underline{a} \in \underline{\mathcal{A}}$.*

Proof. By feasibility of (π, z) we have

$$1 - T \overset{(2.4)}{\le} \pi_i - \pi_j \le \pi_i - \pi_j + L_{\underline{a}} \overset{(2.3)}{\le} z_{\underline{a}}T \overset{(2.2)}{\le} \pi_i - \pi_j + U_{\underline{a}} \overset{(2.4)}{\le} T - 1 + U_{\underline{a}}$$

for $\underline{a} = (i,j) \in \underline{\mathcal{A}}$ and hence $\frac{1}{T} - 1 \le z_{\underline{a}} \le \frac{T-1}{T} + \frac{U_{\underline{a}}}{T}$, which by integrality of $z_{\underline{a}}$ implies

$$0 = \left\lceil \frac{1}{T} - 1 \right\rceil \le z_{\underline{a}} \le \left\lfloor \frac{T-1}{T} + \frac{U_{\underline{a}}}{T} \right\rfloor \le \left\lceil \frac{U_{\underline{a}}}{T} \right\rceil \le b. \qquad \square$$

Theorem 5.10. *RRPT-a and RRPT-pe are equivalent.*

Proof. Let $(\pi, u, F, x, y, Z_1, Z_2, Z, H, p)$ be a solution to RRPT-a. Let $\underline{a} = (i,j) \in \underline{\mathcal{A}}$. Choose the unique t such that $u_{(i_1,j_t)} = 1$, which exists due to (5.5), and define $z_{\underline{a}} := t - 1$. For $i \in \underline{\mathcal{E}}$ set $\tilde{\pi}_i := \pi_{i_1} - t_{\min}$. Note that since $t_{\min} \le \pi_{i_1} \le t_{\min} + T - 1$, it follows $0 \le \tilde{\pi}_i \le T - 1$. We now show that $(\tilde{\pi}, z, \pi, u, x, y, Z_1, Z_2, Z, H, p)$ is feasible for RRPT-pe with the same objective value.

- $\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T = (\pi_{j_1} - t_{\min}) - (\pi_{i_1} - t_{\min}) + (t-1)T = \pi_{j_1} - \pi_{i_1} + (t-1)T = \pi_{j_t} - \pi_{i_1} \in [L_{\underline{a}}, U_{\underline{a}}]$ by choice of t and (5.2) and (5.3), which shows (5.32) and (5.33).
- Let $i \in \underline{\mathcal{E}}, 1 \leq s \leq K + b$. We have $\pi_{i_s} \overset{(5.4)}{=} \pi_{i_1} + (s-1)T = \tilde{\pi}_i + t_{\min} + (s-1)T$, so (5.34) is satisfied.
- Let $\underline{a} \in \underline{\mathcal{A}}, a = (i_s, j_t) \in \mathcal{A}(\underline{a})$. We know from Lemma 4.5 that $u_{(i_s,j_t)} = u_{(i_1,j_{t-s-1})}$. Hence, if $u_a = 1$, then also $u_{(i_1,j_{t-s-1})} = 1$, so by definition $z_{\underline{a}} = t-s$. For the other case, note that by construction of $\mathcal{A}$ we have $0 \leq t - s \leq b$. Furthermore, by Lemma 5.9 we know $0 \leq z_{\underline{a}} \leq b$. Hence, it follows $b + t - s - z_{\underline{a}} \geq 0$ and $-b + t - s - z_{\underline{a}} \leq 0$, which implies that (5.35) and (5.36) are fulfilled.
- All other constraints are clearly fulfilled since they are the same in both formulations.

Furthermore, as seen above, $\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T = \pi_{j_t} - \pi_{i_1} \leq F_a$, with t such that $u_{(i_1,j_t)} = 1$, so the objective value of the constructed solution is not higher than that of the RRPT-a solution.

Now, let a solution $(\tilde{\pi}, z, \pi, u, x, y, Z_1, Z_2, Z, H, p)$ to RRPT-pe be given. In particular, $(\tilde{\pi}, z)$ is a solution to PESP. By constraints (5.5), (5.35) and (5.36) we know that $u_{(i_s,j_t)} = 1$ if and only if $t = s + z_{\underline{a}}$. If we additionally set

$$F_a = \begin{cases} \pi_{j_t} - \pi_{i_1} & \text{if } u_{(i_1,j_t)} = 1, \\ 0 & \text{otherwise,} \end{cases}$$

we know from Lemma 4.8 that (π, u, F) is feasible for PTTA2, i.e. (5.2)-(5.8) are fulfilled. Since all other constraints are fulfilled as well, $(\pi, u, F, x, y, Z_1, Z_2, Z, H, p)$ is feasible for RRPT-a. The equality of the objective function values also follows from Lemma 4.8. □

Remark 5.11. We note that there are two different intuitive possibilities to define $\tilde{\pi}_i$ when π_{i_1} is given:

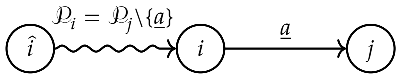
(a) $\tilde{\pi}_i = \pi_{i_1} - t_{\min}$

(b) $\tilde{\pi}_i = \pi_{i_1} \bmod T$.

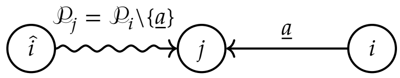
Under the assumption $\frac{t_{\min}}{T} \in \mathbb{Z}$, these are equivalent, but they differ when we drop this assumption. For example, if $T = 60$ and $t_{\min} = 510$, i.e. the planning horizon starts at 8:30, and we have $\pi_{i_1} = 550 \mathrel{\hat{=}} 9{:}10$, then $\pi_{i_1} - t_{\min} = 40$ but $\pi_{i_1} \bmod T = 10$. This is relevant for constraints (5.34): $\pi_{i_s} - \tilde{\pi}_i = t_{\min} + (s-1)T$, which is fulfilled using the first definition. However, if we use the second one, this is not true any more. In the example we have $\pi_{i_1} - \tilde{\pi}_i = 550 - 10 = 540 \neq 550 = t_{\min} + (1-1)T$. In general, the following holds in case (b):

Assume $\frac{t_{\min}}{T} \notin \mathbb{Z}$, i.e. $t_{\min} = q_1 T + \omega$ for $q_1 \in \mathbb{Z}, \omega \in \{1, \ldots, T-1\}$. Furthermore, using definition (b), we have $\pi_{i_1} = q_2 T + \tilde{\pi}_i$ for some $q_2 \in \mathbb{Z}$, where q_2 is minimal such that $\pi_{i_1} \geq t_{\min}$ due to constraints (5.8). Hence, if $\tilde{\pi}_i \geq \omega$, we have $q_1 = q_2$ and therefore $\pi_{i_1} - \tilde{\pi}_i = t_{\min} - \omega$. Otherwise, it holds $q_2 = q_1 + 1$ and hence $\pi_{i_1} - \tilde{\pi}_i = t_{\min} - \omega + T$.

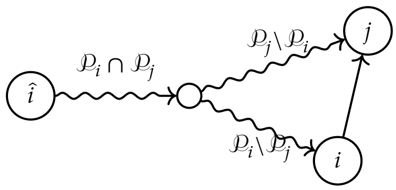
Of course it is possible to adapt constraints (5.34) accordingly. However, to obtain a linear formulation, we would need to introduce further big-M-constraints, which makes



(a) Case $(i,j) \in \mathcal{T}$ with $\mathcal{P}_j = \mathcal{P}_i \cup \{\underline{a}\}$.



(b) Case $(i,j) \in \mathcal{T}$ with $\mathcal{P}_j = \mathcal{P}_i \backslash \{\underline{a}\}$.



(c) Case $(i,j) \notin \mathcal{T}$.

Figure 5.4: Paths $\mathcal{P}_i$ and $\mathcal{P}_j$ in (5.40) and (5.41).

the formulation more complicated. Hence, at this point it makes more sense to use the first definition. However, the second definition is also used in the literature on rolling out timetables, see [Lie+10], and might be considered more intuitive.
Since $\frac{t_{\min}}{T} \in \mathbb{Z}$ seems reasonable, we use this assumption. Hence, both definitions are equivalent and we do not face the problem explained above.

5.3.2 *Cycle-based formulation*

Apart from the standard PESP formulation, we can also use the cycle basis formulation as stated in Section 2.2. However, since we need the times of the events, and these are not present in the cycle basis formulation, we have to extract them from the tensions. We first need to introduce some notation.

Notation 5.12. Let $\mathcal{T}$ be a spanning tree in $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ and $\hat{\imath} \in \underline{\mathcal{E}}$ some fixed event. For $i \in \mathcal{E}$ let $\mathcal{P}_i$ be the unique path from $\hat{\imath}$ to i in $\mathcal{T}$. The set of arcs in $\mathcal{P}_i$ can be partitioned into the sets $\mathcal{P}_i^+$ and $\mathcal{P}_i^-$ of forward and backward arcs.

Note that for $\underline{a} = (i,j) \in \mathcal{T}$ it holds either

$$\mathcal{P}_j^+ = \mathcal{P}_i^+ \cup \{\underline{a}\},\ \mathcal{P}_j^- = \mathcal{P}_i^- \text{ or } \mathcal{P}_j^+ = \mathcal{P}_i^+,\ \mathcal{P}_j^- = \mathcal{P}_i^- \backslash \{\underline{a}\}. \tag{5.40}$$

On the other hand, for $\underline{a} = (i,j) \in \underline{\mathcal{A}} \backslash \mathcal{T}$ we have

$$C_{\underline{a}}^+ = (\mathcal{P}_i^+ \backslash \mathcal{P}_j) \cup (\mathcal{P}_j^- \backslash \mathcal{P}_i) \cup \{\underline{a}\} \text{ and } C_{\underline{a}}^- = (\mathcal{P}_j^+ \backslash \mathcal{P}_i) \cup (\mathcal{P}_i^- \backslash \mathcal{P}_j). \tag{5.41}$$

An illustration is given in Figure 5.4.

Now, if we have a feasible solution to PESP-cb given by (ξ, q), we can use it to obtain the time for every event i by adding respectively subtracting the tensions along the path $\mathcal{P}_i$:

$$\tilde{\pi}_i = \sum_{\underline{a} \in \mathcal{P}_i^+} \xi_{\underline{a}} - \sum_{\underline{a} \in \mathcal{P}_i^-} \xi_{\underline{a}} + \tilde{\pi}_{\hat{\imath}} + \hat{q}_i T$$

for some $\hat{q}_i \in \mathbb{Z}$ such that $\tilde{\pi}_i \in \{0, \ldots, T-1\}$.

We can use q and $\hat{q}$ to obtain the values of z in the formulation RRPT-pe, namely for $\underline{a} = (i, j) \in \underline{\mathcal{A}}$ we have

$$z_{\underline{a}} = \begin{cases} \hat{q}_i - \hat{q}_j & \text{if } \underline{a} \in \mathcal{T}, \\ \hat{q}_i - \hat{q}_j + q_{\underline{a}} & \text{if } \underline{a} \notin \mathcal{T}, \end{cases}$$

as we will see in the proof of Theorem 5.13. Then, as before, we also know the values of the assignment variables u, which leads to the following IP formulation:

$$\min \quad f^{\text{real}} = \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}} \xi_{\underline{a}} \cdot K + Z \qquad \text{(RRPT-cb)}$$

$$\text{s.t.} \quad \Gamma \xi = T q \qquad (5.42)$$

$$\tilde{\pi}_i = \sum_{\underline{a}\in\mathcal{P}_i^+} \xi_{\underline{a}} - \sum_{\underline{a}\in\mathcal{P}_i^-} \xi_{\underline{a}} + \tilde{\pi}_{\hat{\imath}} + \hat{q}_i T \qquad i \in \underline{\mathcal{E}} \qquad (5.43)$$

(5.34)

$$b(1-u_a) + t - s - \hat{q}_i + \hat{q}_j \geq 0 \qquad \underline{a} \in \mathcal{T}, a = (i_s, j_t) \in \mathcal{A}(\underline{a}) \qquad (5.44)$$

$$b(u_a - 1) + t - s - \hat{q}_i + \hat{q}_j \leq 0 \qquad \underline{a} \in \mathcal{T}, a = (i_s, j_t) \in \mathcal{A}(\underline{a}) \qquad (5.45)$$

$$b(1-u_a) + t - s - \hat{q}_i + \hat{q}_j - q_{\underline{a}} \geq 0 \qquad \underline{a} \in \underline{\mathcal{A}} \backslash \mathcal{T}, a = (i_s, j_t) \in \mathcal{A}(\underline{a}) \qquad (5.46)$$

$$b(u_a - 1) + t - s - \hat{q}_i + \hat{q}_j - q_{\underline{a}} \leq 0 \qquad \underline{a} \in \underline{\mathcal{A}} \backslash \mathcal{T}, a = (i_s, j_t) \in \mathcal{A}(\underline{a}) \qquad (5.47)$$

(5.5)

(5.9) – (5.21)

(5.23) – (5.26)

(5.37) – (5.38)

$$\xi_{\underline{a}} \in \mathbb{N} \qquad \underline{a} \in \underline{\mathcal{A}} \qquad (5.48)$$

$$L_{\underline{a}} \leq \xi_{\underline{a}} \leq U_{\underline{a}} \qquad \underline{a} \in \underline{\mathcal{A}} \qquad (5.49)$$

$$q_{\underline{a}} \in \mathbb{Z} \qquad \underline{a} \in \underline{\mathcal{A}} \backslash \mathcal{T} \qquad (5.50)$$

$$\hat{q}_i \in \mathbb{Z} \qquad i \in \underline{\mathcal{E}}. \qquad (5.51)$$

The objective function minimises the weighted sum of the nominal travel time and the worst-case delay. Constraint (5.42) ensures that ξ is indeed a periodic tension (as in PESP-cb). Constraints (5.43) construct the event times from the tensions as explained above. The correspondence between $u, q, \hat{q}$ is respected in Constraints (5.44) to (5.47). The other constraints are the same as in the formulation of RRPT-pe.

Theorem 5.13. *RRPT-pe and RRPT-cb are equivalent.*

Proof. Let $(\tilde{\pi}, z, \pi, u, x, y, Z_1, Z_2, Z, H, p)$ be a solution to RRPT-pe. In particular, $(\tilde{\pi}, z)$ is a solution to PESP.

- From the literature on PESP, see [Pee03], we know that by setting $\xi_{\underline{a}} := \tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}} T$ for $\underline{a} = (i, j) \in \underline{\mathcal{A}}$ and $q_{\underline{a}} := \sum_{\underline{a}'\in C_{\underline{a}}^+} z_{\underline{a}'} - \sum_{\underline{a}'\in C_{\underline{a}}^-} z_{\underline{a}'}$ we obtain a periodic tension ξ such that $\Gamma\xi = Tq$, i.e. (5.42) holds.

- We define $\hat{q}_j := \sum_{\underline{a}\in\mathcal{P}_j^-} z_{\underline{a}} - \sum_{\underline{a}\in\mathcal{P}_j^+} z_{\underline{a}}$. By definition of ξ and induction on the length of the unique path $\mathcal{P}_j$ from $\hat{\imath}$ to j in $\mathcal{T}$ we obtain

$$\tilde{\pi}_j = \tilde{\pi}_{\hat{\imath}} + \sum_{\underline{a}\in\mathcal{P}_j^+} (\xi_{\underline{a}} - z_{\underline{a}}T) + \sum_{\underline{a}\in\mathcal{P}_j^-} (-\xi_{\underline{a}} + z_{\underline{a}}T) = \tilde{\pi}_{\hat{\imath}} + \sum_{\underline{a}\in\mathcal{P}_j^+} \xi_{\underline{a}} - \sum_{\underline{a}\in\mathcal{P}_j^-} \xi_{\underline{a}} + \hat{q}_j T.$$

  See Figure 5.5 for an illustration. This shows (5.43).

- For $\underline{a} = (i,j) \in \mathcal{T}$ by (5.40) it follows $\hat{q}_i - \hat{q}_j = z_{\underline{a}}$, so by (5.35) and (5.36) also (5.44) and (5.45) are fulfilled.

- For $\underline{a} = (i,j) \notin \mathcal{T}$ we get

$$\begin{aligned}
&\hat{q}_i - \hat{q}_j + q_{\underline{a}} \\
&= \left(\sum_{\underline{a}'\in\mathcal{P}_i^-} z_{\underline{a}'} - \sum_{\underline{a}'\in\mathcal{P}_i^+} z_{\underline{a}'}\right) - \left(\sum_{\underline{a}'\in\mathcal{P}_j^-} z_{\underline{a}'} - \sum_{\underline{a}'\in\mathcal{P}_j^+} z_{\underline{a}'}\right) + \left(\sum_{\underline{a}'\in C_{\underline{a}}^+} z_{\underline{a}'} - \sum_{\underline{a}'\in C_{\underline{a}}^-} z_{\underline{a}'}\right) \\
&\overset{(*)}{=} \sum_{\underline{a}'\in\mathcal{P}_i^-\cup\mathcal{P}_j^+\cup C_{\underline{a}}^+} z_{\underline{a}'} - \sum_{\underline{a}'\in\mathcal{P}_i^+\cup\mathcal{P}_j^-\cup C_{\underline{a}}^-} z_{\underline{a}'} \\
&\overset{(**)}{=} z_{\underline{a}},
\end{aligned}$$

  where $(*)$ follows from the fact that the sets we unite are pairwise disjoint and $(**)$ follows from

$$\begin{aligned}
&\mathcal{P}_i^- \cup \mathcal{P}_j^+ \cup C_{\underline{a}}^+ \\
&\overset{(5.41)}{=} \mathcal{P}_i^- \cup \mathcal{P}_j^+ \cup (\mathcal{P}_i^+ \setminus \mathcal{P}_j) \cup (\mathcal{P}_j^- \setminus \mathcal{P}_i) \cup \{\underline{a}\} \\
&= (\mathcal{P}_i^- \setminus \mathcal{P}_j) \cup (\mathcal{P}_i^- \cap \mathcal{P}_j) \cup (\mathcal{P}_j^+ \setminus \mathcal{P}_i) \cup (\mathcal{P}_j^+ \cap \mathcal{P}_i) \cup (\mathcal{P}_i^+ \setminus \mathcal{P}_j) \cup (\mathcal{P}_j^- \setminus \mathcal{P}_i) \cup \{\underline{a}\} \\
&= (\mathcal{P}_i^- \setminus \mathcal{P}_j) \cup (\mathcal{P}_i \cap \mathcal{P}_j^-) \cup (\mathcal{P}_j^+ \setminus \mathcal{P}_i) \cup (\mathcal{P}_j \cap \mathcal{P}_i^+) \cup (\mathcal{P}_i^+ \setminus \mathcal{P}_j) \cup (\mathcal{P}_j^- \setminus \mathcal{P}_i) \cup \{\underline{a}\} \\
&= \mathcal{P}_i^+ \cup \mathcal{P}_j^- \cup (\mathcal{P}_j^+ \setminus \mathcal{P}_i) \cup (\mathcal{P}_i^- \setminus \mathcal{P}_j) \cup \{\underline{a}\} \\
&\overset{(5.41)}{=} \mathcal{P}_i^+ \cup \mathcal{P}_j^- \cup C_{\underline{a}}^- \cup \{\underline{a}\}.
\end{aligned}$$

  Hence, also (5.46) and (5.47) are satisfied.

- Constraints (5.48) to (5.51) are trivially fulfilled.

On the other hand, let $(\xi, \tilde{\pi}, q, \hat{q}, \pi, u, x, y, Z_1, Z_2, Z, H, p)$ be a solution to RRPT-cb. For $\underline{a} = (i,j) \in \underline{\mathcal{A}}$ we define

$$z_{\underline{a}} := \begin{cases} \hat{q}_i - \hat{q}_j & \text{if } \underline{a} \in \mathcal{T}, \\ \hat{q}_i - \hat{q}_j + q_{\underline{a}} & \text{if } \underline{a} \notin \mathcal{T}. \end{cases}$$

- For $\underline{a} = (i,j) \in \underline{\mathcal{A}}$ by (5.43) we have

$$\tilde{\pi}_j - \tilde{\pi}_i = \left(\sum_{\underline{a}'\in\mathcal{P}_j^+} \xi_{\underline{a}'} - \sum_{\underline{a}'\in\mathcal{P}_j^-} \xi_{\underline{a}'} + \tilde{\pi}_{\hat{\imath}} + \hat{q}_j T\right) - \left(\sum_{\underline{a}'\in\mathcal{P}_i^+} \xi_{\underline{a}'} - \sum_{\underline{a}'\in\mathcal{P}_i^-} \xi_{\underline{a}'} + \tilde{\pi}_{\hat{\imath}} + \hat{q}_i T\right).$$

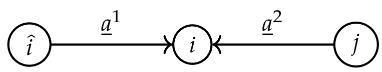
- For the case $\underline{a} \in \mathcal{T}$ this term simplifies to $\xi_{\underline{a}} + (\hat{q}_j - \hat{q}_i)T = \xi_{\underline{a}} - z_{\underline{a}}T$.
- If $\underline{a} \notin \mathcal{T}$, the term above is equal to

$$\begin{aligned}\sum_{\underline{a}' \in C^-_{\underline{a}}} \xi_{\underline{a}'} - \left(\sum_{\underline{a}' \in C^+_{\underline{a}}} \xi_{\underline{a}'} - \xi_{\underline{a}} \right) + (\hat{q}_j - \hat{q}_i)T &= \xi_{\underline{a}} - (\Gamma\xi)_{\underline{a}} + (\hat{q}_j - \hat{q}_i)T \\ &= \xi_{\underline{a}} - (q_{\underline{a}} - \hat{q}_j + \hat{q}_i)T = \xi_{\underline{a}} - z_{\underline{a}}T.\end{aligned}$$

Hence, in both cases we have $\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}T = \xi_{\underline{a}} \in [L_{\underline{a}}, U_{\underline{a}}]$, which shows (5.32) and (5.33).

- Constraints (5.35) and (5.36) are satisfied by definition of z and constraints (5.44) to (5.47).

Overall, we have shown that from each feasible solution to one of the problems we can construct a feasible solution to the other problem with the same objective function value. □



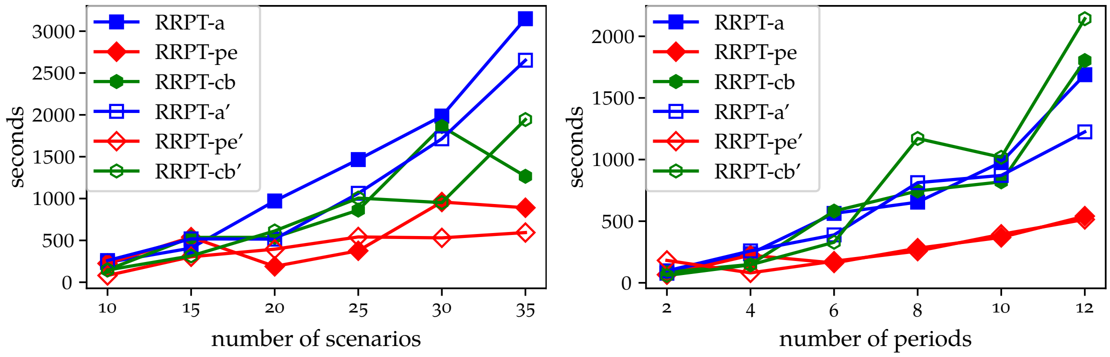
$$\tilde{\pi}_j - \tilde{\pi}_{\hat{i}} = (\tilde{\pi}_j - \tilde{\pi}_i) + (\tilde{\pi}_i - \tilde{\pi}_{\hat{i}}) = (-\xi_{\underline{a}^2} + z_{\underline{a}^2}T) + (\xi_{\underline{a}^1} - z_{\underline{a}^1}T)$$

Figure 5.5: The event times can be retrieved from the tension.

Recall that for PTTA2 we derived the reduced version PTTA3 by noticing that the assignment variables are periodic. We can apply this also for the different versions of RRPT, so for each formulation we obtain an equivalent reduced version with a smaller number of variables. Of course all proofs above can analogously be done for the reduced version, but for simpler notation we used the full version. As already seen for PTTA, we will see in the next section that also for RRPT the reduction does not have a considerable effect on the computing times when solving the IP.

5.4 COMPUTATIONAL EXPERIMENTS

In the previous sections, we derived three equivalent formulations of the recoverable robust periodic timetabling problem. An obvious question is which of these formulations is best. To answer this, we run some experiments and compare their computing times for solving the MIP. However, RRPT is a very hard problem: PESP and DM both are NP-complete [SU89; Gat+05] and RRPT integrates PESP and several delay management problems. Therefore, we are not able to solve RRPT on any large instances. For the experiments we thus used a rather small artificial network from the LinTim library [Sch+23; Sch+] with 156 periodic events and 188 periodic activities without headway constraints, which we call `toy`. An overview of the dataset is given in Figure A.2 and Table A.1 in the appendix. The period length is 60 minutes. For the delay scenarios we generated uniformly distributed source delays: in every scenario we generated a source delay between 1 and 15 minutes for 1% of all *aperiodic* events and activities.



(a) Computing times for $K = 4$ with an increasing number of scenarios $|\mathcal{U}|$.

(b) Computing times for $|\mathcal{U}| = 10$ with an increasing number of periods K.

Figure 5.6: Computing times for solving the MIP formulations of RRPT.

We implemented the MIP formulations in Python and solved them using Gurobi 8.1.1 [Gur23] on a compute server with 48 cores @2.9 GHz and 196 GB RAM. The MIP optimality gap of the solver was set to 0.015%.

We ran two different experiments: one where the number K of periods is fixed to 4 with a varying number of scenarios $|\mathcal{U}|$, and one with $|\mathcal{U}| = 10$ and a varying number of periods. These numbers are quite small, which is due to the high complexity of the problem. However, the experiments provide some insights on the performance of the different formulations. For every formulation we also tested its reduced version with less variables, as we did for PTTA when considering the formulations PTTA2 and PTTA3, which is indicated by an apostrophe behind the name.

The results are shown in Figure 5.6. For the first experiment we can see that RRPT-a has the highest computing times. This is not surprising, since it is based on PTTA, which is slower than PESP for the pure timetabling problem, as we have seen in Chapter 4. The lowest computing times are achieved for RRPT-pe. Since the cycle basis formulation PESP-cb is faster than the standard formulation when only looking for a timetable, this is a bit surprising. However, RRPT-cb not only uses variables for the tensions, but additionally also for the event times, since they are needed for the delay management part. This could be an explanation for the worse performance. For RRPT-a the variable reduction yields a small improvement. For the other formulations the variable reduction does not have a significant effect on the computing time.

For the second experiment, we have similar results. RRPT-pe performs best, while for RRPT-a and RRPT-cb the computing times become much larger with increasing K.

CONCLUSION

We have introduced the Recoverable Robust Periodic Timetabling Problem, which is the first to apply the concept of recoverable robustness to periodic timetabling with aperiodic source delays. We have considered several versions of the problem differing in the focus they put on the different objectives. Furthermore, we have developed three equivalent formulations based on different ways to incorporate the timetabling subproblem. We have compared these formulations with respect to their computing time when solving them with a state-of-the-art solver, showing that – as opposed to the pure timetabling problem – a cycle-base approach is not the best choice.

Due to the high complexity of the problem, the IP formulation is only able to handle rather small instances. Hence, developing heuristic approaches for solving the problem will be the focus of Chapter 7.

6 PROPERTIES OF RECOVERABLE ROBUST TIMETABLES

In this chapter we investigate some properties of recoverable robust timetables.

OUTLINE First, we will see that some intuitive properties of PESP do not hold any more when considering recoverable robust timetables. Then, we will have a closer look at the parameters α, β in RRPT(α, β) and see how we can reduce the number of scenarios if we use an always-wait policy, i.e. do not allow any cancelled transfers. Finally, we will say something about which activities are a good choice for placing buffer times.

An important property of timetables is that they can be shifted by a constant without changing the travel time. Note that due to the aperiodicity of the source delays this does not hold true for the delay, as the following example demonstrates. In particular, for recoverable robust timetables shifting does indeed make a difference.

Example 6.1. We consider the instance given in Figure 6.1a. An optimal solution is given in Figure 6.1b: it has a nominal travel time of 303 and 20 minutes total delay. The source delay of 5 minutes at event i'_2 actually does not have any effect, since the 10 minutes delay from event i_1 spread to i'_2 making the constraint respecting the source delay of 5 minutes redundant. However, if we shift the timetable by 19 minutes, we obtain the timetable given in Figure 6.1c. Using this timetable, the total delay increases to 30. This is due to the chosen arcs in the rolled out network being different. As a result, the two source delays do not "meet" and the delay propagation is different. Hence, we can see that shifting a timetable by a constant can increase its delay.

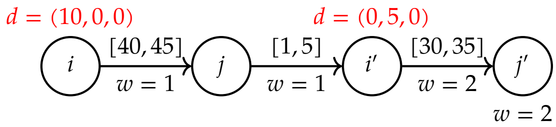
Recall that for PESP we can assume without loss of generality that

$$0 \le L_{\underline{a}} \le T - 1 \text{ and } L_{\underline{a}} \le U_{\underline{a}} \le L_{\underline{a}} + T - 1 \text{ for all } \underline{a} \in \underline{\mathcal{A}} \tag{6.1}$$

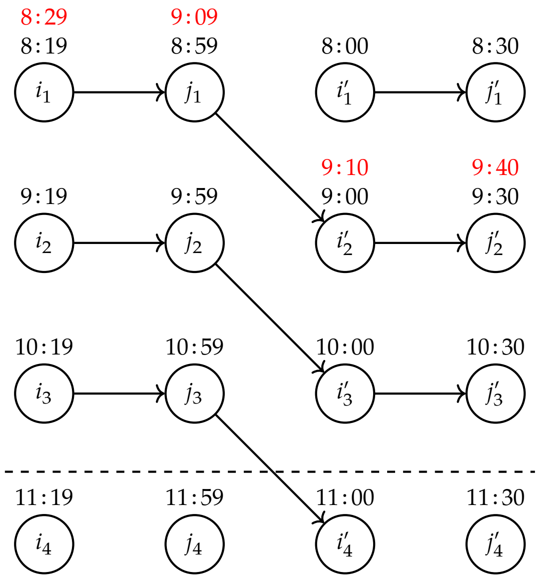
since otherwise we can subtract multiples of T such that this holds without changing the problem. However, for RRPT this is actually not the case, as can be seen in the following example.

Example 6.2. We consider Figure 6.2, with the original instance given in Figure 6.2a. The activity (j, i') does not fulfil (6.1). An optimal solution is given in 6.2b. If we subtract T from the lower and upper bound of activity (j, i'), i.e. $[L_{(j,i')}, U_{(j,i')}] = [1, 5]$, an optimal solution is given in Figure 6.2c. While the nominal travel time changes by $K \cdot T = 180$ as expected, the delay does increase, because with $U_{j,i'}$ also b changes. This results in a different delay propagation, which leads to the same phenomenon as in Example 6.1.

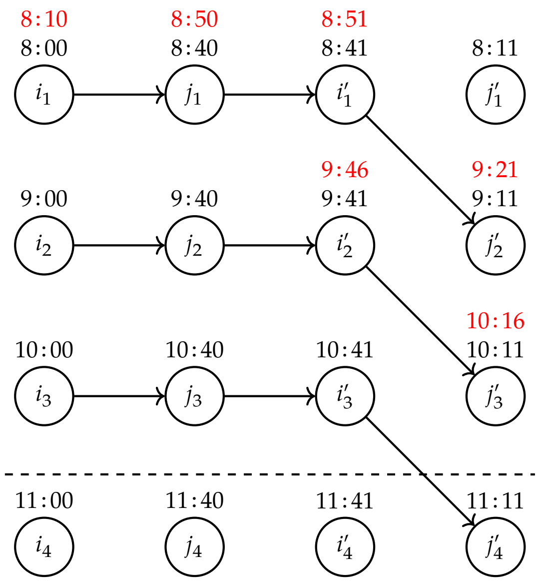
We now have a closer look at the influence of the parameters α and β in the problem variant RRPT(α, β) with bounded delay.



(a) Instance, $I = [8{:}00, 10{:}59]$, $T = 60$, $(K = 3)$.

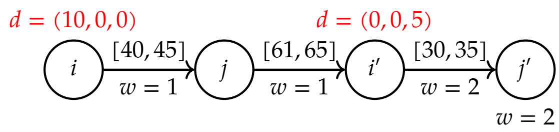


(b) Optimal solution to RRPT. $f^{\text{nom}}(\pi^{\text{RR}}) = 303, f^{\text{del}}(\pi^{\text{RR}}) = 20$.

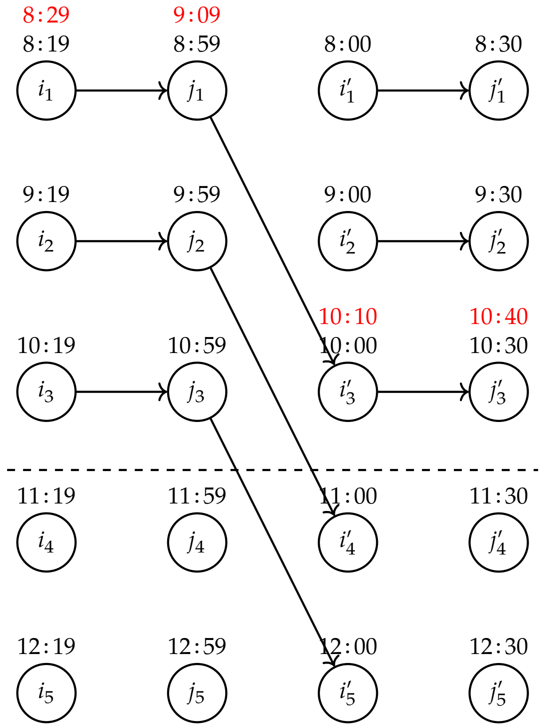


(c) Optimal solution to RRPT when $\pi_{i_1} = 8{:}00$ is fixed. $f^{\text{nom}}(\pi) = 303, f^{\text{del}}(\pi) = 30$.

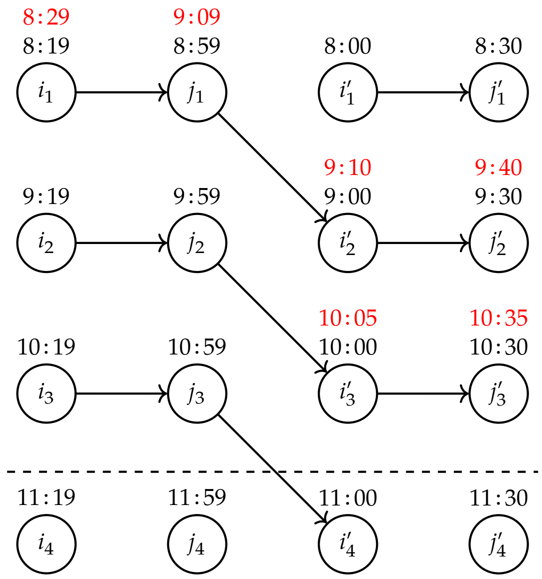
Figure 6.1: Shifting a timetable can increase its delay.



(a) Instance, $I = [8{:}00, 10{:}59]$, $T = 60$, $(K = 3)$.



(b) Optimal solution to RRPT. $f^{\text{nom}}(\pi^{\text{RR}}) = 483, f^{\text{del}}(\pi^{\text{RR}}) = 20$.



(c) Optimal solution to RRPT for the same instance with reduced bounds. $f^{\text{nom}}(\pi^{\text{RR}}) = 303, f^{\text{del}}(\pi^{\text{RR}}) = 30$.

Figure 6.2: Subtracting multiples of T from the lower and upper bounds changes the problem.

Definition 6.3. We are given a scenario set $\mathcal{U}$. A timetable π is called (α, β)-recoverable-robust w.r.t. $\mathcal{U}$ if it is feasible for RRPT(α, β), i.e. for every $r \in \mathcal{U}$ there is a disposition timetable such that the total delay and the number of missed transfers (both weighted with the number of passengers) are bounded from above by α respectively β.

Lemma 6.4. *Let π be an optimal (α, β)-recoverable-robust timetable and π' an optimal (α', β')-recoverable-robust timetable with $\alpha' \geq \alpha$ and $\beta' \geq \beta$. Then $f^{\text{nom}}(\pi') \leq f^{\text{nom}}(\pi)$.*

Proof. π is (α, β)-recoverable-robust. Since $\alpha' \geq \alpha, \beta' \geq \beta$, π is also (α', β')-recoverable-robust. By optimality of π' it follows $f^{\text{nom}}(\pi') \leq f^{\text{nom}}(\pi)$. □

For the rest of this chapter we assume $\mathcal{A}_{\text{head}} = \emptyset$. This is for example a reasonable assumption in a bus network. Furthermore, note that we assume $w_a > 0$ for all $a \in \mathcal{A}_{\text{transfer}}$, because we can delete all transfers which are not used by any passengers.

For strict robustness it is known that a robust feasible solution remains robust feasible if the uncertainty set is extended to its convex hull and hence it is sufficient to only consider the extreme points of $\mathcal{U}$ [BEN09]. For recoverable robustness this is not true in general, as shown in [CGS17]. In the following, we investigate what effect the extension to the convex hull has on RRPT(α, β) and identify a special case in which it indeed suffices to only solve the problem for the extreme points of $\mathcal{U}$.

Note that we slightly abuse notation here: when writing conv$(\mathcal{U})$ we actually mean $\{\bar{r} : d^{\bar{r}} \in \text{conv}(\{d^r : r \in \mathcal{U}\})\}$.

Theorem 6.5. *If π is (α, β)-recoverable-robust w.r.t. $\mathcal{U}$, then it is (α, β')-recoverable-robust w.r.t.* conv$(\mathcal{U})$ *with $\beta' = \sum_{\substack{a \in \mathcal{A}_{\text{transfer}}: \\ \exists r \in \mathcal{U}: y^r_a = 1}} w_a$ if the integrality constraint on x is relaxed.*

Proof. Since π is (α, β)-recoverable-robust w.r.t. $\mathcal{U}$, for every $r \in \mathcal{U}$ there is a DM(r)-solution (x^r, y^r) with delay bounded by α and missed transfers bounded by β. In particular, for every $r \in \mathcal{U}$ and for all $a = (i, j) \in \mathcal{A}^r_{\text{fix}}(\pi) := \mathcal{A}_{\text{train}}(\pi) \cup \{a' \in \mathcal{A}_{\text{transfer}}(\pi) : y^r_{a'} = 0\}$ we have $x^r_j - x^r_i \geq L_a + d^r_a$. Recall that only activities in $\mathcal{A}_{\text{train}}$ have source delays, so for simpler notation we set $d^r_a := 0$ for $a \in \mathcal{A}_{\text{transfer}}$. Let $\bar{r} \in \text{conv}(\mathcal{U})$, i.e. all source delays are of the form $d^{\bar{r}}_a = \sum_{r \in \mathcal{U}} \lambda_r d^r_a$ for $a \in \mathcal{A}_{\text{train}}$ respectively $d^{\bar{r}}_i = \sum_{r \in \mathcal{U}} \lambda_r d^r_i$ for $i \in \mathcal{E}$ with $\sum_{r \in \mathcal{U}} \lambda_r = 1$. For every $i \in \mathcal{E}$ define $x^{\bar{r}}_i := \sum_{r \in \mathcal{U}} \lambda_r x^r_i$ and set $y^{\bar{r}}_a := \max_{r \in \mathcal{U}} y^r_a$ for all $a \in \mathcal{A}_{\text{transfer}}$. It follows for $a = (i, j) \in \mathcal{A}^{\bar{r}}_{\text{fix}}(\pi)$:

$$x^{\bar{r}}_j - x^{\bar{r}}_i = \sum_{r \in \mathcal{U}} \lambda_r (x^r_j - x^r_i) \geq \sum_{r \in \mathcal{U}} \lambda_r (L_a + d^r_a) = \sum_{r \in \mathcal{U}} \lambda_r L_a + \sum_{r \in \mathcal{U}} \lambda_r d^r_a = L_a + d^{\bar{r}}_a. \qquad (6.2)$$

Analogously, we obtain $x^{\bar{r}}_i \geq \pi_i + d^{\bar{r}}_i$ for $i \in \mathcal{E}$. The weighted event delay is

$$\begin{aligned}
&\sum_{i_s \in \mathcal{E}: s \leq K} w_{i_s} (x^{\bar{r}}_{i_s} - \pi_{i_s}) + \sum_{a=(i,j) \in \mathcal{A}_{\text{out}}} w_a (x^{\bar{r}}_j - \pi_j) \\
&= \sum_{i_s \in \mathcal{E}: s \leq K} w_{i_s} \left(\sum_{r \in \mathcal{U}} \lambda_r x^r_{i_s} - \pi_{i_s} \right) + \sum_{a=(i,j) \in \mathcal{A}_{\text{out}}} w_a \left(\sum_{r \in \mathcal{U}} \lambda_r x^r_j - \pi_j \right) \\
&= \sum_{r \in \mathcal{U}} \lambda_r \underbrace{\left(\sum_{i_s \in \mathcal{E}: s \leq K} w_{i_s} (x^r_{i_s} - \pi_{i_s}) + \sum_{a=(i,j) \in \mathcal{A}_{\text{out}}} w_a (x^r_j - \pi_j) \right)}_{=Z^r_1(\pi, x)} \leq \max_{r \in \mathcal{U}} Z^r_1(\pi, x) \leq \alpha
\end{aligned} \qquad (6.3)$$

and the weighted number of missed transfers is

$$\sum_{a \in \mathcal{A}_{\text{transfer}}} w_a y_a^{\bar{r}} = \sum_{\substack{a \in \mathcal{A}_{\text{transfer}}: \\ y_a^{\bar{r}}=1}} w_a = \sum_{\substack{a \in \mathcal{A}_{\text{transfer}}: \\ \exists r \in \mathcal{U}: y_a^r=1}} w_a = \beta'. \qquad \square$$

Corollary 6.6. *If π is $(\alpha, 0)$-recoverable-robust w.r.t. $\mathcal{U}$, then it is also $(\alpha, 0)$-recoverable-robust w.r.t.* $\text{conv}(\mathcal{U})$ *if the integrality constraint on x is relaxed.*

Proof. This follows immediately from Theorem 6.5 since $\beta = 0$ implies $y_a^r = 0$ for all $a \in \mathcal{A}_{\text{transfer}}, r \in \mathcal{U}$ and hence also $\beta' = 0$. $\square$

Corollary 6.7. *For $\beta = 0$ and integral source delays it is sufficient to solve RRPT(α, β) for the extreme points of $\mathcal{U}$.*

Proof. Let $\mathcal{U}'$ be the set of extreme points of $\mathcal{U}$, i.e. $\mathcal{U} = \text{conv}(\mathcal{U}')$. Let π be an optimal timetable w.r.t. $\mathcal{U}'$, i.e. π is $(\alpha, 0)$-recoverable-robust w.r.t. $\mathcal{U}'$ such that the nominal travel time $f^{\text{nom}}(\pi)$ is minimal. By Corollary 6.6 π is also $(\alpha, 0)$-recoverable-robust w.r.t. $\mathcal{U}$ if the integrality constraint on x is relaxed, i.e. for all $r \in \mathcal{U}$ there is a solution $(x^r, 0)$ with weighted event delay $Z_1^r(\pi, x) \leq \alpha$. Due to the DM-constraints we have for all $j \in \mathcal{E}$:

$$x_j^r \geq \max\{\pi_j + d_j^r, \max_{a=(i,j) \in \mathcal{A}_{\text{fix}}^r(\pi)} x_i^r + L_a + d_a^r\}.$$

This inequality is fulfilled with equality in an optimal solution. Since π, L, d are integral, it also follows that x^r is integral. Hence, π is $(\alpha, 0)$-recoverable-robust w.r.t. $\mathcal{U}$.

We now show that π is also optimal w.r.t. $\mathcal{U}$. Assume there is an $(\alpha, 0)$-recoverable-robust solution π' with $f^{\text{nom}}(\pi') < f^{\text{nom}}(\pi)$. Since $\mathcal{U}' \subseteq \mathcal{U}$, π' is also $(\alpha, 0)$-recoverable-robust w.r.t. $\mathcal{U}'$, which is a contradiction to the optimality of π. $\square$

However, if we add headway activities to our EAN, this nice property does not hold any more, as can be seen in the following example.

Example 6.8. We consider the EAN in Figure 6.3. The given timetable is (4,0)-recoverable-robust w.r.t. both the red and the orange scenario. However, for the violet scenario, which is contained in the convex hull, we get a weighted event delay of 6 when not cancelling any transfers. To see this, note that without a delay, the events i_1^3 and i_1^7 would take place one minute apart from each other. Due to the delay propagated to them on the two delayed activities in the violet scenario, they would take place at the same time – which is forbidden due to the headways. Hence, one of the events has to be delayed even further, which increases the total delay.

Next we want to get an idea of which activities should have slack in the timetable and which should not.

Theorem 6.9. *Let $\underline{a} = (i, j) \in \underline{\mathcal{A}}$ with ingoing activities $\delta^-(i) = \{\underline{a}^1, \ldots, \underline{a}^B\}$ such that for the passenger weights we have $\sum_{k=1}^B w_{\underline{a}^k} < w_{\underline{a}}$ and no source delays on any node in $\mathcal{E}(i)$ or any arc in $\mathcal{A}(\underline{a})$ in any scenario. Furthermore, let the $(K+1)$-th repetition of i not be reachable from any source delay. Let (π, u, F, x, y, H) be an optimal solution to RRPT(α, β) and $(\tilde{\pi}, \tilde{z})$ the corresponding timetable in the periodic network. If $\Delta_{\underline{a}^k} := \tilde{\pi}_i - \tilde{\pi}_{i^k} + \tilde{z}_{\underline{a}^k} T - L_{\underline{a}^k} < U_{\underline{a}^k} - L_{\underline{a}^k}$ for all $k \in \{1, \ldots, B\}$, we have $\Delta_{\underline{a}} := \tilde{\pi}_j - \tilde{\pi}_i + \tilde{z}_{\underline{a}} T - L_{\underline{a}} = 0$, i.e. it is better to put slack on $\underline{a}^1, \ldots, \underline{a}^B$ than on $\underline{a}$.*

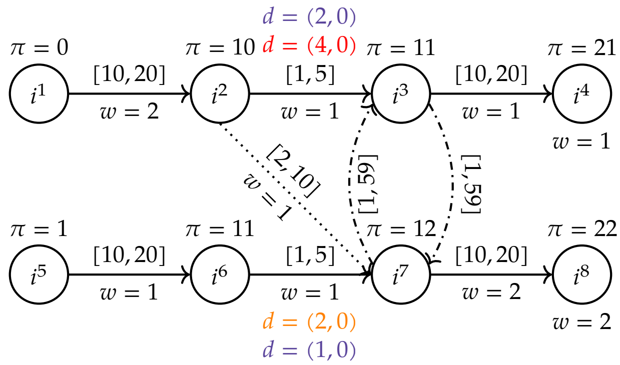


Figure 6.3: Theorem 6.5 does not hold if the EAN has headways ($T = 60$, $K = 2$).

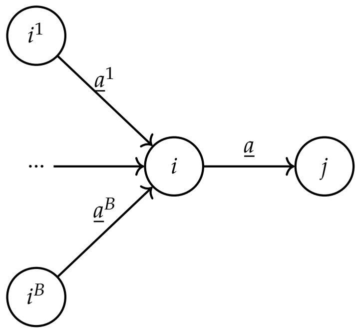


Figure 6.4: Subgraph in Theorem 6.9.

Proof. Assume there is an optimal solution with $\Delta_{\underline{a}^k} < U_{\underline{a}^k} - L_{\underline{a}^k}$ for all $k \in \{1, \dots, B\}$ and $\Delta_{\underline{a}} > 0$. Let $\varepsilon := \min\{\min_{k=1\dots,B}\{U_{\underline{a}^k} - L_{\underline{a}^k} - \Delta_{\underline{a}^k}\}, \Delta_{\underline{a}}, T\} > 0$. We construct a new solution

$$\tilde{\pi}'_l = \begin{cases} \tilde{\pi}_i + \varepsilon - zT & \text{if } l = i, \\ \tilde{\pi}_l & \text{otherwise,} \end{cases}$$

where $z \in \{0, 1\}$ such that $\tilde{\pi}'_i \in \{0, \dots, T-1\}$. Let π' be the corresponding rolled out timetable with assignment u'. By choice of ε, π' is a feasible timetable and for the nominal travel times f, f' of π respectively π' we obtain

$$f' = f + K \cdot \varepsilon \left(\sum_{k=1}^{B} w_{\underline{a}^k} - w_{\underline{a}} \right) < f.$$

It remains to show that π' is still (α, β)-recoverable-robust. For all nodes $l \notin \mathcal{E}(i)$ we set $x'^r_l := x^r_l$ for all $r \in \mathcal{U}$. Furthermore, let $y' := y$. We have to show constraints (5.9) to (5.11) for the subgraph in Figure 6.4.

First case: $z = 0$. In this case $u' := u$. For $1 \leq s \leq K+b$ and $r \in \mathcal{U}$ we set $x'^r_{i_s} := \max\{x^r_{i_s}, \pi'_{i_s}\}$. Constraints (5.9) are clearly fulfilled since by assumption the nodes i_s do not have a source delay. For i^k, $k \in \{1, \dots, B\}$, let $1 \leq s \leq K, s \leq t \leq K+b$ such that $u'_{(i^k_s, i_t)} = 1$. Then it follows $x'^r_{i_t} - x'^r_{i^k_s} \geq x^r_{i_t} - x^r_{i^k_s} \geq L_{\underline{a}^k} + d^r_{(i^k_s, i_t)}$ by feasibility of x. For s, t such that $u'_{(i_s, j_t)} = 1$,

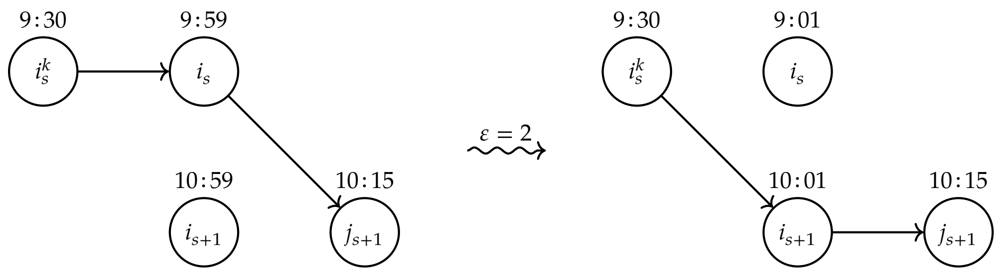


Figure 6.5: Example for the shift in the assignment variables in the proof of Theorem 6.9.

depending on for which value the maximum in the definition of $x'^r_{i_s}$ is attained, we either have

$$x'^r_{j_t} - x'^r_{i_s} = x^r_{j_t} - x^r_{i_s} \geq L_{\underline{a}}$$

or

$$\begin{aligned} x'^r_{j_t} - x'^r_{i_s} &= x^r_{j_t} - \pi'_{i_s} = x^r_{j_t} - \pi_{i_s} - \varepsilon \geq x^r_{j_t} - \pi_{i_s} - (\pi_{j_t} - \pi_{i_s} - L_{\underline{a}}) \\ &= x^r_{j_t} - \pi_{j_t} + L_{\underline{a}} \geq L_{\underline{a}}, \end{aligned}$$

which shows (5.10). Recall that by assumption there is no source delay on (i_s, j_t). The proof for (5.11) is analogous.

Second case: $z = 1$. In this case the assignment in the rolled out subgraph has shifted: for $k \in \{1, \dots, B\}$ we have $u'_{(i_s^k, i_t)} := u_{(i_s^k, i_{t-1})}$ for $t \geq 2$ and $u'_{(i_s^k, i_1)} := 0$. Furthermore, $u'_{(i_s, j_t)} := u_{(i_{s-1}, j_t)}$ for $s \geq 2$ and $u'_{(i_1, j_t)} := 0$, see Figure 6.5 for an example. For $r \in \mathcal{U}$ we set $x'^r_{i_1} := \pi'_{i_1}$ and $x'^r_{i_s} := \max\{x^r_{i_{s-1}}, \pi'_{i_s}\}$ for $s > 1$. Note that due to the assignment shift the node i_1 does not have any ingoing activities which could propagate delays, so $x'^r_{i_1} = \pi'_{i_1}$ is feasible. For $k \in \{1, \dots, B\}$ let $1 \leq s \leq K, s \leq t \leq K + b$ such that $u'_{(i_s^k, i_t)} = 1$. Then $x'^r_{i_t} - x'^r_{i_s^k} \geq x^r_{i_{t-1}} - x^r_{i_s^k} \geq L_{\underline{a}_k} + d^r_{(i_s^k, i_{t-1})}$. For s, t such that $u'_{(i_s, j_t)} = 1$ we either have

$$x'^r_{j_t} - x'^r_{i_s} = x^r_{j_t} - x^r_{i_{s-1}} \geq L_{\underline{a}}$$

or

$$\begin{aligned} x'^r_{j_t} - x'^r_{i_s} &= x^r_{j_t} - \pi'_{i_s} = x^r_{j_t} - (\pi_{i_s} + \varepsilon - T) = x^r_{j_t} - \pi_{i_{s-1}} - \varepsilon \\ &\geq x^r_{j_t} - \pi_{i_{s-1}} - (\pi_{j_t} - \pi_{i_{s-1}} - L_{\underline{a}}) = x^r_{j_t} - \pi_{j_t} + L_a \geq L_{\underline{a}}. \end{aligned}$$

Hence, (x', y') yields a feasible disposition timetable. Since $y' = y$, the bound β is respected. Furthermore, note that for every $r \in \mathcal{U}$ in the first case $(z = 0)$ we have

$$x'^r_{i_s} - \pi'_{i_s} = \begin{cases} x^r_{i_s} - \pi_{i_s} - \varepsilon < x^r_{i_s} - \pi_{i_s} & \text{if } x'^r_{i_s} = x^r_{i_s}, \\ 0 \leq x^r_{i_s} - \pi_{i_s} & \text{if } x'^r_{i_s} = \pi'_{i_s}. \end{cases}$$

Analogous for the second case. In particular, we obtain

$$\begin{aligned}
&\sum_{l_s \in \mathcal{E}: s \leq K} w_{l_s}(x'^r_{l_s} - \pi'_{l_s}) + \sum_{\substack{a'=(l',l)\in \mathcal{A}_{\text{out}}:\\ u'_{a'}=1}} w_{a'}(x'^r_l - \pi'_l) \\
\leq &\sum_{l_s \in \mathcal{E}: s \leq K} w_{l_s}(x^r_{l_s} - \pi_{l_s}) + \sum_{\substack{a'=(l',l)\in \mathcal{A}_{\text{out}}:\\ u'_{a'}=1}} w_{a'}(x^r_l - \pi_l) \\
\overset{(*)}{\leq} &\sum_{l_s \in \mathcal{E}: s \leq K} w_{l_s}(x^r_{l_s} - \pi_{l_s}) + \sum_{\substack{a'=(l',l)\in \mathcal{A}_{\text{out}}:\\ u_{a'}=1}} w_{a'}(x^r_l - \pi_l) \leq \alpha.
\end{aligned}$$

To see $(*)$, we have to give special attention to the arcs in $\mathcal{A}_{\text{out}}$ in the case that we have a shift in the assignment variables (i.e. $z = 1$). For all arcs a' which are not in the considered subgraph, the value of $u_{a'}$ does not change. Since i_{K+1}, and hence also i_{K+c} for $c \geq 1$, is not reachable from any source delay, we have $x^r_{i_{K+c}} - \pi_{i_{K+c}} = 0$, so we only have to consider those arcs corresponding to (i, j). If $u'_{(i_s,j_t)} = 1$, also $u_{(i_{s-1},j_t)} = 1$, which implies

$$\{t : \exists a = (i_s, j_t) \in \mathcal{A}_{\text{out}} \text{ with } u'_a = 1\} \subseteq \{t : \exists a = (i_s, j_t) \in \mathcal{A}_{\text{out}} \text{ with } u_a = 1\}.$$

Since the passenger weights w are periodic, $(*)$ follows.

Overall, we have found a better solution, which is a contradiction to π being an optimal timetable. □

Example 6.10. Theorem 6.9 does not hold if the EAN contains headway activities. To see this we consider the EAN given in Figure 6.6 with the induced subgraph given by the nodes $\{i^2, i^3, i^4\}$. The assumptions of the theorem are fulfilled. However, putting one minute of slack on the activity (i^3, i^4) and no slack on (i^2, i^3) yields an optimal solution. We cannot simply shift the time of event i^3 as done in the proof, because due to the headway activities this also influences the time of event i^7.

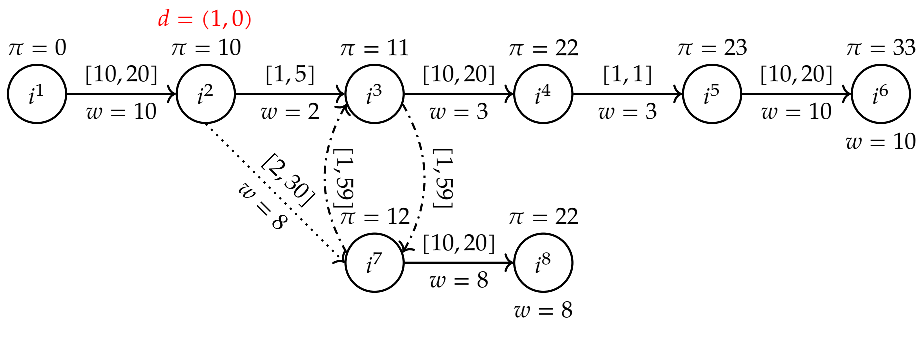


Figure 6.6: Theorem 6.9 does not hold if the EAN has headways ($T = 60$, $K = 2$).

7

HEURISTICS

Due to the high complexity of RRPT, it is unlikely to be solved to optimality in reasonable time – even for medium-sized instances. Hence, this chapter is dedicated to developing heuristic methods. We pursue iterative approaches, alternatingly solving the timetabling subproblem and the delay management subproblem.

OUTLINE In Section 7.1 we describe three algorithms, which all follow the same idea, but differ in the choice of activities on which to put buffer times. They all can be modified by choosing some parameters. In Section 7.2 we compare them in a computational study to assess their solution quality.

7.1 DESCRIPTION OF THE ALGORITHMS

The idea of the first algorithm is simple. We first solve PESP and for the obtained solution π we solve DM$(\mathcal{U})$, i.e. we solve DM for every scenario $r \in \mathcal{U}$. Then we choose some transfer activity $a' \in \texttt{candidates} = \mathcal{A}_{\text{transfer}}(\pi)$ and increase the lower bound for the corresponding periodic activity $\underline{a}'$, i.e. we enforce some buffer time on that transfer. If the increased lower bound causes PESP to become infeasible, we decrease it again until it becomes feasible. In every iteration we check if the worst-case delay has decreased compared to the incumbent solution. If so, we continue. Otherwise, the algorithm terminates with the incumbent solution. If all transfers are maintained, the algorithm also terminates. Note that since the worst-case delay is a natural number which decreases in every iteration, the algorithm terminates. The basic scheme of the heuristic is given in Algorithm 1.

It remains to specify how we choose the transfer a' and how we set the new lower bound. For the choice of the transfer we give three options:

- `simple`: $a' \in \operatorname{argmax}_{a=(i,j)\in\texttt{candidates}} |\{r \in \mathcal{U} : x_i^r - x_j^r + L_a > 0\}|$
- `weighted`: $a' \in \operatorname{argmax}_{a=(i,j)\in\texttt{candidates}} w_a|\{r \in \mathcal{U} : x_i^r - x_j^r + L_a > 0\}|$
- `tight`: $a' \in \operatorname{argmin}_{a\in\texttt{candidates}:C_a>0} C_a$, where $C_a := \frac{\sum_{r\in\mathcal{U}} \max(0, x_i^r - x_j^r + L_a)}{\max(1, |\{r\in\mathcal{U} : x_i^r - x_j^r + L_a > 0\}|)}$ for $a = (i,j) \in \texttt{candidates}$.

The option `simple` counts for every transfer activity in the set `candidates` in how many scenarios $r \in \mathcal{U}$ it is cancelled and chooses that activity for which this number is maximal. The option `weighted` basically does the same thing, but takes the passenger weights of the transfers into account. For the option `tight` the idea is the following: if a transfer $a = (i,j)$ is cancelled in scenario $r \in \mathcal{U}$, we look at the time which would have been necessary to still catch it, i.e. $x_i^r - x_j^r + L_a$. If this number is small, this means only a small additional buffer would have been necessary to maintain that transfer, i.e. the increase in nominal travel time to decrease the number of missed transfers is small. Hence, we choose that transfer for which the average time missing in order to secure the transfer is minimal.

For the choice of the new lower bound for the chosen transfer $a' = (i, j)$ we also use three different options. Let $\underline{a}' = (\underline{i}, \underline{j})$ be the periodic activity corresponding to the chosen transfer $a' = (i, j)$. Recall that $L_{\underline{a}'} = L_{a'}$. We consider the following options:

- `0`: $\tilde{L}_{\underline{a}'} = \pi_{\underline{j}} - \pi_{\underline{i}} + z_{\underline{a}'}T + \max_{r \in \mathcal{U}} \max(0, x_i^r - x_j^r + L_{a'})$
- `1`: $\tilde{L}_{\underline{a}'} = \max_{r \in \mathcal{U}} \max(0, x_i^r - x_j^r + L_{a'})$
- `2`: $\tilde{L}_{\underline{a}'} = \pi_{\underline{j}} - \pi_{\underline{i}} + z_{\underline{a}'}T + \frac{\max_{r \in \mathcal{U}} \max(0, x_i^r - x_j^r + L_{a'})}{2}$.

We want to increase the lower bound such that a' is cancelled less often. The term

$$\max_{r \in \mathcal{U}} \max(0, x_i^r - x_j^r + L_{a'})$$

gives the maximal time missing to maintain the transfer. In option `0` we add this to the current duration of activity a' (respectively $\underline{a}'$), with the idea to ensure the transfer a' in all scenarios. Since this new lower bound can be rather large, and hence might lead to overly large nominal travel times, option `2` only adds half of the maximal time missing to the current duration. Option `1` might choose bounds even lower than that. In fact, for this option it is not guaranteed that $\tilde{L}_{\underline{a}'}$ is larger than $L_{\underline{a}'}$, in which case the lower bound is not changed and the algorithm terminates.

The basic idea of the next heuristic is similar to the previous one. However, instead of only adding buffer to a transfer activity in every iteration, we also choose a set `critical-wait` consisting of $\lceil n \cdot |\underline{\mathcal{A}}_{\text{wait}}| \rceil$ waiting activities, $0 < n < 1$, for which we increase the lower bound by a given percentage p. The heuristic is given in Algorithm 2.

For the choice of the set `critical-wait` we have two options:

- `smallest`: Choose the activities $\underline{a} = (\underline{i}, \underline{j}) \in \texttt{candidates}^{wait}$ with the smallest buffer $(\pi_{\underline{j}} - \pi_{\underline{i}} - L_{\underline{a}}) \mod T$
- `quotient`: Choose the activities $\underline{a} = (\underline{i}, \underline{j}) \in \texttt{candidates}^{wait}$ for which the ratio $\frac{w_{\underline{a}}}{w_{\underline{e}}}$ of the passenger weights of $\underline{a}$ and the succeeding driving activity $\underline{e}$ is smallest.

The motivation for `smallest` is that buffer should be put on activities which currently only have a small buffer. For the idea for the option `quotient` consider the following situation: a small number of passengers sit in a train currently waiting at a station where a lot of passengers board the train. Let $\underline{a}$ be the waiting activity and $\underline{e}$ the succeeding driving activity, i.e. $w_{\underline{a}}$ is small and $w_{\underline{e}}$ is large. It might be a good idea to put buffer on $\underline{a}$, because the increase of the nominal travel time only affects $w_{\underline{a}}$ passengers, but $w_{\underline{e}}$ passengers might profit from it when the buffer can be used to decrease the delay.

A straightforward extension of Algorithm 2 is to also include driving activities. They are then treated in the same manner as the waiting activities, i.e. we also choose a subset `critical-drive` of the driving activities in every iteration. However, note that the reasoning for the option `quotient` does not make sense for driving activities, so we only use the option `smallest`. The procedure is given in Algorithm 3.

Due to the modular structure of the algorithms, we can also replace solving PESP and DM exactly by using a heuristic of our choice. This will be necessary for large instances due to the high complexity of PESP and DM.

Algorithm 1: BUFFER-TRANSFER

Input : Periodic EAN $\mathcal{N} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$.
Output : Periodic timetable π and disposition timetables (x^r, y^r) for all $r \in \mathcal{U}$.

`bound` $= \infty$
Solve PESP $\rightsquigarrow \tilde{\pi}$
Roll out w.r.t. $\tilde{\pi}$
`candidates` $= \mathcal{A}_{\text{transfer}}(\tilde{\pi})$
while *true* **do**
    `infeasible` = true
    for $r \in \mathcal{U}$ **do**
        Solve DM$(r) \rightsquigarrow (\tilde{x}^r, \tilde{y}^r)$
    end
    if $f^{\text{del}}(\tilde{\pi}) \geq$ *bound* **then**
        return $\pi, (x^r, y^r)_{r \in \mathcal{U}}$
    end
    $\pi = \tilde{\pi}, x^r = \tilde{x}^r, y^r = \tilde{y}^r$, `bound` $= f^{\text{del}}(\tilde{\pi})$
    if *all transfers are maintained* **then**
        return $\pi, (x^r, y^r)_{r \in \mathcal{U}}$
    end
    Choose critical transfer $a' \in$ `candidates` and determine $\underline{a}'$ such that $a' \in \mathcal{A}(\underline{a}')$
    Choose new lower bound $\tilde{L}_{\underline{a}'}$
    while *`infeasible`* **do**
        Solve PESP with new lower bound $\tilde{L}_{\underline{a}'} \rightsquigarrow \tilde{\pi}$
        if *PESP is infeasible* **then**
            Reduce lower bound: $\tilde{L}_{\underline{a}'} = \left\lfloor \frac{\tilde{L}_{\underline{a}'} + L_{\underline{a}'}}{2} \right\rfloor$
        end
        else
            `infeasible` = false
            `candidates` = `candidates` $\setminus \{a'\}$
            Roll out w.r.t. $\tilde{\pi}$.
        end
    end
end

Algorithm 2: BUFFER-TRANSFER-WAIT

Input : Periodic EAN $\mathcal{N} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$, $0 < n < 1$, $0 < p < 1$.

Output : Periodic timetable π and disposition timetables (x^r, y^r) for all $r \in \mathcal{U}$.

```
bound = \infty
N = \lceil n \cdot |\mathcal{A}_{wait}| \rceil
Solve PESP \rightsquigarrow \tilde{\pi}
Roll out w.r.t. \tilde{\pi}
candidates^{trans} = \mathcal{A}_{transfer}(\tilde{\pi})
candidates^{wait} = \underline{\mathcal{A}}_{wait}
while true do
    infeasible = true
    for r \in \mathcal{U} do
        Solve DM(r) \rightsquigarrow (\tilde{x}^r, \tilde{y}^r)
    end
    if f^{del}(\tilde{\pi}) \geq bound then
        return \pi, (x^r, y^r)_{r \in \mathcal{U}}
    end
    \pi = \tilde{\pi}, x^r = \tilde{x}^r, y^r = \tilde{y}^r, bound = f^{del}(\tilde{\pi})
    if all transfers are maintained then
        return \pi, (x^r, y^r)_{r \in \mathcal{U}}
    end
    Choose critical transfer a' \in candidates^{trans} and determine \underline{a}' such that
      a' \in \mathcal{A}(\underline{a}')
    Choose new lower bound \tilde{L}_{\underline{a}'}
    Choose up to N waiting activities critical-wait \subseteq candidates^{wait}
    for \underline{a} \in critical-wait do
        \tilde{L}_{\underline{a}} = \lceil L_{\underline{a}} \cdot (1 + p) \rceil
    end
    while infeasible do
        Solve PESP with new lower bounds \tilde{L} \rightsquigarrow \tilde{\pi}
        if PESP is infeasible then
            Reduce lower bounds: \tilde{L}_{\underline{a}'} = \lfloor (\tilde{L}_{\underline{a}'} + L_{\underline{a}'}) / 2 \rfloor, \tilde{L}_{\underline{a}} = \lfloor (\tilde{L}_{\underline{a}} + L_{\underline{a}}) / 2 \rfloor
        end
        else
            infeasible = false
            candidates^{trans} = candidates^{trans} \ {a'}
            candidates^{wait} = candidates^{wait} \ critical-wait
            Roll out w.r.t. \tilde{\pi}
        end
    end
end
```

Algorithm 3: BUFFER-TRANSFER-WAIT-DRIVE

Input : Periodic EAN $\mathcal{N} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$, $0 < n < 1$, $0 < p < 1$.

Output: Periodic timetable π and disposition timetables (x^r, y^r) for all $r \in \mathcal{U}$.

```
bound = ∞
N_1 = ⌈n · |A_wait|⌉
N_2 = ⌈n · |A_drive|⌉
Solve PESP ⇝ π̃
Roll out w.r.t. π̃
candidates^trans = A_transfer(π̃)
candidates^wait = A̲_wait
candidates^drive = A̲_drive
while true do
    infeasible = true
    for r ∈ U do
        Solve DM(r) ⇝ (x̃^r, ỹ^r)
    end
    if f^del(π̃) ≥ bound then
        return π, (x^r, y^r)_{r∈U}
    end
    π = π̃, x^r = x̃^r, y^r = ỹ^r, bound = f^del(π̃)
    if all transfers are maintained then
        return π, (x^r, y^r)_{r∈U}
    end
    Choose critical transfer a' ∈ candidates^trans and determine a̲' such that
      a' ∈ A(a̲')
    Choose new lower bound L̃_a̲'
    Choose up to N_1 waiting activities critical-wait ⊆ candidates^wait
    Choose up to N_2 driving activities critical-drive ⊆ candidates^drive
    for a̲ ∈ critical-wait ∪ critical-drive do
        L̃_a̲ = ⌈L_a̲ · (1 + p)⌉
    end
    while infeasible do
        Solve PESP with new lower bounds L̃ ⇝ π̃
        if PESP is infeasible then
            Reduce lower bounds: L̃_a̲' = ⌊(L̃_a̲' + L_a̲')/2⌋, L̃_a̲ = ⌊(L̃_a̲ + L_a̲)/2⌋
        end
        else
            infeasible = false
            candidates^trans = candidates^trans \ {a'}
            candidates^wait = candidates^wait \ critical-wait
            Roll out w.r.t. π̃
        end
    end
end
```

7.2 COMPUTATIONAL EXPERIMENTS

To test the heuristics, we used the dataset `lowersaxony` from LinTim [Sch+23; Sch+], which we already used in Chapter 4 for different line concepts. An overview of the EAN used for the following experiments is given in Table 7.1. This dataset is small enough to still be able to solve PESP and DM exactly, but already too large for RRPT. We rolled out the network for $K \in \{4, 12\}$ periods and generated random source delays between 1 and 15 minutes for 5% of the *aperiodic* events and activities.

We implemented the algorithms in Python and ran them on a compute server with 48 cores @2.9 GHz and 196 GB RAM. For solving the MIP formulations we used Gurobi 8.1.1 [Gur23].

Name	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{A}_{\text{drive}}\rvert$	$\lvert\mathcal{A}_{\text{wait}}\rvert$	$\lvert\mathcal{A}_{\text{transfer}}\rvert$
`lowersaxony`	180	187	90	80	17

Table 7.1: Size of the EAN.

ALGORITHM 1 We have nine different variants of Algorithm 1, namely for each parameter combination from $\{\texttt{simple}, \texttt{weighted}, \texttt{tight}\} \times \{0, 1, 2\}$. In Figure 7.1 we plotted their solutions evaluated with respect to the worst-case delay, which is the objective of DM, on the x-axis and the nominal travel time, i.e. the objective of PESP, on the y-axis for $|\mathcal{U}| \in \{50, 100, 150, 200\}$. For comparison, we also added the PESP solution, which is always in the bottom right corner, since it has the smallest possible nominal travel time and does not take delays into account.

As we can see, with only a few exceptions the solutions for `tight` tend to be in the bottom right corner, close to the PESP solution. For `simple` and `weighted` the algorithm finds solutions with significantly less delay – of course this comes at the cost of an increase in the nominal travel time.

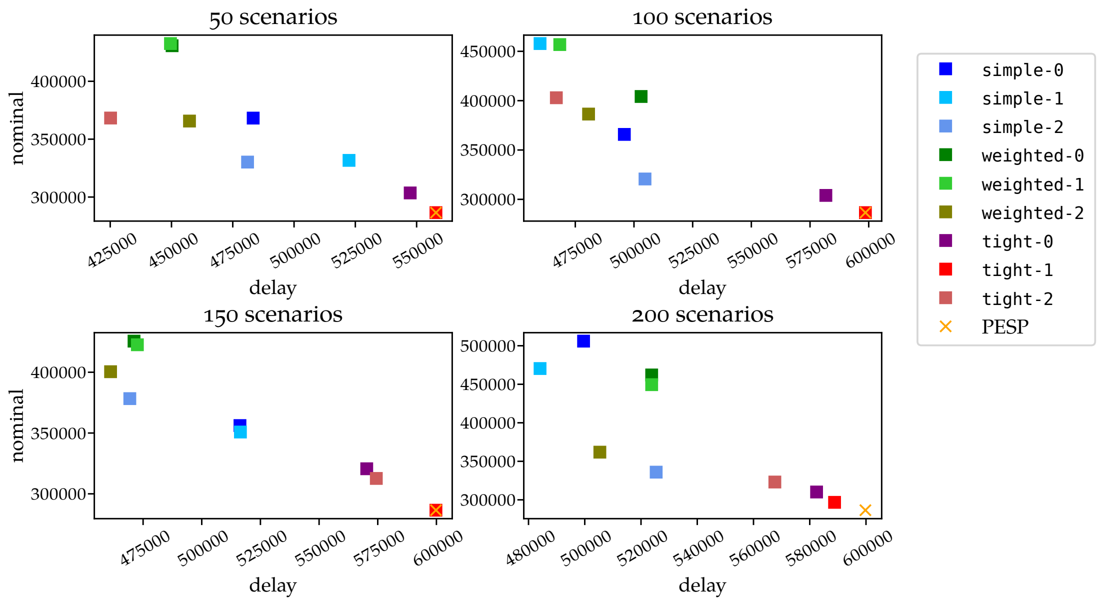
Since we are interested in the real travel time, we also evaluated the solutions with respect to the real travel time and computed how much it differs from the real travel time of the PESP solution. In Tables 7.2 and 7.3 we see the results for those parameter combinations which yield an improvement compared to the PESP solution. As we can see, the combinations `simple-2` and `weighted-2` are the only ones yielding a significant improvement. For $K = 4$ the algorithm was able to improve the PESP solution by 3.6%, for $K = 12$ we have an improvement of 2.7%. For the option `tight` we were not able to achieve any improvement.

Parameters	50	100	150	200	Average
`simple-2`	-2.85	-7.01	-1.71	-2.82	**-3.6**
`weighted-2`	-2.54	-1.51	-1.59	-2.15	-1.95

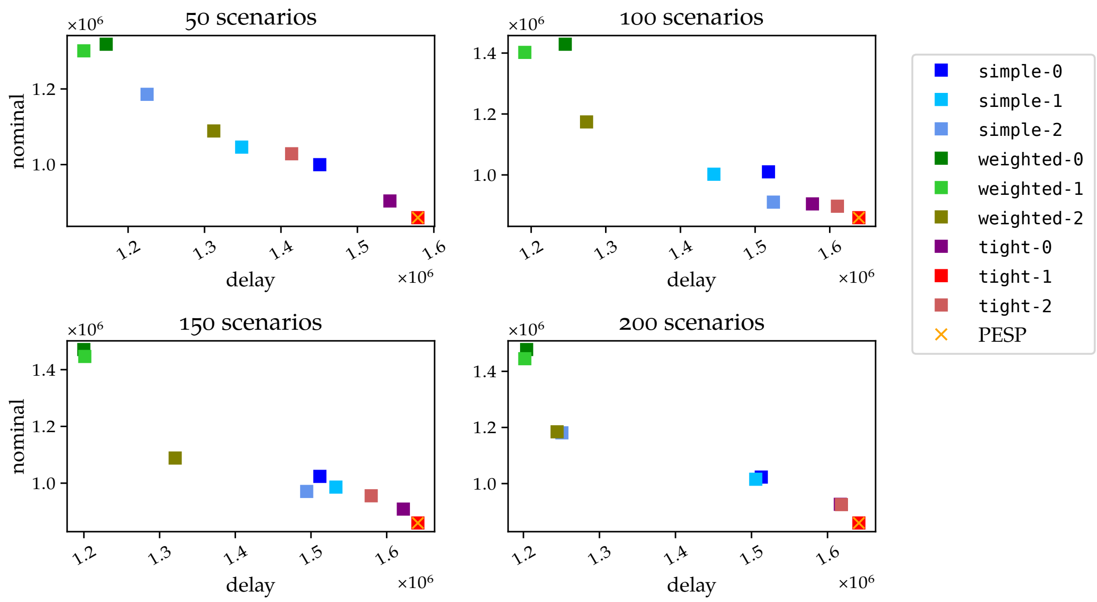
Table 7.2: Change (%) in real travel time for Algorithm 1 for $K = 4$ compared to the PESP solution.

Parameters	50	100	150	200	Average
`simple-2`	-1.74	-2.54	-1.43	-1.97	-1.92
`weighted-1`	-1.72	-2.06	0.74	0.81	-0.56
`weighted-2`	-2.5	-3.21	-2.84	-2.26	**-2.7**

Table 7.3: Change (%) in real travel time for Algorithm 1 for $K = 12$ compared to the PESP solution.



(a) For $K = 4$.



(b) For $K = 12$.

Figure 7.1: Objective values of solutions of Algorithm 1.

ALGORITHM 2 For Algorithm 2 we chose $p = 0.1$, i.e. the lower bound of the chosen waiting activities `critical-wait` is increased by 10%. The percentage n of the number of chosen waiting activities was taken from the set $\{0.02, 0.05, 0.1\}$. In total, we tried 54 parameter combinations from

$$\{\texttt{simple}, \texttt{weighted}, \texttt{tight}\} \times \{\texttt{0}, \texttt{1}, \texttt{2}\} \times \{\texttt{smallest}, \texttt{quotient}\} \times \{0.02, 0.05, 0.1\}.$$

The results for $K = 4$ are depicted in Figure 7.3. Note that the legend can be found separately in Figure 7.2. One can roughly say that the values for `weighted` are located in the top left corner, those for `tight` in the bottom right corner, and the values for `simple` in-between. Thus we can say that `weighted` puts more emphasis on reducing the delay, `tight` yields solutions with small nominal travel time, and `simple` finds trade-offs between these extremes. Due to the large number of different parameter combinations these graphics are quite crowded. Hence, in Figure 7.3b we show a reduced version, where all points which are dominated by others, and those which yield the same value as PESP, are removed. As we can see, the remaining solutions mainly belong to `simple-2`, `weighted-1` and `weighted-2`. The value for n and the choice between `smallest` and `quotient` seems to have less influence on the quality of the solution. If we increase the number of considered periods from $K = 4$ to $K = 12$, the different emphasis of `simple`, `weighted` and `tight` on the conflicting objectives becomes even more prominent, as can be seen in Figure 7.4. Also in this setting we removed the dominated values, leaving those in Figure 7.4b. Here, additionally to `simple-2`, `weighted-1` and `weighted-2` also `weighted-0` and `simple-1` yield values not dominated by the others.

Also for Algorithm 2 we investigated the improvement of the real travel time compared to the PESP solution. The results are shown in Table 7.4 for $K = 4$ and in Table 7.5 for $K = 12$. Note that due to the large number of variants we only show those results with an average improvement of at least 2% compared to the PESP solution. We can see that `simple-2` and `weighted-2` always yield good results – regardless of the choice of n and whether we use `smallest` or `quotient`. While for $K = 4$ `simple-2` and `weighted-2` seem to be similarly good – both are able to improve the PESP solution by more than 5% – for $K = 12$ `weighted-2` yields the best solutions with an improvement of roughly 4.5% for several parameter combinations. As already seen in the results for Algorithm 1, also for Algorithm 2 the option `tight` does not prove useful.

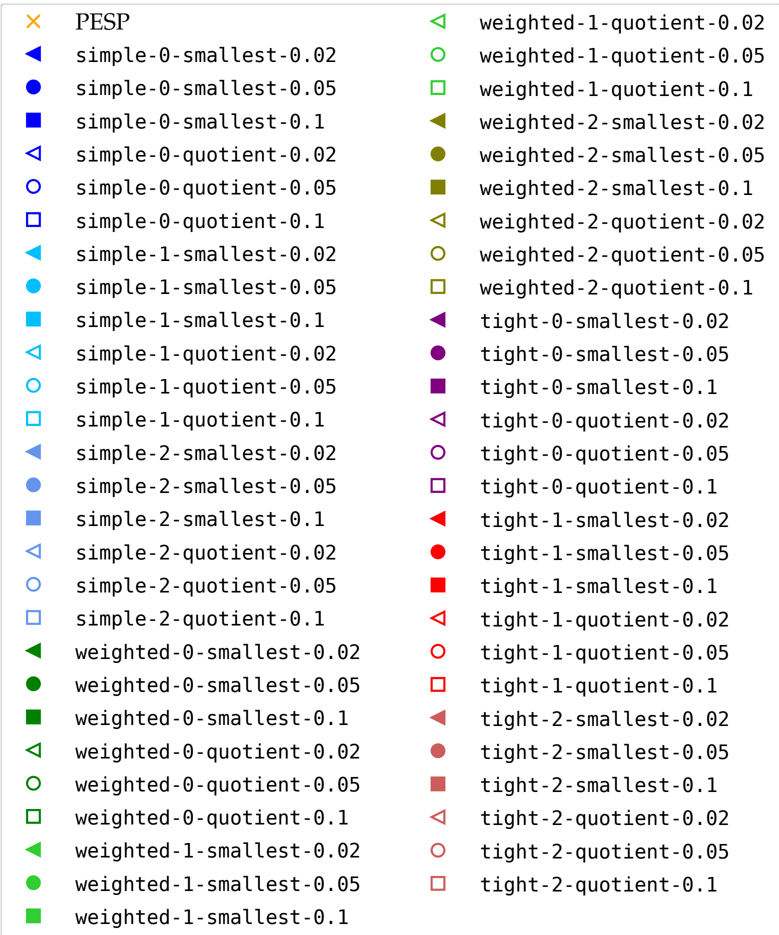
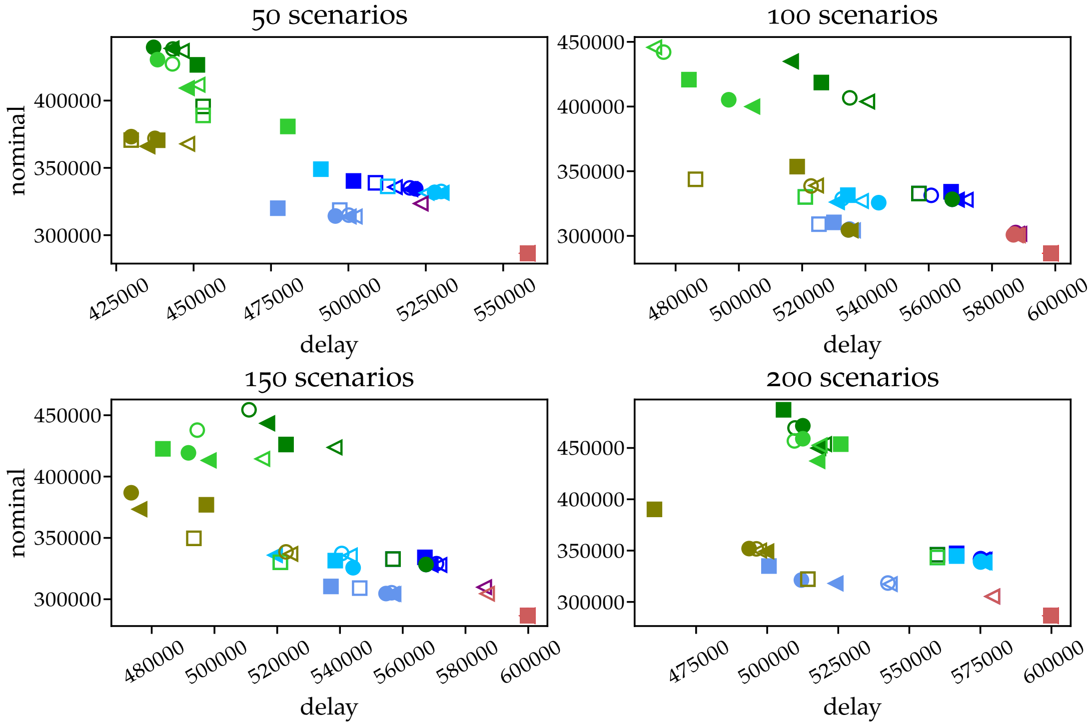
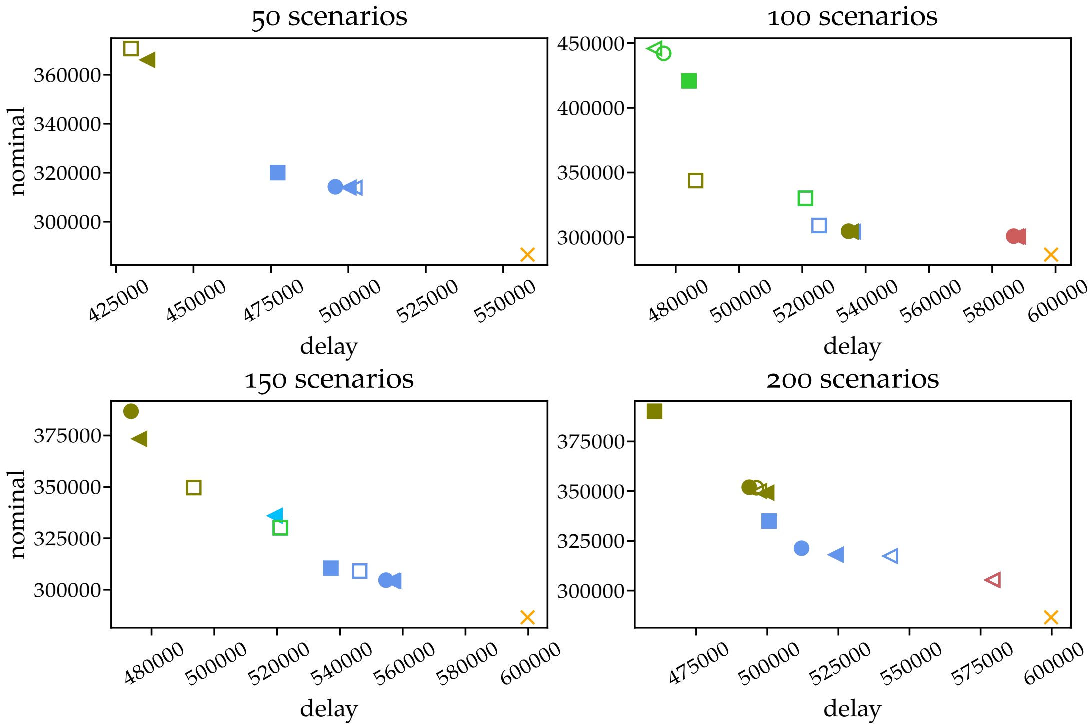


Figure 7.2: Legend for Figures 7.3 to 7.6.

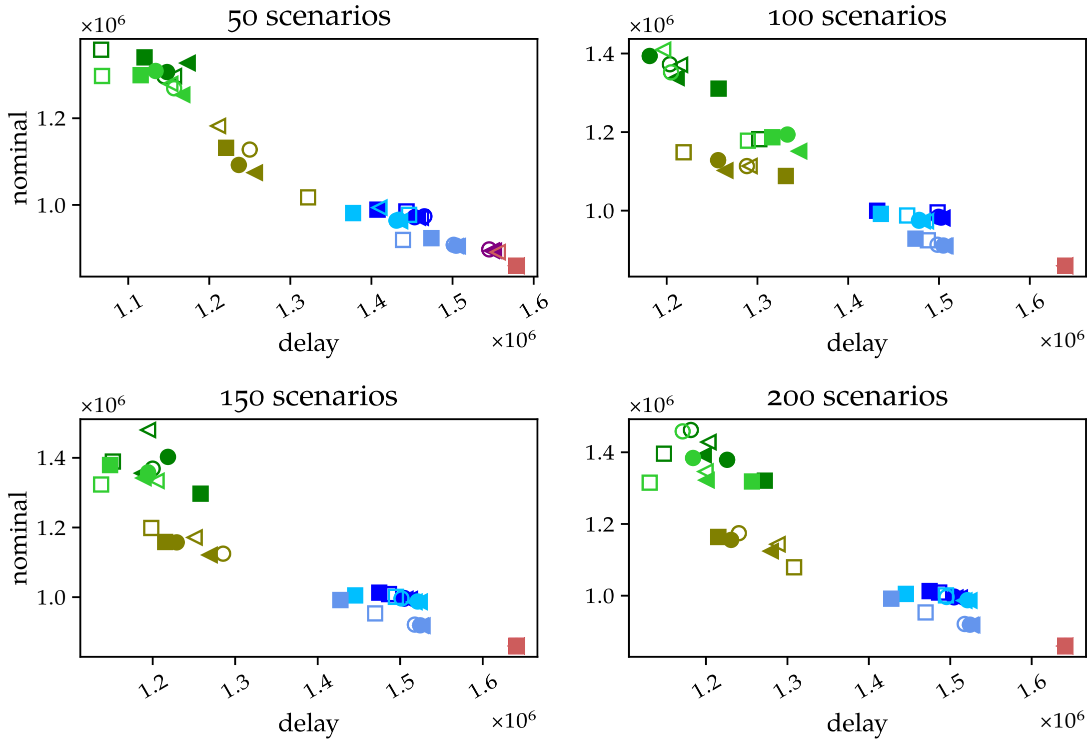


(a) Objective values of all solutions.

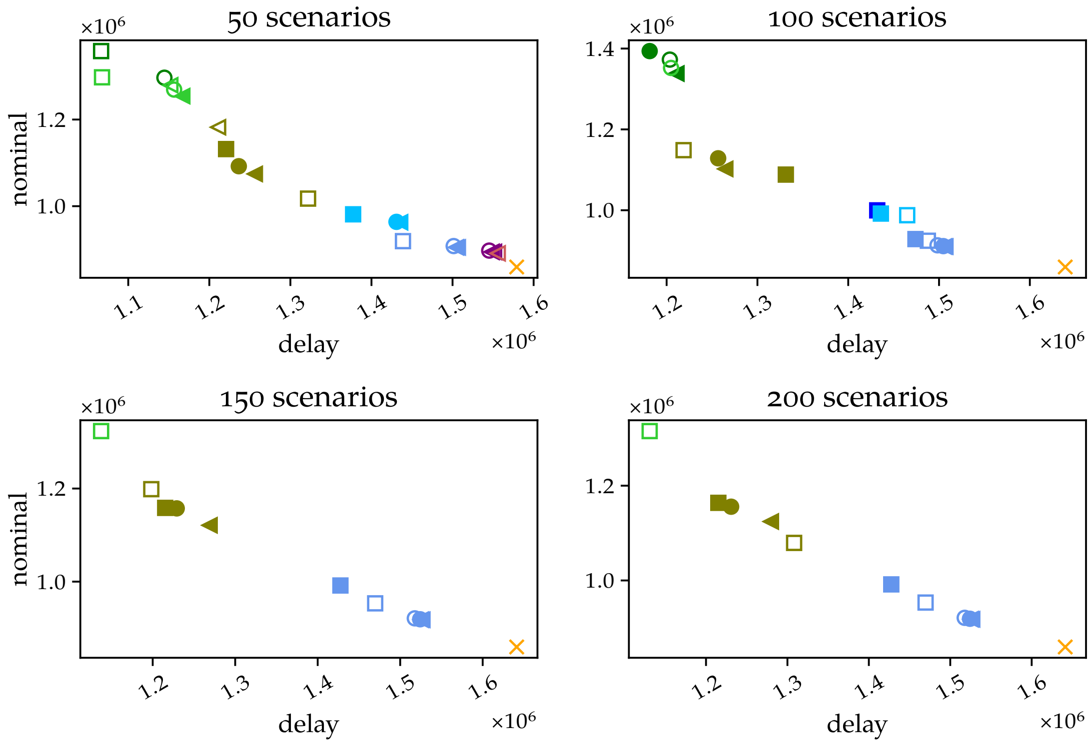


(b) Objective values which are not dominated by others.

Figure 7.3: Objective values of solutions of Algorithm 2 for $K = 4$.



(a) Objective values of all solutions.



(b) Objective values which are not dominated by others.

Figure 7.4: Objective values of solutions of Algorithm 2 for $K = 12$.

Parameters	50	100	150	200	Average
simple-2-smallest-0.02	-3.59	-5.13	-2.87	-4.98	-4.15
simple-2-smallest-0.05	-4.07	-5.18	-3.05	-5.98	-4.57
simple-2-smallest-0.1	-5.59	-5.05	-4.37	-5.73	**-5.18**
simple-2-quotient-0.02	-3.37	-5.06	-2.82	-2.91	-3.54
simple-2-quotient-0.05	-3.48	-5.07	-2.78	-2.87	-3.55
simple-2-quotient-0.1	-3.38	-5.73	-3.49	-5.62	-4.56
weighted-2-smallest-0.02	-5.12	-5.13	-4.15	-4.21	-4.65
weighted-2-smallest-0.05	-4.9	-5.18	-2.94	-4.59	-4.4
weighted-2-smallest-0.1	-4.2	-1.49	-1.35	-4.04	-2.77
weighted-2-quotient-0.02	-3.38	-2.46	-2.83	-4.4	-3.27
weighted-2-quotient-0.05	-4.14	-2.68	-2.82	-4.34	-3.49
weighted-2-quotient-0.1	-5.2	-6.22	-4.88	-5.62	**-5.48**

Table 7.4: Change (%) in real travel time for Algorithm 2 for $K = 4$ for those parameter combinations with at least 2% average improvement compared to the PESP solution.

Parameters	50	100	150	200	Average
simple-1-smallest-0.1	-3.29	-2.84	-2.01	-2.01	-2.54
simple-2-smallest-0.02	-1.2	-3.31	-2.3	-2.3	-2.28
simple-2-smallest-0.05	-1.16	-3.31	-2.3	-2.3	-2.27
simple-2-smallest-0.1	-1.7	-3.84	-3.26	-3.26	-3.02
simple-2-quotient-0.02	-1.12	-3.26	-2.23	-2.23	-2.21
simple-2-quotient-0.05	-1.21	-3.48	-2.48	-2.48	-2.41
simple-2-quotient-0.1	-3.31	-3.46	-3.11	-3.11	-3.25
weighted-1-quotient-0.1	-3.02	-1.24	-1.6	-2.19	-2.01
weighted-2-smallest-0.02	-4.43	-5.27	-4.43	-3.87	**-4.5**
weighted-2-smallest-0.05	-4.51	-4.54	-4.57	-4.57	**-4.55**
weighted-2-smallest-0.1	-3.54	-3.17	-5.09	-4.87	-4.17
weighted-2-quotient-0.02	-1.87	-3.74	-3.14	-2.77	-2.88
weighted-2-quotient-0.05	-2.51	-3.87	-3.62	-3.45	-3.36
weighted-2-quotient-0.1	-4.08	-5.25	-4.16	-4.53	**-4.51**

Table 7.5: Change (%) in real travel time for Algorithm 2 for $K = 12$ for those parameter combinations with at least 2% average improvement compared to the PESP solution.

Parameters	50	100	150	200	Average
simple-2-0.02	-3.48	-7.02	-2.88	-4.31	-4.42
simple-2-0.05	-3.56	-5.45	-3.24	-4.67	-4.23
simple-2-0.1	-4.12	-5.8	-4.52	-5.96	**-5.1**
weighted-2-0.02	-3.54	-4.52	-3.93	-3.76	-3.94
weighted-2-0.05	-3.56	-5.07	-4.49	-4.67	-4.45
weighted-2-0.1	-4.12	-5.8	-4.52	-5.96	**-5.1**
tight-0-0.1	-5.66	-6.69	-6.82	-6.82	**-6.5**

Table 7.6: Change (%) in real travel time for Algorithm 3 for $K = 4$ for those parameter combinations with at least 2% average improvement compared to the PESP solution.

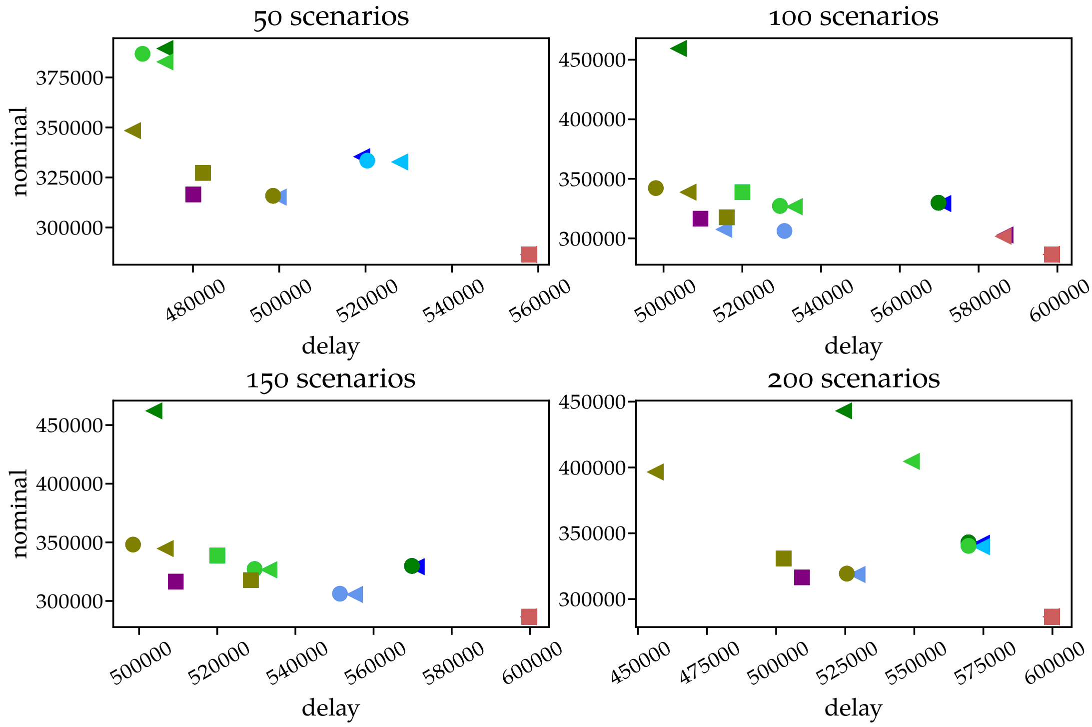
Parameters	50	100	150	200	Average
simple-2-0.02	-1.28	-3.27	-3.05	-3.05	-2.66
simple-2-0.05	-1.28	-3.46	-3.19	-3.19	-2.78
simple-2-0.1	0.0	0.0	-4.71	-4.71	-2.36
weighted-2-0.02	-5.63	-5.97	-4.26	-4.6	**-5.12**
weighted-2-0.05	-4.4	-4.63	-5.08	-2.98	-4.27
weighted-2-0.1	0.0	0.0	-4.71	-4.71	-2.36

Table 7.7: Change (%) in real travel time for Algorithm 3 for $K = 12$ for those parameter combinations with at least 2% average improvement compared to the PESP solution.

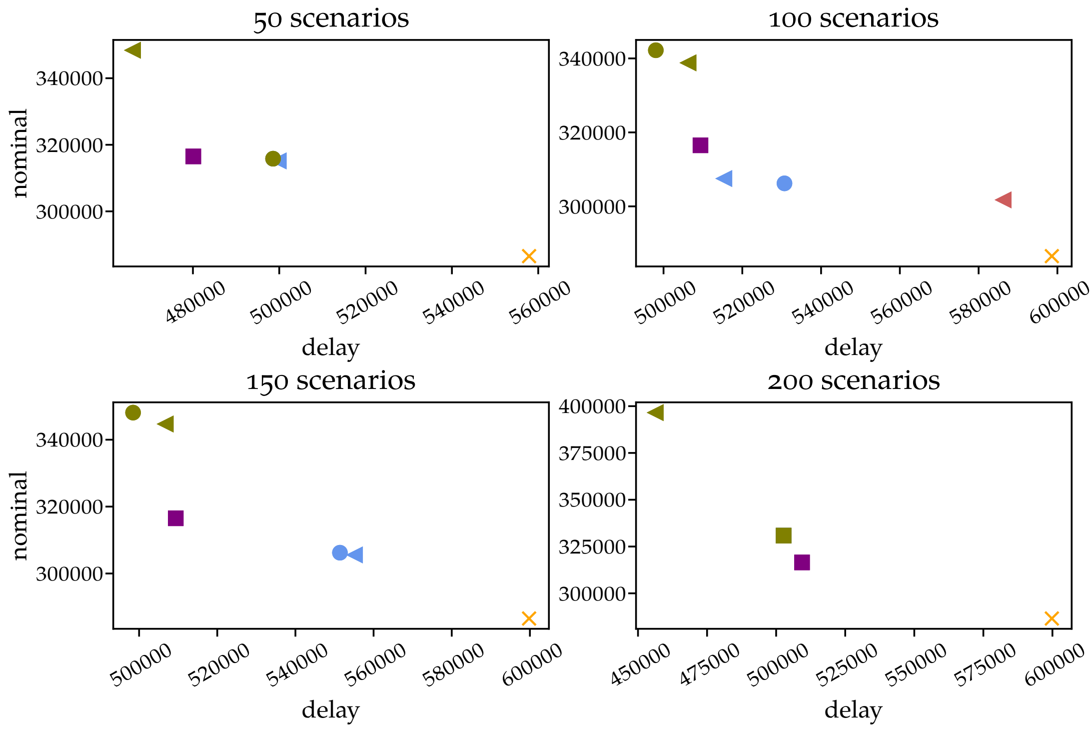
ALGORITHM 3 Finally, we look at the results of Algorithm 3. Since the choice between `smallest` and `quotient` does not have a significant influence for the waiting activities, as we have seen in the experiments for Algorithm 2, and the option `quotient` is not reasonable for driving activities, we omit this option for Algorithm 3. Hence, we test the 27 parameter combinations from

$$\{\texttt{simple}, \texttt{weighted}, \texttt{tight}\} \times \{0, 1, 2\} \times \{\texttt{smallest}\} \times \{0.02, 0.05, 0.1\}.$$

Note that we omit the term "`smallest`" when specifying the parameter combination. As before, we looked at the nominal travel time and the worst-case delay in Figures 7.5 and 7.6. The results look similar to those of Algorithm 2, with the exception of `tight-0-0.1` for $K = 4$. While in the previous experiments the option `tight` always yielded solutions close to that of PESP, in this instance we have solutions with much smaller delay for this specific parameter combination. In fact, when looking at the improvement of the real travel time in Table 7.6, we can see that this is actually the best variant with an improvement of 6.5%. Apart from that, `simple-2` and `weighted-2` are again a good choice with an improvement of up to 5.1%. For $K = 12$, see Table 7.7, these options again turn out to be the best ones, with `weighted-2` outperforming `simple-2`. Here, the option `tight` cannot achieve a significant improvement.

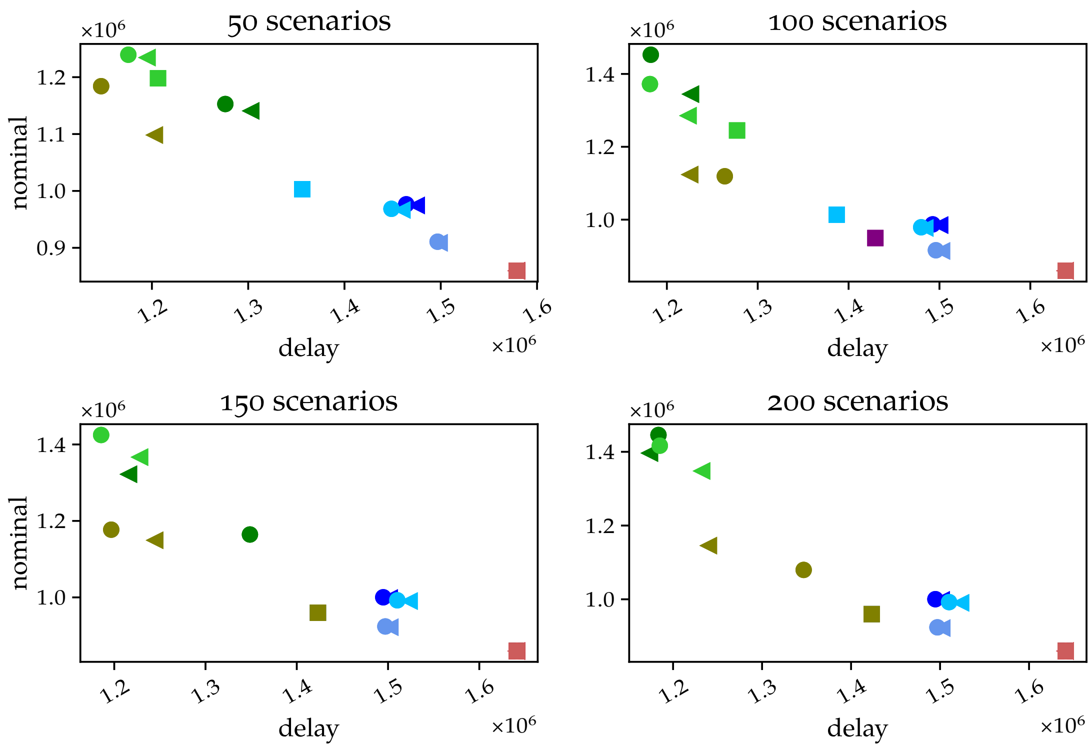


(a) Objective values of all solutions.

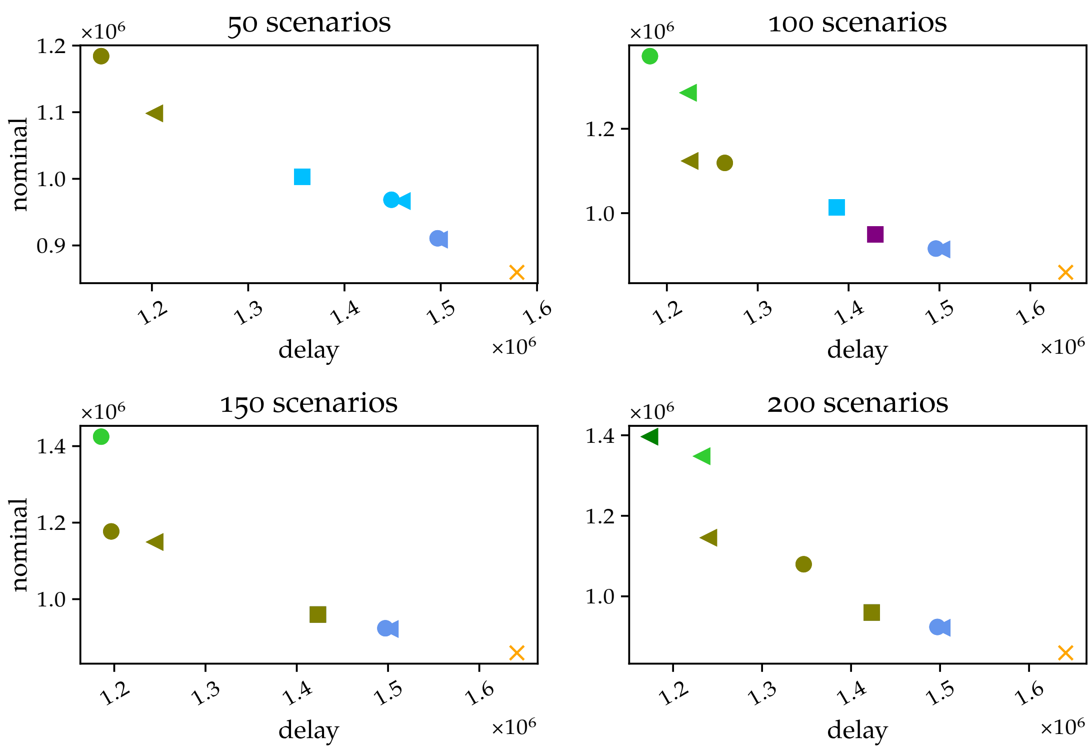


(b) Objective values which are not dominated by others.

Figure 7.5: Objective values of solutions of Algorithm 3 for $K = 4$.



(a) Objective values of all solutions.



(b) Objective values which are not dominated by others.

Figure 7.6: Objective values of solutions of Algorithm 3 for $K = 12$.

In total, we can say that `simple-2` and `weighted-2` are the only combinations which yield good results in all considered instances, and for a larger planning horizon `weighted-2` has an advantage over `simple-2`. Comparing the different algorithms, we see that putting buffer on different activity types is better than only doing so for transfer activities.

COMPUTING TIMES To conclude the evaluation of the algorithms, we examine their computing times. Since we pursued iterative approaches, the computing times highly depend on the number of iterations the algorithms need. Hence, for every algorithm we plotted histograms of the number of iterations until termination which are shown in Figure 7.7. We always fixed the length K of the planning horizon as well as one of the options `simple`, `weighted` and `tight` and looked at the number of iterations for all combinations of the other parameters.

Hence, the sample size is

$$|\{50, 100, 150, 200\} \times \{\texttt{0}, \texttt{1}, \texttt{2}\} \times \{\texttt{smallest}, \texttt{quotient}\} \times \{0.2, 0.05, 0.1\}| = 72$$

for Algorithm 2 and

$$|\{50, 100, 150, 200\} \times \{\texttt{0}, \texttt{1}, \texttt{2}\} \times \{0.2, 0.05, 0.1\}| = 36$$

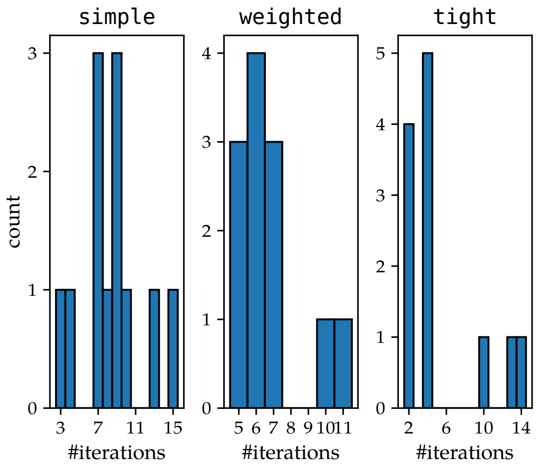
for Algorithm 3. We can clearly see that for the option `tight` we have the lowest number of iterations. In most cases the algorithms terminate in the second iteration, meaning that they return the PESP solution. This fits to the previous results we looked at. For `simple` the number of iterations is only slightly higher: most of the time we have three iterations. Since we have seen that we can get good results for this option, this means that the algorithm finds a good solution very early, but most of the time is not able to improve this solution any further. For `weighted` the number of iterations is higher: we often have six or even more iterations. Since the quality of the solutions is similar to those for `simple` this means that instead of one big improvement we have several smaller improvements.

For Algorithm 1 the results are less clear. Here, we have a much larger variance in the number of iterations, which could be due to the much smaller sample size of

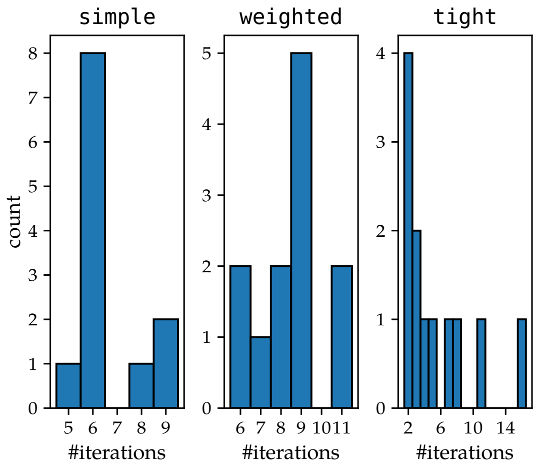
$$|\{50, 100, 150, 200\} \times \{\texttt{0}, \texttt{1}, \texttt{2}\}| = 12$$

since we have less variants of this algorithm.

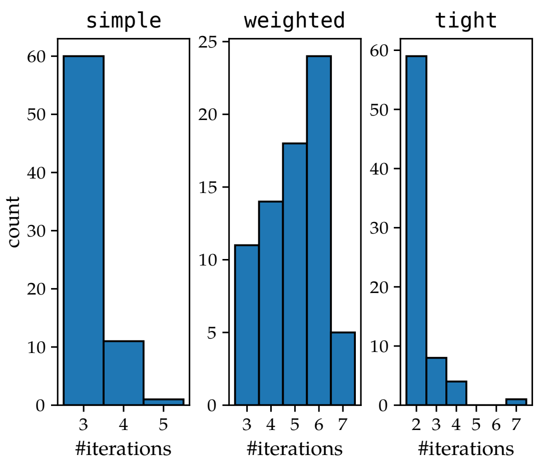
Finally, we look at the computing times of the heuristics in Figure 7.8, where we restrict the attention to the options which proved useful, namely `simple-2` and `weighted-2`. All instances could be solved within a few minutes. As expected, both the length of the planning horizon and the number of scenarios have a significant impact on the computing time. Comparing the different heuristics, we can see that they perform similarly, with the option `simple` being a bit faster than `weighted`, which is in line with our previous observations on the number of iterations. However, for Algorithm 3 some variants of `weighted` are as fast as those using `simple`.



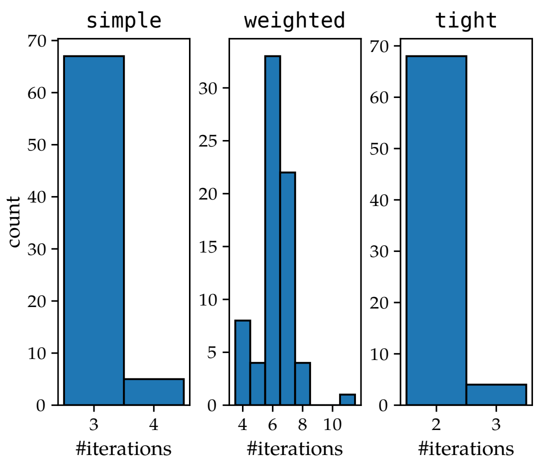
(a) Algorithm 1 for $K = 4$.



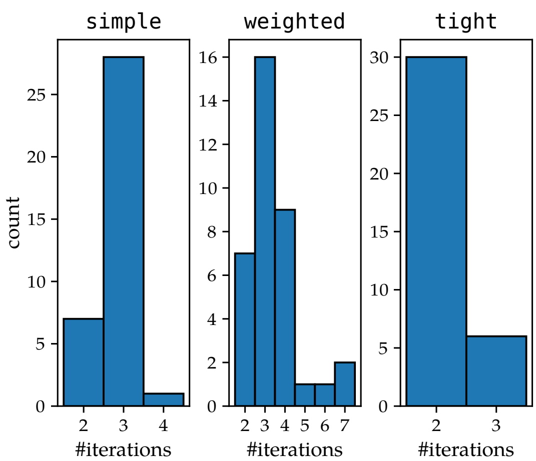
(b) Algorithm 1 for $K = 12$.



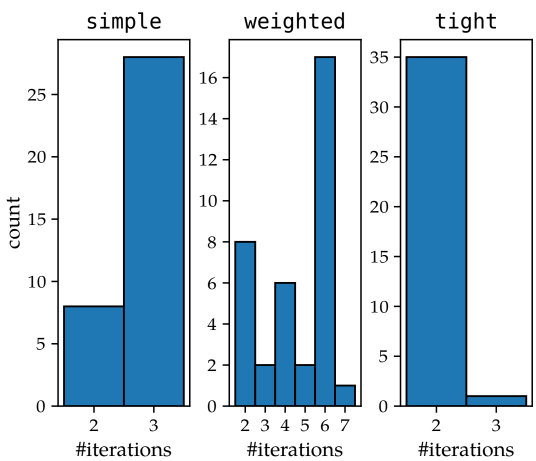
(c) Algorithm 2 for $K = 4$.



(d) Algorithm 2 for $K = 12$.



(e) Algorithm 3 for $K = 4$.



(f) Algorithm 3 for $K = 12$.

Figure 7.7: Number of iterations until termination.

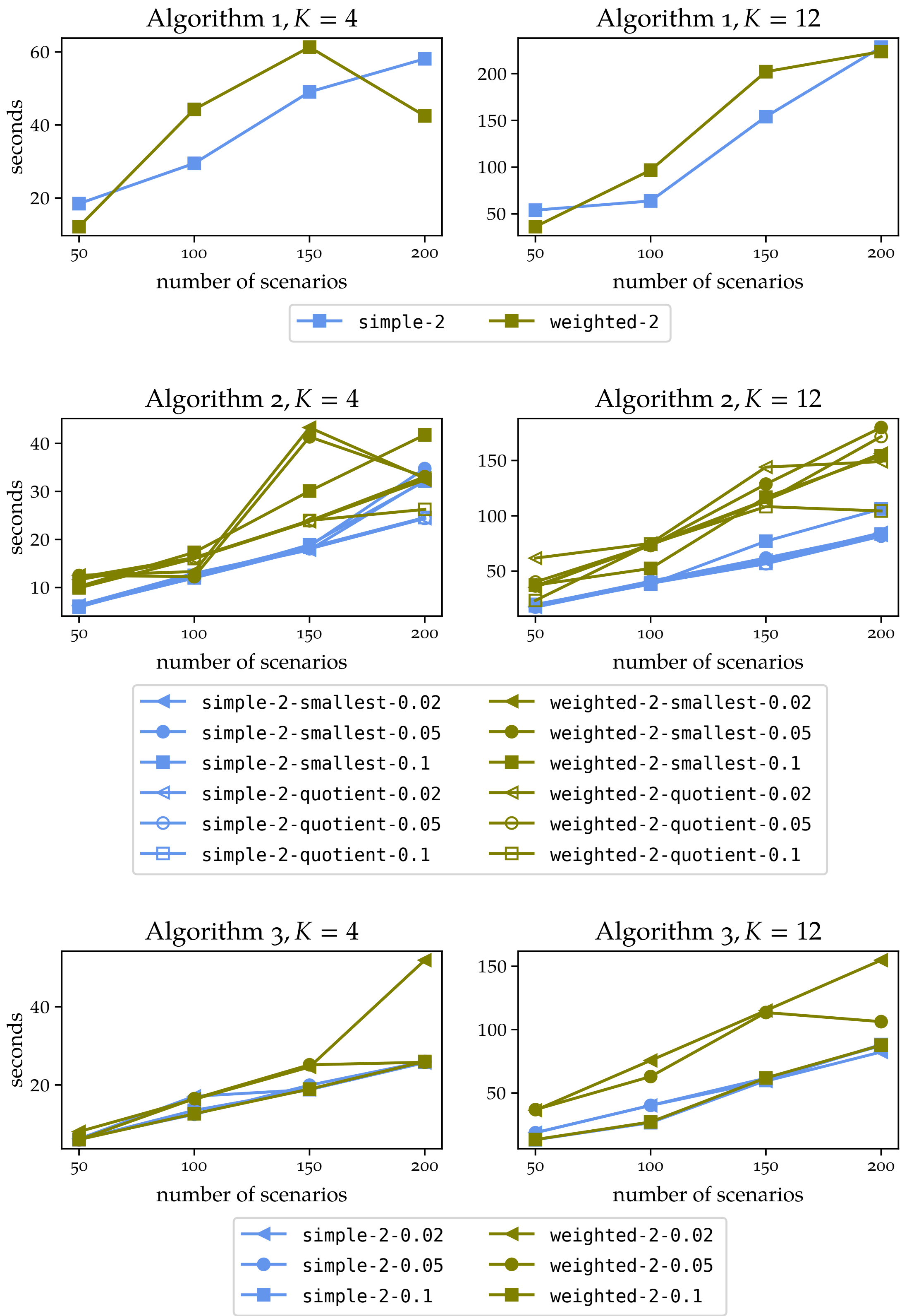


Figure 7.8: Computing times of heuristics for selected parameter combinations.

CONCLUSION

We have seen that simple heuristics iteratively solving the timetabling and the delay management subproblems can yield good results if the algorithm parameters are chosen with care – often yielding an improvement of the real travel time of around 5% compared to the PESP solution. On the used dataset the heuristics run reasonably fast. Due to the modularity of the algorithms it is even possible to speed them up by using heuristics for the single subproblems.

An obvious line for further research lies in the modification of the algorithms, adding even more options for the different parameters and investigating which of these are reasonable. Furthermore, it could be interesting to have a look at different types of heuristics, working more with the MIP formulations instead of the iterative approach we pursued, or solving the problem for reasonably chosen subsets of $\mathcal{U}$.

8 DIFFERENT ROBUSTNESS CONCEPTS APPLIED TO PERIODIC TIMETABLING

After having focussed on recoverable robustness so far, the goal of this chapter is to also apply other robustness concepts to the periodic timetabling problem, namely strict robustness, light robustness and adjustable robustness, which were briefly introduced in Section 2.5. For recoverable robustness we were in particular interested in allowing an aperiodic recovery by using delay management as recovery algorithm. By doing so, we make sure that the aperiodicity of the source delays is taken into account. In this chapter we investigate whether models using the other robustness concepts are also able to handle the aperiodicity. Another aim is to compare the models with respect to the real travel time of their solutions to see if some of them are superior to others.

OUTLINE In Sections 8.1 to 8.3 we apply the mentioned robustness concepts to the periodic timetabling problem. We show that some of them are better able to take the aperiodicity of the source delays into account than others. We compare the different models with respect to their real travel time, first theoretically in Section 8.4 and afterwards in a computational study in Section 8.5.

In this chapter we assume that we only have source delays on the activities, not on the events, i.e. $d_i^r = 0$ for all $r \in \mathcal{U}, i \in \mathcal{E}$. This is due to the fact that for some of the considered robustness concepts it does not make sense to consider this kind of uncertainty: for example, there cannot exist a strictly robust timetable if we have a source delay $d_i^r > 0$ for an event i in some scenario r. Recall from Chapter 3 that we can theoretically transform the event delays to activity delays. However, this might not be realistic, since the virtual activities which have to be added model processes outside of the transport system. Another assumption in this chapter is that the network does not have headway activities, which is for example reasonable for a bus network.

Recall the uncertain PESP from Chapter 3:

$$\begin{aligned} \min \quad & f^{\text{PESP}}(\pi) && \text{(PESP(r))} \\ \text{s.t.} \quad & g^{U}(\pi) \le 0 \\ & g^{L}(\pi, r) \le 0 \\ & \pi \in \{0, \dots, T-1\}^{\mathcal{E}}, \end{aligned}$$

with

$$g^L(\pi, r) := (g^L(\pi) + d)_{d \in \mathcal{U}^r}$$

and

$$\mathcal{U}^r := \{(d^r_{\underline{a}_s})_{\underline{a}\in\underline{\mathcal{A}}} : s \in \{1,\dots,K\}\}.$$

8.1 STRICT ROBUSTNESS

Applying strict robustness to the periodic timetabling problem yields the *strictly robust periodic timetabling problem*.

STRICTLY ROBUST PERIODIC TIMETABLING (SRPT)

Input: Periodic EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$.

Task: Find a periodic timetable π which is feasible in every scenario $r \in \mathcal{U}$ such that the travel time of π is minimal.

Writing the strictly robust counterpart as an IP we get the following form:

$$\begin{aligned} \min\quad & f^{\text{PESP}}(\pi) \\ \text{s.t.}\quad & g^U(\pi) \le 0 \\ & g^L(\pi, r) \le 0 && r \in \mathcal{U} \\ & \pi \in \{0,\dots,T-1\}^{|\underline{\mathcal{E}}|} \end{aligned}$$

and plugging in the definitions yields

$$\min \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\pi_j - \pi_i + z_{\underline{a}}T) \tag{SRPT}$$

$$\text{s.t. } \pi_j - \pi_i + z_{\underline{a}}T \le U_{\underline{a}} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}} \tag{8.1}$$

$$\pi_j - \pi_i + z_{\underline{a}}T \ge L_{\underline{a}} + d^r_{\underline{a}_s} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}}, s \in \{1,\dots,K\}, r \in \mathcal{U} \tag{8.2}$$

$$z_{\underline{a}} \in \mathbb{Z} \qquad \underline{a} \in \underline{\mathcal{A}} \tag{8.3}$$

$$\pi_i \in \{0,\dots,T-1\} \qquad i \in \underline{\mathcal{E}}. \tag{8.4}$$

The variables and the objective function are the same as in PESP. The upper bounds are taken into account by Constraints (8.1). Constraints (8.2) ensure that for all activities the lower bounds and all possible source delays are respected. Note that for every periodic activity the delays of all corresponding aperiodic activities need to be considered here. The timetable π in an optimal solution is denoted by π^{SR}.

We can equivalently write (8.2) as

$$\pi_j - \pi_i + z_{\underline{a}}T \ge L_{\underline{a}} + \max_{r\in\mathcal{U}} \max_{s\in\{1,\dots,K\}} d^r_{\underline{a}_s} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}}$$

and obtain the following formulation:

$$\min \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\pi_j - \pi_i + z_{\underline{a}}T) \qquad \text{(SRPT')}$$

$$\text{s.t. } \pi_j - \pi_i + z_{\underline{a}}T \geq L_{\underline{a}} + \max_{r\in\mathcal{U}} \max_{s\in\{1,\dots,K\}} d^r_{\underline{a}_s} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}} \qquad (8.5)$$

$$(8.1), (8.3), (8.4).$$

Remark 8.1. If we had not used the definition for $\mathcal{U}^r$ from above, but instead $\mathcal{U}^r := \{(\max_{s\in\{1,\dots,K\}} d^r_{\underline{a}_s})_{\underline{a}\in\underline{\mathcal{A}}}\}$, we would directly have received SRPT′ as the strictly robust counterpart. Since SRPT and SRPT′ are equivalent, this means that both definitions of $\mathcal{U}^r$ lead to the same strictly robust counterpart. Thus, we can say that it actually makes no difference if the source delays are periodic or aperiodic, since for every $\underline{a} \in \underline{\mathcal{A}}$ only the maximum delay over all corresponding aperiodic activities $\max_{s\in\{1,\dots,K\}} d^r_{\underline{a}_s}$ is relevant. We can hence conclude that strict robustness is not suitable to take the aperiodicity of the delays into account.

Remark 8.2. SRPT′ is again a PESP with lower bounds $L_{\underline{a}} + \max_{r\in\mathcal{U}} \max_{s\in\{1,\dots,K\}} d^r_{\underline{a}_s}$ for $\underline{a} = (i,j) \in \underline{\mathcal{A}}$. Hence, it can be solved using the same methods as solving PESP.

The strictly robust model has another drawback: often, there does not exist a strictly robust timetable.

Lemma 8.3. *If there are some $\underline{a} \in \underline{\mathcal{A}}$, $s \in \{1,\dots,K\}$ and $r \in \mathcal{U}$ with $d^r_{\underline{a}_s} > U_{\underline{a}} - L_{\underline{a}}$, a strictly robust timetable does not exist.*

Proof. If $d^r_{\underline{a}_s} > U_{\underline{a}} - L_{\underline{a}}$, (8.1) and (8.2) contradict each other. Hence, SRPT is infeasible. □

8.2 LIGHT ROBUSTNESS

For the context of periodic timetabling we first have to choose a nominal scenario r. This is the undisturbed setting without any delays, i.e. the optimal solution for $\hat{r}$ is the PESP solution π^{PESP}. We present the *lightly robust periodic timetabling problem*:

Lightly robust periodic timetabling (LRPT)

Input: Periodic EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$, optimal nominal value $f^{\text{PESP}}(\pi^{\text{PESP}})$, $\rho \geq 0$.

Task: Find a periodic timetable π which is feasible in the undisturbed scenario $\hat{r}$ and has a travel time of at most $f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho$ in all scenarios $r \in \mathcal{U}$ such that the grade of infeasibility is minimal.

The lightly robust counterpart can be written as follows:

$$
\begin{aligned}
\min \quad & \|\gamma\| \\
\text{s.t.} \quad & f^{\text{PESP}}(\pi) \le f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho \\
& g^{U}(\pi) \le 0 \\
& g^{L}(\pi) \le 0 \\
& g^{L}(\pi, r) \le \gamma && r \in \mathcal{U} \\
& \pi_i \in \{0, \dots, T-1\}^{|\mathcal{E}|} \\
& \gamma \in \mathbb{R}^{K|\underline{\mathcal{A}}|}.
\end{aligned}
$$

Again, we plug in the definitions and obtain

$$
\begin{aligned}
\min \quad & \|\gamma\| && && (\text{LRPT}(\rho)) \\
\text{s.t.} \quad & \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\pi_j - \pi_i + z_{\underline{a}}T) \le f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho && && (8.6) \\
& \pi_j - \pi_i + z_{\underline{a}}T \le U_{\underline{a}} && \underline{a} = (i,j) \in \underline{\mathcal{A}} && (8.7) \\
& \pi_j - \pi_i + z_{\underline{a}}T \ge L_{\underline{a}} && \underline{a} = (i,j) \in \underline{\mathcal{A}} && (8.8) \\
& \pi_j - \pi_i + z_{\underline{a}}T + \gamma_{\underline{a}_s} \ge L_{\underline{a}} + d^r_{\underline{a}_s} && \underline{a} = (i,j) \in \underline{\mathcal{A}}, s \in \{1, \dots, K\}, \\
& && r \in \mathcal{U} && (8.9) \\
& \gamma_{\underline{a}_s} \ge 0 && \underline{a} \in \underline{\mathcal{A}}, s \in \{1, \dots, K\} && (8.10) \\
& z_{\underline{a}} \in \mathbb{Z} && \underline{a} \in \underline{\mathcal{A}} && (8.11) \\
& \pi_i \in \{0, \dots, T-1\} && i \in \underline{\mathcal{E}}. && (8.12)
\end{aligned}
$$

The PESP objective function value (i.e. the travel time) in the nominal (undelayed) scenario is bounded by $f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho$, which is ensured by Constraints (8.6). Constraints (8.7) and (8.8) are the same as in PESP since the timetable has to be feasible in the nominal scenario. For all scenarios $r \in \mathcal{U}$ the constraints respecting lower bounds and source delays are relaxed by allowing some deviation γ, see (8.9). The objective function minimises some norm of the vector γ. We denote the timetable π in an optimal solution by $\pi^{\text{LR},\rho}$.

Interestingly, if we rewrite the constraints

$$
\pi_j - \pi_i + z_{\underline{a}}T \ge L_{\underline{a}} + d^r_{\underline{a}_s} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}}, s \in \{1, \dots, K\}
$$

in the uncertain problem PESP(r) to

$$
\pi_j - \pi_i + z_{\underline{a}}T \ge L_{\underline{a}} + \max_{s \in \{1,\dots,K\}} d^r_{\underline{a}_s} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}}
$$

and then use this equivalent formulation to derive the lightly robust counterpart, we get

$$\min \quad \|\gamma\| \qquad \text{(LRPT}'(\rho))$$

$$\text{s.t.} \quad \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}}(\pi_j - \pi_i + z_{\underline{a}}T) \leq f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho \qquad (8.13)$$

$$\pi_j - \pi_i + z_{\underline{a}}T \leq U_{\underline{a}} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}} \qquad (8.14)$$

$$\pi_j - \pi_i + z_{\underline{a}}T \geq L_{\underline{a}} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}} \qquad (8.15)$$

$$\pi_j - \pi_i + z_{\underline{a}}T + \gamma_{\underline{a}} \geq L_{\underline{a}} + \max_{s\in\{1,\ldots,K\}} d^r_{\underline{a}_s} \qquad \underline{a} = (i,j) \in \underline{\mathcal{A}}, r \in \mathcal{U} \qquad (8.16)$$

$$\gamma_{\underline{a}} \geq 0 \qquad \underline{a} \in \underline{\mathcal{A}} \qquad (8.17)$$

$$z_{\underline{a}} \in \mathbb{Z} \qquad \underline{a} \in \underline{\mathcal{A}} \qquad (8.18)$$

$$\pi_i \in \{0,\ldots,T-1\} \qquad i \in \underline{\mathcal{E}}. \qquad (8.19)$$

This is also the formulation we would receive when using the alternative definition of $\mathcal{U}^r$ from Remark 8.1. In this formulation, we only have a slack variable $\gamma_{\underline{a}}$ for every $\underline{a} \in \underline{\mathcal{A}}$, not for every corresponding aperiodic activity $\underline{a}_s$. Furthermore, we only consider the maximum $\max_{s\in\{1,\ldots,K\}} d^r_{\underline{a}_s}$. In general, LRPT$(\rho)$ and LRPT$'(\rho)$ are not equivalent.

For the lightly robust model we have to choose a norm $\|\cdot\|$ for the objective function. In fact, the choice of the norm can have a large impact on the model.

Remark 8.4 (Choice of the norm)**.** When using $\|\cdot\| = \|\cdot\|_\infty$, the model does not take the aperiodicity of the source delays into account, since for every $a \in \underline{\mathcal{A}}$ only the maximum $\max_{s\in\{1,\ldots,K\}} d^r_{\underline{a}_s}$ is relevant. We have seen this already for strict robustness. However, for $\|\cdot\| = \|\cdot\|_1$, indeed all source delays are relevant. Another reasonable choice for evaluating the vector γ is to take the activity weights into account: $\|\gamma\|_w := \sum_{\underline{a}\in\underline{\mathcal{A}},s\in\{1,\ldots,K\}} w_{\underline{a}}\gamma_{\underline{a}_s}$. Note that if there are activities with weight zero (which might be reasonable, e.g. for waiting activities), this is actually not a norm, since then we can have $\|\gamma\|_w = 0$ even if $\gamma \neq 0$. However, we do not need the norm property and it still fulfils the purpose of evaluating the vector γ. Unless stated otherwise, we use $\|\cdot\| = \|\cdot\|_1$.

Lemma 8.5. *For $\|\cdot\| = \|\cdot\|_\infty$, LRPT$(\rho)$ and LRPT$'(\rho)$ are equivalent.*

Proof. If we have a feasible solution (π, z, γ) to LRPT(ρ), we can set $\tilde{\gamma}_{\underline{a}} := \max_{s\in\{1,\ldots,K\}} \gamma_{\underline{a}_s}$ for $\underline{a} \in \underline{\mathcal{A}}$. Then $(\pi, z, \tilde{\gamma})$ is a feasible solution to LRPT$'(\rho)$ with the same objective value since

$$\|(\gamma_{\underline{a}_s})_{\underline{a}\in\underline{\mathcal{A}},s\in\{1,\ldots,K\}}\|_\infty = \max_{\underline{a}\in\underline{\mathcal{A}}} \max_{s\in\{1,\ldots,K\}} \gamma_{\underline{a}_s} = \max_{\underline{a}\in\underline{\mathcal{A}}} \tilde{\gamma}_{\underline{a}} = \|(\tilde{\gamma}_{\underline{a}})_{\underline{a}\in\underline{\mathcal{A}}}\|_\infty.$$

On the other hand, for every feasible solution $(\pi, z, \tilde{\gamma})$ to LRPT$'(\rho)$ we obtain a feasible solution (π, z, γ) to LRPT(ρ) with the same objective value by setting $\gamma_{\underline{a}_s} := \gamma_{\underline{a}}$ for all $s \in \{1,\ldots,K\}$. □

In the following we will use LRPT(ρ) when talking about lightly robust timetables. We will sometimes omit the parameter ρ and simply write LRPT.

8.3 ADJUSTABLE ROBUSTNESS

When applying adjustable robustness to periodic timetabling, we have to specify how we partition the variables into "here-and-now variables" and "wait-and-see variables". First, we introduce the necessary notation.

Let $\underline{\mathcal{E}} = \underline{\mathcal{E}}^1 \cup \underline{\mathcal{E}}^2$ be a partition of the events. For $i \in \underline{\mathcal{E}}^1$ we treat π_i as here-and-now variable, while the times for those events in $\underline{\mathcal{E}}^2$ are wait-and-see variables, which can depend on the scenario.

The *adjustable robust periodic timetabling problem* then is as follows:

ADJUSTABLE ROBUST PERIODIC TIMETABLING (ARPT)

Input: Periodic EAN $\underline{\mathcal{N}} = (\underline{\mathcal{E}}, \underline{\mathcal{A}})$ with period T, uncertainty set $\mathcal{U}$, subset $\underline{\mathcal{E}}^2 \subseteq \underline{\mathcal{E}}$.

Task: Find times π_i for $i \in \underline{\mathcal{E}}^1 = \underline{\mathcal{E}} \backslash \underline{\mathcal{E}}^2$ and π_i^r for $i \in \underline{\mathcal{E}}^2, r \in \mathcal{U}$ such that for every $r \in \mathcal{U}$ the resulting timetable is periodic and feasible and the travel time in the worst-case is minimal.

Due to the partition of the events we obtain four different cases for the activities, which yield the following subsets:

$$\underline{\mathcal{A}}^1 = \{(i,j) \in \underline{\mathcal{A}} : i,j \in \underline{\mathcal{E}}^1\},$$
$$\underline{\mathcal{A}}^2 = \{(i,j) \in \underline{\mathcal{A}} : i \in \underline{\mathcal{E}}^1, j \in \underline{\mathcal{E}}^2\},$$
$$\underline{\mathcal{A}}^3 = \{(i,j) \in \underline{\mathcal{A}} : i \in \underline{\mathcal{E}}^2, j \in \underline{\mathcal{E}}^1\},$$
$$\underline{\mathcal{A}}^4 = \{(i,j) \in \underline{\mathcal{A}} : i,j \in \underline{\mathcal{E}}^2\}.$$

Consequently, for $\underline{a} \in \underline{\mathcal{A}}^1$ the modulo variable $z_{\underline{a}}$ does not depend on the realised scenario $r \in \mathcal{U}$ (here-and-now), while for $\underline{a} \in \underline{\mathcal{A}}^2 \cup \underline{\mathcal{A}}^3 \cup \underline{\mathcal{A}}^4$ we have variables $z_{\underline{a}}^r$ for $r \in \mathcal{U}$ (wait-and-see). Let $n_1 := |\underline{\mathcal{E}}^1|, n_2 := |\underline{\mathcal{E}}^2|$.

Denoting

$$\begin{aligned} f(\pi, \pi^r) = &\sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}^1} w_{\underline{a}}(\pi_j - \pi_i + z_{\underline{a}}T) + \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}^2} w_{\underline{a}}(\pi_j^r - \pi_i + z_{\underline{a}}^r T) \\ &+ \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}^3} w_{\underline{a}}(\pi_j - \pi_i^r + z_{\underline{a}}^r T) + \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}^4} w_{\underline{a}}(\pi_j^r - \pi_i^r + z_{\underline{a}}^r T), \end{aligned}$$

the *adjustable robust periodic timetabling problem* $(ARPT)$ can be be written as follows:

$$\begin{aligned} \min \quad & c \\ \text{s.t.} \quad & f(\pi, \pi^r) \le c && r \in \mathcal{U} \\ & g^U(\pi, \pi^r) \le 0 && r \in \mathcal{U} \\ & g^L(\pi, \pi^r, r) \le 0 && r \in \mathcal{U} \\ & \pi \in \mathbb{R}^{n_1} \\ & \pi^r \in \mathbb{R}^{n_2} && r \in \mathcal{U}, \end{aligned}$$

which yields

$$\begin{aligned}
\min\quad & c && && (\text{ARPT}(\underline{\mathcal{E}}^2))\\
\text{s.t.}\quad & f(\pi, \pi^r) \leq c && r \in \mathcal{U} && (8.20)\\
& \pi_j - \pi_i + z_{\underline{a}} T \leq U_{\underline{a}} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^1 && (8.21)\\
& \pi_j^r - \pi_i + z_{\underline{a}}^r T \leq U_{\underline{a}} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^2, r \in \mathcal{U} && (8.22)\\
& \pi_j - \pi_i^r + z_{\underline{a}}^r T \leq U_{\underline{a}} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^3, r \in \mathcal{U} && (8.23)\\
& \pi_j^r - \pi_i^r + z_{\underline{a}}^r T \leq U_{\underline{a}} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^4, r \in \mathcal{U} && (8.24)\\
& \pi_j - \pi_i + z_{\underline{a}} T \geq L_{\underline{a}} + d_{\underline{a}_s}^r && \underline{a} = (i,j) \in \underline{\mathcal{A}}^1, s \in \{1, \dots, K\}, r \in \mathcal{U} && (8.25)\\
& \pi_j^r - \pi_i + z_{\underline{a}}^r T \geq L_{\underline{a}} + d_{\underline{a}_s}^r && \underline{a} = (i,j) \in \underline{\mathcal{A}}^2, s \in \{1, \dots, K\}, r \in \mathcal{U} && (8.26)\\
& \pi_j - \pi_i^r + z_{\underline{a}}^r T \geq L_{\underline{a}} + d_{\underline{a}_s}^r && \underline{a} = (i,j) \in \underline{\mathcal{A}}^3, s \in \{1, \dots, K\}, r \in \mathcal{U} && (8.27)\\
& \pi_j^r - \pi_i^r + z_{\underline{a}}^r T \geq L_{\underline{a}} + d_{\underline{a}_s}^r && \underline{a} = (i,j) \in \underline{\mathcal{A}}^4, s \in \{1, \dots, K\}, r \in \mathcal{U} && (8.28)\\
& \pi_i \in \{0, \dots, T-1\} && i \in \underline{\mathcal{E}}^1 && (8.29)\\
& \pi_i^r \in \{0, \dots, T-1\} && i \in \underline{\mathcal{E}}^2, r \in \mathcal{U} && (8.30)\\
& z_{\underline{a}} \in \mathbb{Z} && \underline{a} \in \underline{\mathcal{A}}^1 && (8.31)\\
& z_{\underline{a}}^r \in \mathbb{Z} && \underline{a} \in \underline{\mathcal{A}} \setminus \underline{\mathcal{A}}^1, r \in \mathcal{U}. && (8.32)
\end{aligned}$$

The upper bounds are respected due to Constraints (8.21) to (8.24). Depending on whether the variables are here-and-now or wait-and-see variables, they may or may not depend on r. Analogously, Constraints (8.25) to (8.28) ensure that the timetable respects the lower bounds and source delays for every $r \in \mathcal{U}$. The objective function minimises the travel time in the worst case. We denote the timetable $((\pi_i)_{i \in \underline{\mathcal{E}}^1}, (\pi_i^r)_{i \in \underline{\mathcal{E}}^2}^{r \in \mathcal{U}})$ in an optimal solution by $\pi^{\text{AR}, \underline{\mathcal{E}}^2}$.

Remark 8.6. It has to be discussed which event partition is reasonable. One possibility would be to choose $\underline{\mathcal{E}}^1$ as the events at the main stations, while $\underline{\mathcal{E}}^2$ might be events at small intermediate stations, where only a small number of passengers enter or leave the train, as those might be considered less important. However, such a timetable comes with a drawback for the passengers. If π_i for $i \in \underline{\mathcal{E}}^{\text{dep}}$ was adjustable, this would mean that the passengers do not know the departure time of their train beforehand. Hence, they would have to be at the station early and wait for the departure. Besides being inconvenient, this waiting time at the station would moreover not be counted in the objective function and therefore not accurately represent the actual travel time the passengers face. A way around this is to choose $\underline{\mathcal{E}}^1 = \underline{\mathcal{E}}^{\text{dep}}$ and $\underline{\mathcal{E}}^2 = \underline{\mathcal{E}}^{\text{arr}}$ such that all departure times are reliable and only arrival times may vary. We use this partition later in our comparisons.

Analogously to strict robustness, also in this case we can rewrite some of the constraints and obtain the following equivalent formulation:

$$
\begin{aligned}
\min \quad & c && && (\text{ARPT}'(\underline{\mathcal{E}}^2))\\
\text{s.t.} \quad & \pi_j - \pi_i + z_{\underline{a}}T \geq L_{\underline{a}} + \max_{r\in\mathcal{U}} \max_{s\in\{1,\ldots,K\}} d^r_{\underline{a}_s} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^1 && (8.33)\\
& \pi^r_j - \pi_i + z^r_{\underline{a}}T \geq L_{\underline{a}} + \max_{s\in\{1,\ldots,K\}} d^r_{\underline{a}_s} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^2, r\in\mathcal{U} && (8.34)\\
& \pi_j - \pi^r_i + z^r_{\underline{a}}T \geq L_{\underline{a}} + \max_{s\in\{1,\ldots,K\}} d^r_{\underline{a}_s} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^3, r\in\mathcal{U} && (8.35)\\
& \pi^r_j - \pi^r_i + z^r_{\underline{a}}T \geq L_{\underline{a}} + \max_{s\in\{1,\ldots,K\}} d^r_{\underline{a}_s} && \underline{a} = (i,j) \in \underline{\mathcal{A}}^4, r\in\mathcal{U} && (8.36)\\
& (8.20) - (8.24), (8.29) - (8.32).
\end{aligned}
$$

Remark 8.7. As in Remark 8.1, using the alternative definition of $\mathcal{U}^r$ yields the formulation ARPT'($\underline{\mathcal{E}}^2$), meaning that also for adjustable robustness it makes no difference if we consider periodic or aperiodic delays.

Unfortunately, the adjustable robust model has the same disadvantage as the strictly robust one: it becomes infeasible very easily.

Lemma 8.8. *If there are some $\underline{a} \in \underline{\mathcal{A}}, s \in \{1,\ldots,K\}, r \in \mathcal{U}$ with $d^r_{\underline{a}_s} > U_{\underline{a}} - L_{\underline{a}}$, then no adjustable robust timetable exists.*

Proof. If $d^r_{\underline{a}_s} > U_{\underline{a}} - L_{\underline{a}}$ for some $\underline{a} \in \underline{\mathcal{A}}^1$, (8.21) and (8.25) contradict each other. This holds analogously for $\underline{a} \in \underline{\mathcal{A}}^2 \cup \underline{\mathcal{A}}^3 \cup \underline{\mathcal{A}}^4$ and Constraints (8.22) to (8.24) and (8.26) to (8.28). □

The model depends a lot on the choice of the variable partition. The larger the set $\underline{\mathcal{E}}^2$, the more flexibility the model has. Hence, the travel time in the worst case (neglecting the passengers' waiting times at their first stations) decreases when $\underline{\mathcal{E}}^2$ is increased.

Lemma 8.9. *Let $\underline{\mathcal{E}}^2 \subseteq \underline{\tilde{\mathcal{E}}}^2$. Then the optimal objective value of ARPT($\underline{\tilde{\mathcal{E}}}^2$) is less or equal to the optimal objective value of ARPT($\underline{\mathcal{E}}^2$).*

Proof. Let (π, z) be a feasible solution to ARPT($\underline{\mathcal{E}}^2$). We define

$$
\begin{aligned}
\tilde{\pi}_i &= \pi_i && \text{if } i \in \underline{\tilde{\mathcal{E}}}^1,\\
\tilde{z}_{(i,j)} &= z_{(i,j)} && \text{if } i,j \in \underline{\tilde{\mathcal{E}}}^1,
\end{aligned}
$$

which is well-defined since $\underline{\tilde{\mathcal{E}}}^1 \subseteq \underline{\mathcal{E}}^1$. Furthermore, for $r \in \mathcal{U}$ we set

$$
\begin{aligned}
\tilde{\pi}^r_i &= \begin{cases} \pi^r_i & \text{if } i \in \underline{\mathcal{E}}^2,\\ \pi_i & \text{if } i \in \underline{\tilde{\mathcal{E}}}^2 \backslash \underline{\mathcal{E}}^2, \end{cases}\\
\tilde{z}^r_{(i,j)} &= \begin{cases} z_{(i,j)} & \text{if } (i,j) \in \underline{\mathcal{A}}^1,\\ z^r_{(i,j)} & \text{if } (i,j) \in \underline{\mathcal{A}}^2 \cup \underline{\mathcal{A}}^3 \cup \underline{\mathcal{A}}^4. \end{cases}
\end{aligned}
$$

Then $(\tilde{\pi}, \tilde{z})$ is feasible with respect to $\tilde{\underline{\mathcal{E}}}^2$. We check the constraints exemplary for the case $\underline{a} = (i,j) \in \tilde{\underline{\mathcal{A}}}^2$, i.e. $i \in \tilde{\underline{\mathcal{E}}}^1, j \in \tilde{\underline{\mathcal{E}}}^2$. Then $i \in \underline{\mathcal{E}}^1$ and by definition we have $\tilde{\pi}_i = \pi_i$. For $r \in \mathcal{U}$ we obtain

$$\tilde{\pi}_j^r - \tilde{\pi}_i + \tilde{z}_{\underline{a}}^r T = \begin{cases} \pi_j^r - \pi_i + z_{\underline{a}T}^r & \text{if } j \in \underline{\mathcal{E}}^2 (\text{i.e. } \underline{a} \in \underline{\mathcal{A}}^2), \\ \pi_j - \pi_i + z_{\underline{a}T} & \text{if } j \in \underline{\mathcal{E}}^1 (\text{i.e. } \underline{a} \in \underline{\mathcal{A}}^1), \end{cases}$$

which is in the feasible interval due to Constraints (8.21) and (8.22). The other constraints can be checked in the same manner.

Furthermore, for every $r \in \mathcal{U}$ the value on the left hand side of (8.20) is the same for both solutions since every $\underline{a} \in \underline{\mathcal{A}}$ is part of exactly one of the four sums. Hence, also the objective value stays the same and the claim follows. □

In the following we will sometimes omit the parameter $\underline{\mathcal{E}}^2$ in the notation and simply write ARPT.

8.4 COMPARISON

In this section we compare RRPT and the models introduced in the previous sections. Note that the output of the four models is different: the strictly, lightly and recoverable robust models determine *one* timetable which is defined on all events. The strictly robust timetable is feasible for all scenarios, while the lightly and recoverable robust timetables only need to be feasible for the undisturbed scenario. For the latter, we already determine disposition timetables separately for each scenario. The adjustable robust model "cheats" a bit by only determining a timetable for those events in $\underline{\mathcal{E}}^1$ and leaving it open for the events in $\underline{\mathcal{E}}^2$.

8.4.1 *Relations between the models*

We first analyse how the models relate to each other for specific parameter choices. For an overview of the parameters see Table 8.1.

Model	Parameter	Optimal timetable	Objective function
PESP	-	π^{PESP}	nominal travel time
SRPT	-	π^{SR}	nominal/real travel time
LRPT(ρ)	ρ	$\pi^{\text{LR},\rho}$	grade of infeasibility
ARPT($\underline{\mathcal{E}}^2$)	$\underline{\mathcal{E}}^2$	$\pi^{\text{AR},\underline{\mathcal{E}}^2}$	real travel time
RRPT	-	π^{RR}	real travel time
RRPT(α, β)	α, β	$\pi^{\text{RR},\alpha,\beta}$	nominal travel time
RRPT($\bar{f}$)	$\bar{f}$	$\pi^{\text{RR},\bar{f}}$	worst-case delay

Table 8.1: The different (robust) periodic timetabling models and their parameters. The notations can also be found on page 127 and following.

Lemma 8.10. *Every strictly robust timetable is also adjustable robust.*

Proof. Let $\tilde{\pi}$ be a strictly robust timetable. Setting $\pi_i := \tilde{\pi}_i$ for $i \in \underline{\mathcal{E}}^1$ and $\pi_i^r := \tilde{\pi}_i$ for $i \in \underline{\mathcal{E}}^2, r \in \mathcal{U}$ yields a feasible solution to ARPT (with appropriately chosen z). □

Lemma 8.11. *For $\underline{\mathcal{E}}^2 = \emptyset$, ARPT is equivalent to SRPT.*

Proof. For the case $\underline{\mathcal{E}}^1 = \underline{\mathcal{E}}, \underline{\mathcal{E}}^2 = \emptyset$ it follows $\underline{\mathcal{A}}^1 = \underline{\mathcal{A}}$ and $\underline{\mathcal{A}}^2 = \underline{\mathcal{A}}^3 = \underline{\mathcal{A}}^4 = \emptyset$. In the formulation of ARPT most constraints vanish and what remains is exactly SRPT. □

Next we show that, depending on the choice of ρ, the problem LRPT is related to either PESP or SRPT, and can hence be seen as "in-between" these two models.

Lemma 8.12. *If $\rho \geq f^{\text{PESP}}(\pi^{\text{SR}}) - f^{\text{PESP}}(\pi^{\text{PESP}})$ and SRPT is feasible, every optimal solution to LRPT is strictly robust.*

Proof. Since for the chosen ρ Constraint (8.6) poses no restriction on the nominal travel time any more, π^{SR} with $\gamma = 0$ is feasible for LRPT. Since the objective is to minimise the norm of γ, it is also optimal. Hence, every optimal solution has objective value zero and thus fulfils $\gamma = 0$. This implies that (8.1) to (8.4) are fulfilled, so every optimal solution is strictly robust. □

Remark 8.13. Note that in the proof of Lemma 8.12 we need that $\|\gamma\| = 0$ implies $\gamma = 0$, which is fulfilled if $\|\cdot\|$ really is a norm. However, for $\|\cdot\|_w$ from Remark 8.4 this is only true if we have strictly positive passenger weights w on every activity.

Lemma 8.14. *If $\rho = 0$, every feasible solution to LRPT is an optimal solution to PESP.*

Proof. Due to Constraints (8.7) and (8.8), any feasible solution is also feasible for PESP. If $\rho = 0$, (8.6) additionally implies that it is optimal for PESP. □

Next we consider the bounded versions of the recoverable robust model.

Lemma 8.15. *Every feasible solution to SRPT with $f^{\text{PESP}}(\pi^{\text{SR}}) \leq \bar{f}$ yields an optimal solution to RRPT($\bar{f}$).*

Proof. Every feasible solution to SRPT with $f^{\text{PESP}}(\pi^{\text{SR}}) \leq \bar{f}$ also yields a feasible solution to RRPT($\bar{f}$) without any delay. Since the objective is to minimise f^{del}, this solution is indeed optimal. □

Lemma 8.16. *If α and β are very large, RRPT(α, β) is equivalent to PESP. For example, this is the case for $\alpha = M''K \cdot \sum_{i \in \underline{\mathcal{E}}} w_i + M'' \cdot \sum_{a \in \mathcal{A}_{\text{out}}} w_a$ and $\beta = \sum_{a \in \mathcal{A}_{\text{transfer}}} w_a$.*

Proof. By Corollary 4.10 we know that PESP is equivalent to PTTA. Hence, it suffices to show the equivalence of RRPT(α, β) and PTTA.
Since PTTA is a subproblem of RRPT(α, β) with the same objective function, one direction is clear. On the other hand, if we have a solution to PTTA, we can simply extend it to a solution to RRPT(α, β) by doing delay management for every scenario. As shown in Lemma 5.7, there is always an optimal solution with $x_i^r - \pi_i \leq M''$ for all $i \in \mathcal{E}, r \in \mathcal{U}$, so the α as chosen above poses no restriction. Obviously also β is an upper bound on the number of missed transfers. Hence, there is always a feasible solution to RRPT(α, β). Since the objective value stays the same, the claim follows. □

8.4.2 *Evaluation functions*

The different models have different objective functions. To be able to compare them, we evaluate their real travel time. We briefly repeat the necessary notation from Definition 5.1.

For the nominal travel time, note that we consider the whole planning horizon, i.e. for models using the periodic network the duration of every arc has to be multiplied by K. In particular, if $\tilde{\pi}$ is a feasible timetable for PESP, SRPT or LRPT, and π the corresponding timetable in the rolled out network, we have

$$f^{\text{nom}}(\pi) = f^{\text{nom}}(\tilde{\pi}) = K \cdot f^{\text{PESP}}(\tilde{\pi}) = K \cdot \sum_{\underline{a}=(i,j)\in\underline{\mathcal{A}}} w_{\underline{a}} \cdot (\tilde{\pi}_j - \tilde{\pi}_i + z_{\underline{a}}(\tilde{\pi})T),$$

where $z_{\underline{a}}(\tilde{\pi}) := \min\{z \in \mathbb{Z} : \tilde{\pi}_j - \tilde{\pi}_i + zT \geq L_{\underline{a}}\}$. Note that for simplicity we use the same notation for the nominal travel time of the timetable in the periodic and in the rolled out network, but this should not lead to any confusion.

The worst-case delay can be calculated by computing an optimal disposition timetable (x^r, y^r) for every $r \in \mathcal{U}$ and taking the maximum

$$f^{\text{del}}(\pi) = f^{\text{del}}(\tilde{\pi}) = \max_{r\in\mathcal{U}} \sum_{i\in\mathcal{E}(\pi)} w_i(x_i^r - \pi_i) + \sum_{a=(i,j)\in\mathcal{A}_{\text{out}}(\pi)} w_a(x_j^r - \pi_j) + T \sum_{a\in\mathcal{A}_{\text{transfer}}(\pi)} w_a y_a^r.$$

These two components give the real travel time

$$f^{\text{real}}(\tilde{\pi}) = f^{\text{real}}(\pi) = f^{\text{nom}}(\pi) + f^{\text{del}}(\pi).$$

Note that for the adjustable robust case we can neither define the nominal travel time nor the delay, since we do not fix a timetable for the events in $\underline{\mathcal{E}}^2$. Nevertheless, we can compute the real travel time which in this case is just the objective function value multiplied by K, i.e. $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) = K \cdot c$.

8.4.3 *Performance of the robustness concepts*

The goal of this section is to establish an order for the real travel time of the optimal solutions to the timetable models we have introduced so far. This means that for every pair of models, we either prove that one of them is better than the other, or we demonstrate that none of them is superior in general.

The recoverable robust model yields better timetables than the non-robust model PESP:

Lemma 8.17. *It holds* $f^{\text{real}}(\pi^{\text{PESP}}) \geq f^{\text{real}}(\pi^{\text{RR}})$.

Proof. By Lemma 4.11 the solution π^{PESP} can be rolled out to a feasible solution to PTTA. For the rolled out timetable we can compute a disposition timetable (x^r, y^r) for every delay scenario $r \in \mathcal{U}$, which yields a solution to RRPT. Since RRPT minimises the real travel time, it follows $f^{\text{real}}(\pi^{\text{PESP}}) \geq f^{\text{real}}(\pi^{\text{RR}})$. □

Lemma 8.18. *Comparing strictly robust timetables to timetables using other robustness concepts, we get the following inequalities:*

(a) $f^{\text{real}}(\pi^{\text{SR}}) \geq f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2})$ *for all* $\underline{\mathcal{E}}^2 \subseteq \underline{\mathcal{E}}$.

(b) $f^{\text{nom}}(\pi^{\text{SR}}) = f^{\text{real}}(\pi^{\text{SR}}) \geq f^{\text{real}}(\pi^{\text{RR}}) \geq f^{\text{nom}}(\pi^{\text{RR}})$.

(c) $f^{\text{nom}}(\pi^{\text{SR}}) \geq f^{\text{nom}}(\pi^{\text{RR},\alpha,\beta})$ *for all* $\alpha, \beta \geq 0$.

(d) *If RRPT*$(\bar{f})$ *is feasible for some* $\bar{f}$, *then there is an optimal solution with* $f^{\text{nom}}(\pi^{\text{SR}}) \geq f^{\text{nom}}(\pi^{\text{RR},\bar{f}})$.

Proof. (a) This follows directly from Lemma 8.10 and the fact that both SRPT and ARPT minimise the real travel time.

(b) Since a strictly robust timetable is by definition feasible for all $r \in \mathcal{U}$, we have $f^{\text{del}}(\pi^{\text{SR}}) = 0$ and hence $f^{\text{nom}}(\pi^{\text{SR}}) = f^{\text{real}}(\pi^{\text{SR}})$.
The optimal solution to SRPT (π^{SR}, z) is feasible for PESP and can therefore be rolled out to a feasible solution to PTTA with objective value $f^{\text{nom}}(\pi^{\text{SR}})$. Furthermore, by definition of SRPT, (π^{SR}, z) is feasible for all delay scenarios $r \in \mathcal{U}$. Hence, we can set $x_i^r = \pi_i^{\text{SR}}$ for $i \in \mathcal{E}$, $y_a^r = 0$ for $a \in \mathcal{A}_{\text{transfer}}$, $H_a^r = 0$ for $a \in \mathcal{A}_{\text{out}}$ and $Z_1 = Z_2 = Z = 0$.
This yields a feasible solution to RRPT, which implies $f^{\text{real}}(\pi^{\text{SR}}) \geq f^{\text{real}}(\pi^{\text{RR}})$.
The last inequality is clear since $f^{\text{del}} \geq 0$.

(c) Since it has no delay, the solution constructed above is also feasible for RRPT(α, β), so $f^{\text{nom}}(\pi^{\text{RR},\alpha,\beta}) \leq f^{\text{nom}}(\pi^{\text{SR}})$.

(d) If $\bar{f} \leq f^{\text{nom}}(\pi^{\text{SR}})$, the claim immediately follows since $f^{\text{nom}}(\pi^{\text{RR},\bar{f}}) \leq \bar{f}$ by Constraint (5.31). Otherwise, π^{SR} can be rolled out to a feasible solution to RRPT$(\bar{f})$. Since $f^{\text{del}}(\pi^{\text{SR}}) = 0$, it is even optimal.

□

Note that although we have $f^{\text{nom}}(\pi^{\text{SR}}) \geq f^{\text{nom}}(\pi^{\text{RR},\alpha,\beta})$, in general we do not have $f^{\text{real}}(\pi^{\text{SR}}) \geq f^{\text{real}}(\pi^{\text{RR},\alpha,\beta})$, as we will see in Example 8.23. However, for the special case $\alpha = \beta = 0$ this is indeed the case.

Corollary 8.19. *It holds* $f^{\text{real}}(\pi^{\text{SR}}) \geq f^{\text{real}}(\pi^{\text{RR},0,0})$.

Proof. Since $f^{\text{del}}(\pi^{\text{RR},0,0}) = 0$ and $f^{\text{del}}(\pi^{\text{SR}}) = 0$, from Lemma 8.18 we get

$$f^{\text{real}}(\pi^{\text{SR}}) = f^{\text{nom}}(\pi^{\text{SR}}) \geq f^{\text{nom}}(\pi^{\text{RR},0,0}) = f^{\text{nom}}(\pi^{\text{RR},0,0}) + f^{\text{del}}(\pi^{\text{RR},0,0}) = f^{\text{real}}(\pi^{\text{RR},0,0}).$$

□

One might think that actually $f^{\text{real}}(\pi^{\text{SR}}) = f^{\text{real}}(\pi^{\text{RR},0,0})$, because in both models no passenger is allowed to have any delay. However, we remark that the recoverable robust model still has more flexibility than the strictly robust one, because there are events with passenger weight zero, i.e. $w_i = 0$ for some $i \in \mathcal{E}$ (e.g. all departure events). Hence, not every source delay has to be absorbed immediately, as demonstrated in the following example.

Example 8.20. We have a look at Figure 8.1. In a strictly robust timetable, non of the events is allowed to have any delay in either one of the scenarios. Hence, we need to put one minute of buffer time on activity (i, j) (to ensure that j is on time in the red scenario) and one minute of buffer time on activity (j, i') (to ensure that i' is on time in the orange scenario). In a recoverable robust timetable it is sufficient to put the buffer only on (j, i').

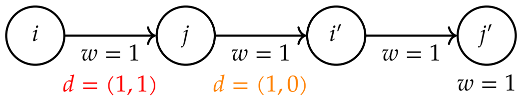


Figure 8.1: The inequality in Corollary 8.19 can be strict.

Then the event j will have one minute delay in the red scenario, but since $w_j = 0$, this does not have any effect on the weighted event delay. Hence, we need to put less buffer times in the timetable and get a smaller nominal travel time.

Lemma 8.21. *For every $\rho \geq 0$ we have $f^{\text{real}}(\pi^{\text{LR},\rho}) \geq f^{\text{real}}(\pi^{\text{RR}})$.*

Proof. Let $(\pi^{\text{LR},\rho}, z, \gamma)$ be an optimal solution to LRPT. Then $(\pi^{\text{LR},\rho}, z)$ is a feasible solution to PESP, which can be rolled out to a solution (π, F, u) to PTTA such that for the objective value we have $f^{\text{nom}}(\pi) = K \cdot f^{\text{PESP}}(\pi^{\text{LR},\rho}) \leq K \cdot (f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho)$. For this timetable π an optimal disposition timetable (x^r, y^r) can be found for every $r \in \mathcal{U}$ by solving DM, which gives the real travel time $f^{\text{real}}(\pi^{\text{LR},\rho})$. Then (π, F, u, x, y) yields a feasible solution to RRPT (by appropriately setting H, Z_1, Z_2). Since RRPT minimises the real travel time we get $f^{\text{real}}(\pi^{\text{LR},\rho}) \geq f^{\text{real}}(\pi^{\text{RR}})$. □

Remark 8.22. Even if $\bar{f} = K(f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho)$, i.e. the upper bounds for the nominal travel time in RRPT($\bar{f}$) and LRPT are equal, the recoverable robust model can still be better than the lightly robust model. This is due to the fact that the recoverable robust model uses the real delay in the objective function, while $\|\gamma\|$ only takes into account the source delays, but not the propagated delays.

After having proved some relations for the real travel time, we will now see in several examples that for some model pairs we do not have a general order.

Example 8.23. We now show that π^{SR} and $\pi^{\text{AR},\underline{\mathcal{E}}^2}$ may have a better real travel time than $\pi^{\text{PESP}}, \pi^{\text{LR},\rho}, \pi^{\text{RR},\alpha,\beta}$ and $\pi^{\text{RR},\bar{f}}$. To see this, we consider the EAN in Figure 8.2 with $K = 2$. We have one delay scenario with a source delay of 5 minutes in the first period and 4 minutes in the second period on the first activity, which results in

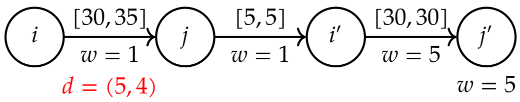
$$f^{\text{real}}(\pi^{\text{SR}}) = 2 \cdot (35 + 5 + 5 \cdot 30) = 2 \cdot 190 = 380.$$

For ARPT with $\underline{\mathcal{E}}^2 = \{j, j'\}$ we obtain the same timetable for both the nominal and the delayed scenario, and we also have $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) = 380$. We want to find a lightly robust timetable with $\rho = 4$. Note that $f^{\text{PESP}}(\pi^{\text{PESP}}) = 30 + 5 + 5 \cdot 30 = 185$ with a delay of $f^{\text{del}}(\pi^{\text{PESP}}) = 5 \cdot 5 + 5 \cdot 4 = 45$, yielding

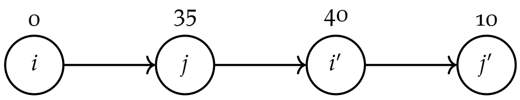
$$f^{\text{real}}(\pi^{\text{PESP}}) = 2 \cdot 185 + 45 = 415 > f^{\text{real}}(\pi^{\text{SR}}) = f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}).$$

Since we can put at most 4 minutes buffer time, the delay in the first period cannot be absorbed completely, which means event j', which has a passenger weight of 5, will have at least one minute delay. We obtain

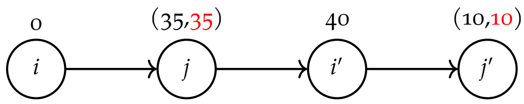
$$f^{\text{real}}(\pi^{\text{LR},4}) = 2 \cdot (185 + 4) + 5 \cdot 1 + 0 = 383 > 380 = f^{\text{real}}(\pi^{\text{SR}}) = f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}).$$



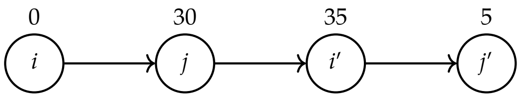
(a) Instance, $K = 2$, $T = 60$.



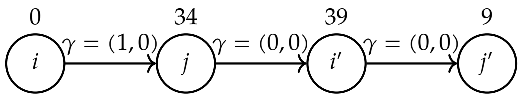
(b) Optimal solution to SRPT, $f^{\text{real}}(\pi^{\text{SR}}) = 380$.



(c) Optimal solution to ARPT with $\underline{\mathcal{E}}^2 = \{j, j'\}$, $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) = 380$.



(d) Optimal solution to PESP and RRPT(45, 0), $f^{\text{real}}(\pi^{\text{PESP}}) = f^{\text{real}}(\pi^{\text{RR},45,0}) = 415$.



(e) Optimal solution to LRPT(4) and RRPT(378), $f^{\text{real}}(\pi^{\text{LR},4}) = f^{\text{real}}(\pi^{\text{RR},378}) = 383$.

Figure 8.2: SRPT and ARPT can yield good results.

Note that the same is true for RRPT($\bar{f}$) with $\bar{f} = 378$. If we consider RRPT(α, β) with a large value for α (here, $\alpha = 45$ is sufficient), then the constraint (5.30) does not impose any restriction and thus becomes redundant. In this case the optimal solution will be the PESP solution, which has no buffer times at all. The delay cannot be absorbed by the timetable, which means that all passengers will have a delay of 5 respectively 4 minutes at event j'_1 respectively j'_2, which results in a real travel time of

$$f^{\text{real}}(\pi^{\text{RR},\alpha,\beta}) = 2 \cdot 185 + 45 + 0 = 415 > 380 = f^{\text{real}}(\pi^{\text{SR}}) = f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}).$$

To summarise, we can see that although strictly robust timetables might be overly conservative, there are indeed examples where they have a better real travel time than some of the other models. An explanation is that by putting buffer on activities with a small passenger weight, it might be possible to achieve a delay reduction which is larger than the increased travel time caused by the buffer. Since LRPT and RRPT($\bar{f}$) bound the nominal travel time, they might be too restrictive. If α and β are chosen very large, RRPT(α, β) neglects the delay and thus also focusses too much on the nominal travel time.

Example 8.24. In this example we will see that for a lot of the models there is no general order for the real travel time. We consider Figure 8.3 on pages 102 to 104. The EAN and two delay scenarios, given in red and orange, are shown in Figure 8.3a. We have $f^{\text{nom}}(\pi^{\text{PESP}}) = 3 \cdot (30 + 1 + 2 \cdot 40) = 333$ and $f^{\text{del}}(\pi^{\text{PESP}}) = 34$, i.e.

$$f^{\text{real}}(\pi^{\text{PESP}}) = 333 + 34 = 367.$$

A strictly robust timetable is given in Figure 8.3c and has a travel time of

$$f^{\text{real}}(\pi^{\text{SR}}) = 3 \cdot (35 + 1 + 2 \cdot 47) = 390.$$

An optimal adjustable robust timetable with $\underline{\mathcal{E}}^1 = \{i, i'\}$, $\underline{\mathcal{E}}^2 = \{j, j'\}$ is shown in Figure 8.3d and has a travel time of $3 \cdot (31 + 5 + 2 \cdot 47) = 390$ in the worst case (which is the orange scenario). We compute two different lightly robust timetables with $\rho = 10$ in Figure 8.3e and with $\rho = 1$ in Figure 8.3f. To compare these timetables, we compute their real travel times by doing delay management. We obtain

$$f^{\text{real}}(\pi^{\text{LR},10}) = 3 \cdot (34 + 1 + 2 \cdot 43) + 12 = 375 < 390 = f^{\text{real}}(\pi^{\text{SR}}) = f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}).$$

We consider three different versions of recoverable robust timetables, namely RRPT, RRPT(α, β) and RRPT$(\bar{f})$ in Figures 8.3g to 8.3k with different parameter choices. Comparing lightly and recoverable robust timetables, Figures 8.3f and 8.3i show that

$$f^{\text{real}}(\pi^{\text{LR},1}) = 3 \cdot (31 + 1 + 2 \cdot 40) + 28 + 0 = 364 < 378 = 378 + 0 + 0 = f^{\text{real}}(\pi^{\text{RR},0,0}),$$

while

$$f^{\text{real}}(\pi^{\text{LR},10}) = 375 > 366 = 354 + 12 + 0 = f^{\text{real}}(\pi^{\text{RR},12,0})$$

by Figures 8.3e and 8.3h. For the version with restricted nominal time we get

$$f^{\text{real}}(\pi^{\text{LR},1}) = 364 < 368 = 360 + 8 + 0 = f^{\text{real}}(\pi^{\text{RR},360}) < 375 = f^{\text{real}}(\pi^{\text{LR},10})$$

by looking at Figures 8.3e, 8.3f and 8.3k. To compare the different versions of recoverable robustness, consider

$$f^{\text{real}}(\pi^{\text{RR},0,0}) = 378 > 368 = f^{\text{real}}(\pi^{\text{RR},360}) > 366 = f^{\text{real}}(\pi^{\text{RR},12,0})$$

in Figures 8.3h, 8.3i and 8.3k. For PESP we see

$$f^{\text{real}}(\pi^{\text{RR},12,0}) = 366 < 367 = f^{\text{real}}(\pi^{\text{PESP}}) < 378 = f^{\text{real}}(\pi^{\text{RR},0,0}),$$

$$f^{\text{real}}(\pi^{\text{RR},350}) = 366 < 367 = f^{\text{real}}(\pi^{\text{PESP}}) < 368 = f^{\text{real}}(\pi^{\text{RR},360}),$$

$$f^{\text{real}}(\pi^{\text{LR},1}) = 364 < 367 = f^{\text{real}}(\pi^{\text{PESP}}) < 375 = f^{\text{real}}(\pi^{\text{LR},10}).$$

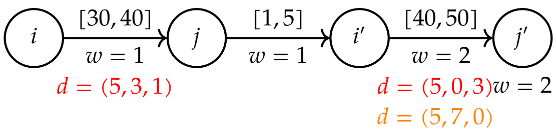
Example 8.25. We show that $\pi^{\text{AR},\underline{\mathcal{E}}^2}$ may have a better real travel time than π^{RR}. To see this, we consider the EAN in Figure 8.4a with two delay scenarios for $K = 2$. An optimal PESP solution is given in Figure 8.4b, which yields

$$f^{\text{real}}(\pi^{\text{PESP}}) = 19690 + 3000 = 22690.$$

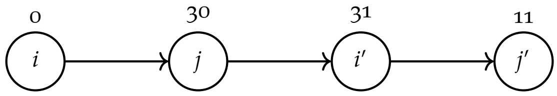
We compute an optimal adjustable timetable with $\underline{\mathcal{E}}^2 = \underline{\mathcal{E}}$ and a recoverable robust timetable in Figures 8.4c and 8.4d, respectively. We obtain

$$f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}}) = 2 \cdot 10835 = 21670 < 22180 = f^{\text{real}}(\pi^{\text{RR}}),$$

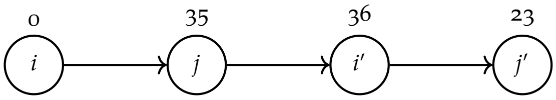
which shows that an adjustable robust timetable can be better than a recoverable robust timetable. The intuition behind this is that, since $\underline{\mathcal{E}}^2 = \underline{\mathcal{E}}$, ARPT finds a timetable for every delay scenario separately, and thus has more flexibility than RRPT, where all disposition timetables depend on the nominal timetable (i.e. it is not allowed that an event happens earlier than planned).



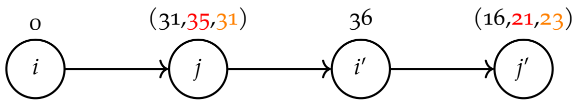
(a) Instance, $I = [8{:}00, 10{:}59]$, $T = 60$, $(K = 3)$.



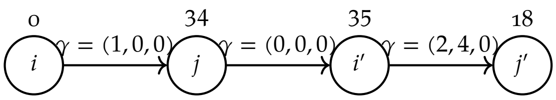
(b) Optimal solution to PESP, $f^{\text{nom}}(\pi^{\text{PESP}}) = 333, f^{\text{del}}(\pi^{\text{PESP}}) = 34$.



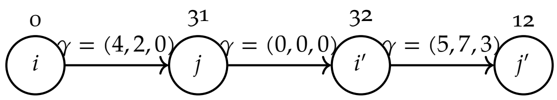
(c) Optimal solution to SRPT, $f^{\text{real}}(\pi^{\text{SR}}) = 390$.



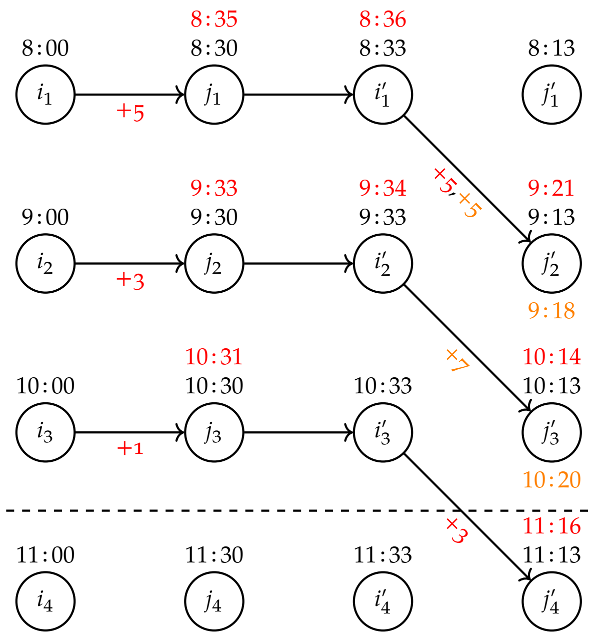
(d) Optimal solution to ARPT with $\underline{\mathcal{E}}^1 = \{i, i'\}, \underline{\mathcal{E}}^2 = \{j, j'\}, f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) = 390$.



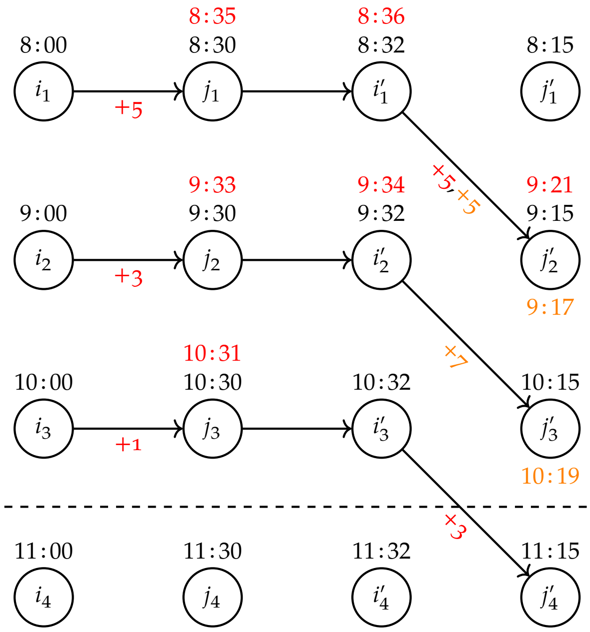
(e) Lightly robust timetable with $\rho = 10$. $f^{\text{nom}}(\pi^{\text{LR},10}) = 363, f^{\text{del}}(\pi^{\text{LR},10}) = 12$.



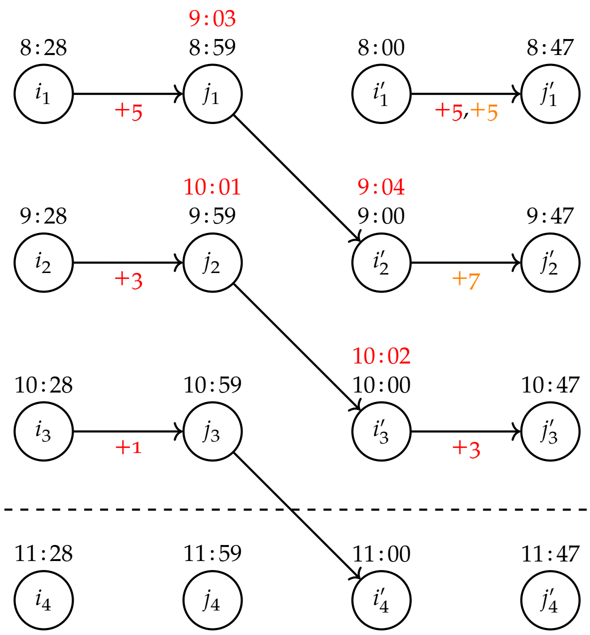
(f) Lightly robust timetable with $\rho = 1$. $f^{\text{nom}}(\pi^{\text{LR},1}) = 336, f^{\text{del}}(\pi^{\text{LR},1}) = 28$.



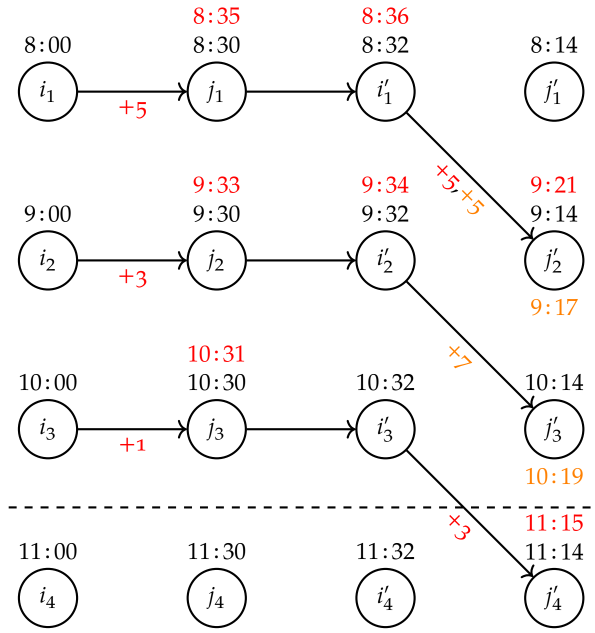
(g) Optimal solution to RRPT, $f^{\text{nom}}(\pi^{\text{RR}}) = 339, f^{\text{del}}(\pi^{\text{RR}}) = 24$.



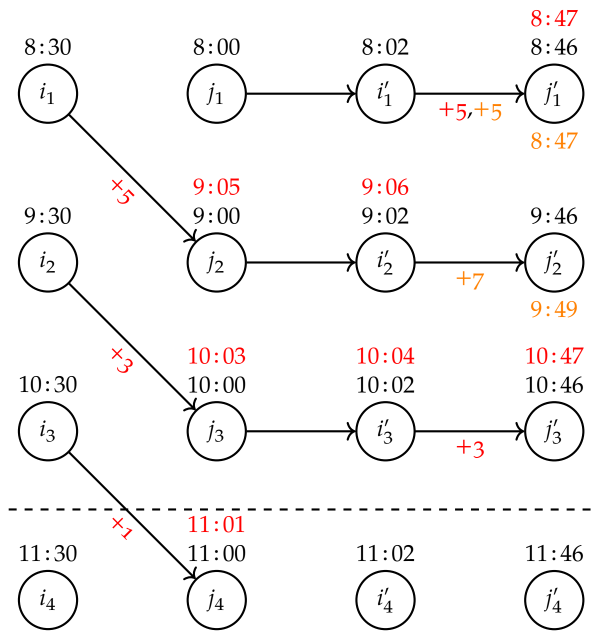
(h) Optimal solution to RRPT(12, 0), $f^{\text{nom}}(\pi^{\text{RR},12,0}) = 354$, $f^{\text{del}}(\pi^{\text{RR},12,0}) = 12$.



(i) Optimal solution to RRPT(0, 0), $f^{\text{nom}}(\pi^{\text{RR},0,0}) = 378$, $f^{\text{del}}(\pi^{\text{RR},0,0}) = 0$.



(j) Optimal solution to RRPT(350), $f^{\mathrm{nom}}(\pi^{\mathrm{RR},350}) = 350, f^{\mathrm{del}}(\pi^{\mathrm{RR},350}) = 16$.



(k) Optimal solution to RRPT(360), $f^{\mathrm{nom}}(\pi^{\mathrm{RR},360}) = 360, f^{\mathrm{del}}(\pi^{\mathrm{RR},360}) = 8$.

Figure 8.3: Comparison of timetables using different robustness concepts and parameters.

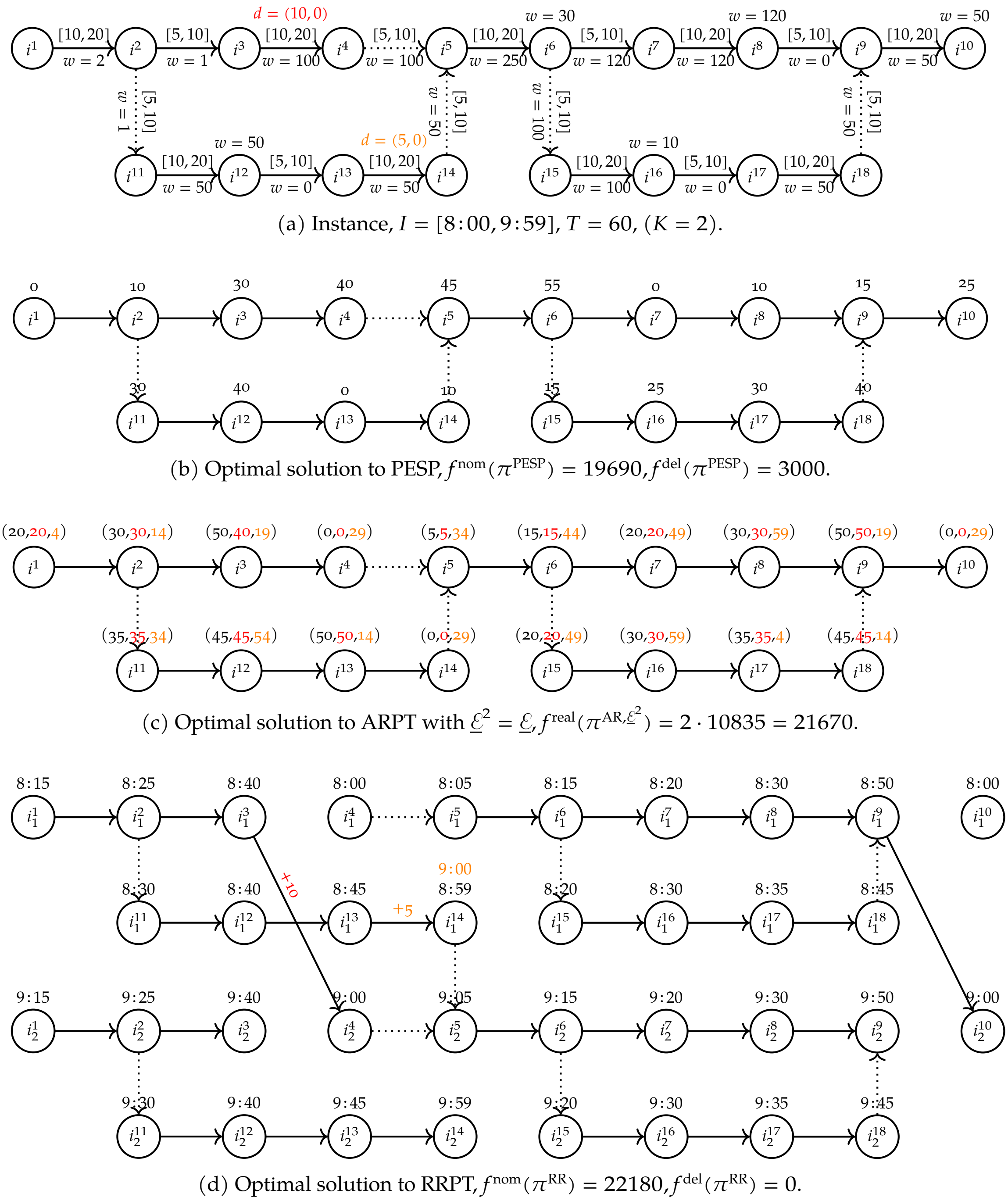
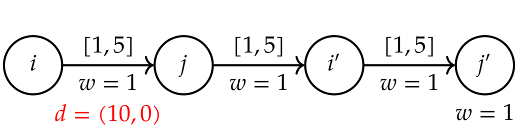
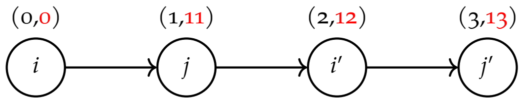


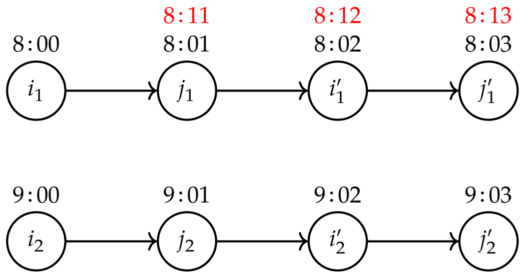
Figure 8.4: An adjustable robust timetable can be better than a recoverable robust timetable.



(a) Instance, $I = [8{:}00, 9{:}59]$, $T = 60$, $(K = 2)$.



(b) Optimal solution to ARPT, $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) = 2 \cdot 13 = 26$.



(c) Optimal solution to RRPT, $f^{\text{nom}}(\pi^{\text{RR}}) = 6$, $f^{\text{del}}(\pi^{\text{RR}}) = 10$.

Figure 8.5: Even with $\underline{\mathcal{E}}^2 = \underline{\mathcal{E}}$ ARPT can be worse than RRPT.

Example 8.26. Even in the case that $\underline{\mathcal{E}}^1 = \emptyset, \underline{\mathcal{E}}^2 = \underline{\mathcal{E}}$, there are instances where RRPT is still better than ARPT. Although the adjustable robust model has the advantage of not having to respect some nominal timetable, it still has the disadvantage that the timetable has to be periodic in every scenario, while for the recoverable robust model only the nominal timetable has to be periodic. An example can be found in Figure 8.5. Here, we have

$$f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}}) = 2 \cdot 13 = 26 > 16 = 6 + 10 = f^{\text{real}}(\pi^{\text{RR}}).$$

We have seen that for some of the introduced models we have a general order between the real travel times. In particular, the model RRPT is superior to most of the other considered models. The relations between the travel times are summarised in Figure 8.6. However, for many model pairs there is no general order. If we have two optimisation problems (P) and (P') with optimal solutions π and π', we write $f^{\text{real}}(\pi) \lessgtr f^{\text{real}}(\pi')$, if there is some instance with $f^{\text{real}}(\pi) < f^{\text{real}}(\pi')$ and some instance with $f^{\text{real}}(\pi) >$

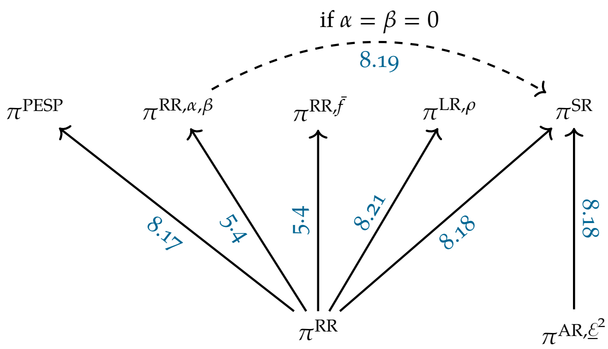


Figure 8.6: There is an arc from π to π' if $f^{\text{real}}(\pi) \leq f^{\text{real}}(\pi')$. If there is no arc, there is no general order between the real travel times. The dashed arc only holds in a special case.

$f^{\text{real}}(\pi')$. Specifically, from the previous examples we obtain:

- $f^{\text{real}}(\pi^{\text{SR}}) \lesseqgtr f^{\text{real}}(\pi^{\text{LR},\rho})$ (8.23, 8.24)
- $f^{\text{real}}(\pi^{\text{SR}}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR},\bar{f}})$ (8.23, 8.24)
- $f^{\text{real}}(\pi^{\text{SR}}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR},\alpha,\beta})$ (8.23, 8.24)
- $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) \lesseqgtr f^{\text{real}}(\pi^{\text{LR},\rho})$ (8.23, 8.24)
- $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR}})$ (8.24, 8.25)
- $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR},\bar{f}})$ (8.24, 8.25)
- $f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR},\alpha,\beta})$ (8.24, 8.25)
- $f^{\text{real}}(\pi^{\text{RR},\alpha,\beta}) \lesseqgtr f^{\text{real}}(\pi^{\text{LR},\rho})$ (8.24)
- $f^{\text{real}}(\pi^{\text{RR},\bar{f}}) \lesseqgtr f^{\text{real}}(\pi^{\text{LR},\rho})$ (8.24)
- $f^{\text{real}}(\pi^{\text{RR},\bar{f}}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR},\alpha,\beta})$ (8.24)
- $f^{\text{real}}(\pi^{\text{PESP}}) \lesseqgtr f^{\text{real}}(\pi^{\text{LR},\rho})$ (8.24)
- $f^{\text{real}}(\pi^{\text{PESP}}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR},\alpha,\beta})$ (8.24)
- $f^{\text{real}}(\pi^{\text{PESP}}) \lesseqgtr f^{\text{real}}(\pi^{\text{RR},\bar{f}})$ (8.24)
- $f^{\text{real}}(\pi^{\text{PESP}}) \lesseqgtr f^{\text{real}}(\pi^{\text{AR},\underline{\mathcal{E}}^2})$ (8.24, 8.25)
- $f^{\text{real}}(\pi^{\text{PESP}}) \lesseqgtr f^{\text{real}}(\pi^{\text{SR}})$ (8.23, 8.24).

8.5 COMPUTATIONAL EXPERIMENTS

After having compared the models theoretically, we now compare them in a computational study. For the experiments we used the dataset `toy` which we have already seen in Chapter 5, which we rolled out over 4 hours. The period length of the timetable is 60 minutes. We considered several delay scenario sets $\mathcal{U}$, differing in the number (given as the percentage of aperiodic activities which have a source delay) and size (in minutes) of the source delays. The different settings are summarised in Table 8.2.

We implemented the MIP formulations in Python and solved them using Gurobi 8.1.1 [Gur23] on a compute server with 48 cores @2.9 GHz and 196 GB RAM. Note that for RRPT we used the formulation RRPT-pe.

Experiment	Activities with source delay	Size of source delays
1	1%	$[1, U_a - L_a]$
2	5%	[1,15]

Table 8.2: Experiment settings.

EXPERIMENT 1 For the first experiment, we generated random source delays for 1% of the (rolled out) activities. Recall that by Lemma 8.3 and Lemma 8.8, SRPT and ARPT are infeasible for large source delays. Hence, for $a \in \mathcal{A}$, $r \in \mathcal{U}$ we chose d_a^r from the interval $[1, U_a - L_a]$.

The results are shown in Figure 8.7. As expected, RRPT has the smallest real travel time. However, the difference to PESP and LRPT is not too big. Also the restricted versions RRPT(α, β) and RRPT$(\bar{f})$ find solutions very close to that of RRPT. For these five models the real travel time hardly depends on the size of $\mathcal{U}$. As expected, SRPT has by far the highest real travel time and it increases with the number of scenarios. For ARPT we chose the partition $\underline{\mathcal{E}}^1 = \underline{\mathcal{E}}^{\text{dep}}, \underline{\mathcal{E}}^2 = \underline{\mathcal{E}}^{\text{arr}}$ as motivated in Remark 8.6. Here, the real travel time is much smaller than for SRPT, but still far higher than for the other models.

For LRPT, RRPT(α, β) and RRPT$(\bar{f})$ we also looked at the real travel time for different parameter choices. Since both RRPT$(\bar{f})$ and LRPT have a bound for f^{nom}, we chose the parameters such that these bounds are equal, i.e. $\bar{f} = K(f^{\text{PESP}}(\pi^{\text{PESP}}) + \rho)$. Additionally, for LRPT we also used two different norms, namely $\|\cdot\|_1$ ("sum") and $\|\cdot\|_w$ ("weighted") from Remark 8.4.

The results are shown in Figure 8.8. For the sum norm we obtain slightly better results than for the weighted norm, so taking the weights into account does not seem to be beneficial. For RRPT(α, β) we chose $\alpha_2 < Z_1(\pi^{\text{RR}}, x) < \alpha_1 < Z_1(\pi^{\text{PESP}}, x^{\text{PESP}})$ and $\beta_2 < Z_2(\pi^{\text{RR}}, y) < \beta_1 < Z_2(\pi^{\text{PESP}}, y^{\text{PESP}})$, where (π^{RR}, x, y) is an optimal RRPT solution and $(x^{\text{PESP}}, y^{\text{PESP}})$ is an optimal DM-strategy for π^{PESP}. The combinations (α_1, β_1), (α_2, β_2), (α_1, β_2), (α_2, β_1) all yield rather good results (note that for two of them the objective values coincide), but the ones with a smaller value for α perform a bit better, meaning that stronger restricting the total delay is beneficial.

Although RRPT yields a better real travel time than PESP, the difference is not as big as one might have expected. Hence, we look at the real travel time in more detail by evaluating f^{nom} and f^{del} separately. We expect that π^{RR} has a higher nominal travel time since buffer times are added to the timetable, but less delay, since said buffers can be used to absorb some of the delay. As can be seen in Table 8.3, the increase in the nominal travel time is only very small: it is at most 0.56%. On the other hand, the decrease in the delay is significant: we can reduce the delay by up to 27%. This means that by increasing f^{nom} only slightly, RRPT is able to decrease f^{del} considerably. However, since f^{nom} is much larger than f^{del}, when we consider the sum $f^{\text{real}} = f^{\text{nom}} + f^{\text{del}}$, this advantage becomes less significant: in total we have an improvement between 1.12% and 3.57%.

EXPERIMENT 2 If the contribution of f^{del} to f^{real} becomes bigger, we expect that the advantage of RRPT becomes more prominent. Hence, we repeated the experiment with more and larger source delays: we chose source delays between 1 and 15 minutes for 5% of the (rolled out) activities. Since the dataset contains a lot of activities $\underline{a}$ for which

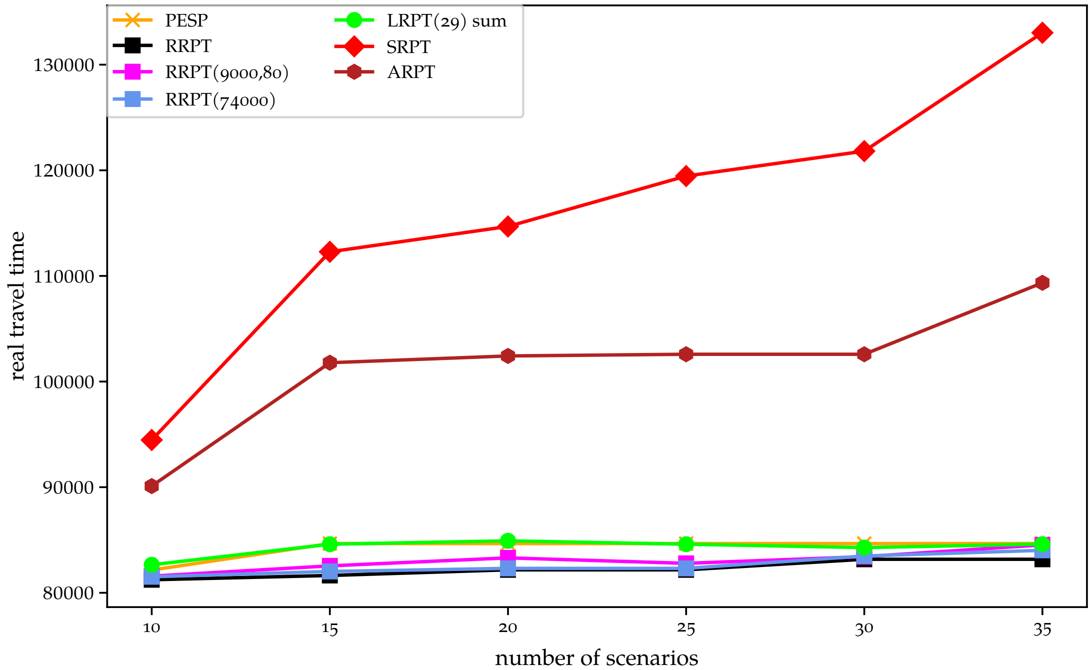


Figure 8.7: The real travel time of all models in Experiment 1.

Scenarios	10	15	20	25	30	35
$f^{\text{nom}}(\pi^{\text{PESP}})$	73184	73184	73184	73184	73184	73184
$f^{\text{nom}}(\pi^{\text{RR}})$	73304	73264	73592	73592	73224	73224
increase (%)	0.16	0.11	0.56	0.56	0.05	0.05
$f^{\text{del}}(\pi^{\text{PESP}})$	8970	11477	11477	11477	11477	11477
$f^{\text{del}}(\pi^{\text{RR}})$	7930	8376	8582	8582	9950	9950
increase (%)	-11.59	-27.02	-25.22	-25.22	-13.3	-13.3
$f^{\text{real}}(\pi^{\text{PESP}})$	82154	84661	84661	84661	84661	84661
$f^{\text{real}}(\pi^{\text{RR}})$	81234	81640	82174	82174	83174	83174
increase (%)	-1.12	-3.57	-2.94	-2.94	-1.76	-1.76

Table 8.3: f^{nom}, f^{del} and f^{real} for RRPT compared to PESP in Experiment 1.

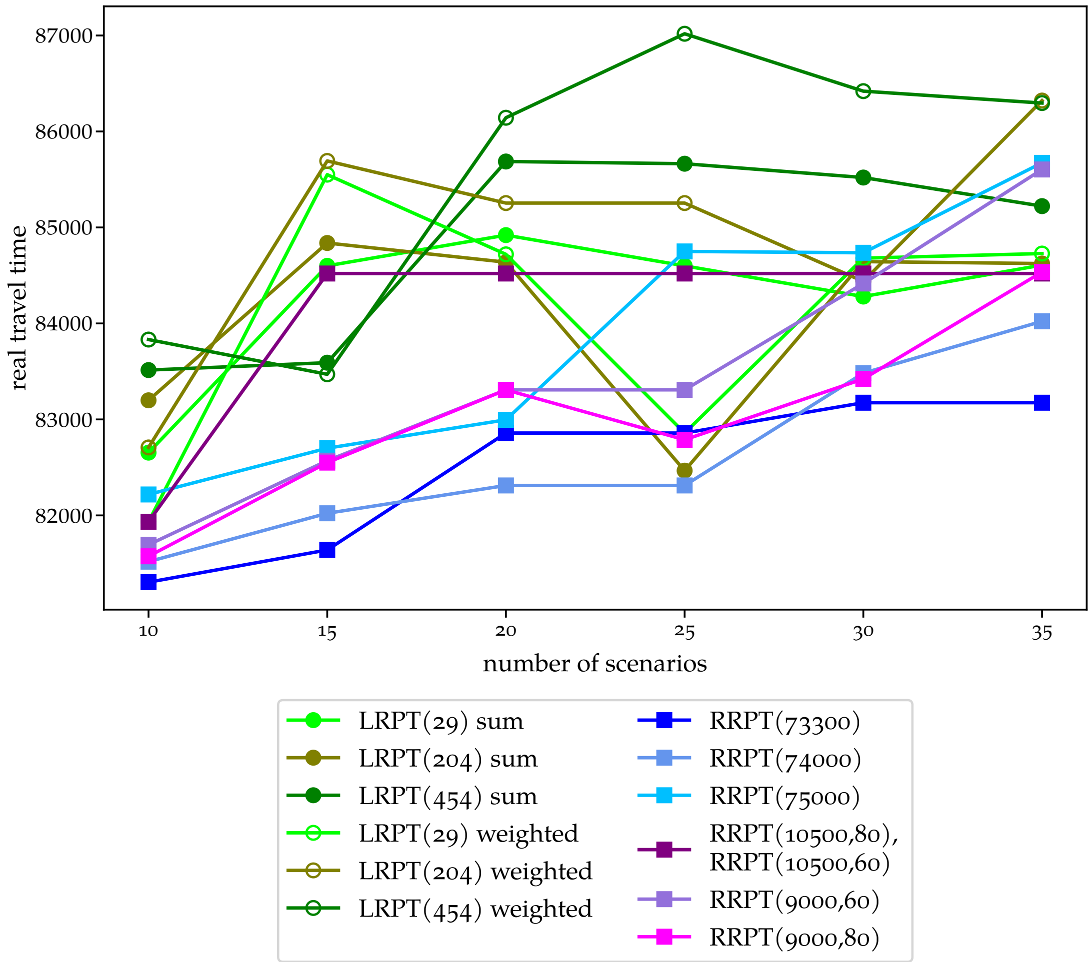


Figure 8.8: The real travel time of RRPT(α, β), RRPT$(\bar{f})$ and LRPT for different parameters in Experiment 1.

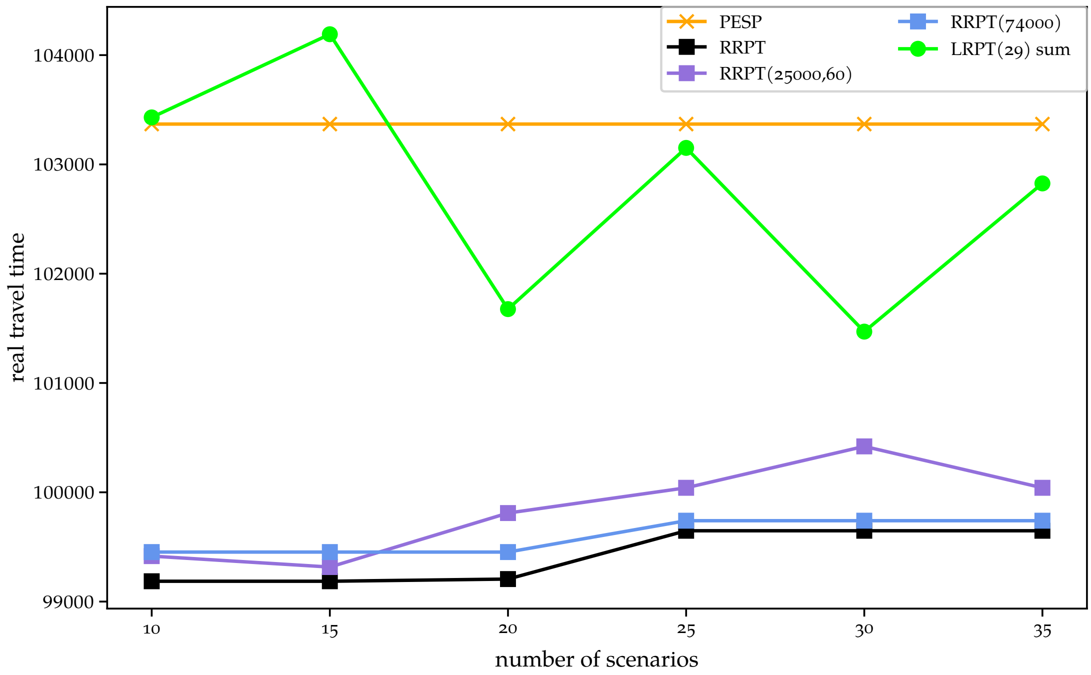


Figure 8.9: The real travel time of all models in Experiment 2.

$U_{\underline{a}} - L_{\underline{a}}$ is rather small (often only 1 or 2 minutes), SRPT and ARPT are infeasible for this choice of $\mathcal{U}$. Hence, in this experiment we can only compare the values of the other models.

The results are presented in Figure 8.9 and Table 8.4. Here, the real travel time for RRPT is 3.6% to 4.05% smaller than for PESP. We can conclude that the potential of RRPT to reduce the real travel time increases with the delay, so it is especially suited in the presence of many and large source delays. Again, RRPT(α, β) and RRPT$(\bar{f})$ are nearly as good as RRPT, while LRPT performs similar to PESP.

Looking at the impact of the parameter choice in Figure 8.10, there is no significant difference regarding the choice of the norm.

For RRPT(α, β) we again use four different parameter combinations. The one where both $\alpha > Z_1(\pi^{RR}, x)$ and $\beta > Z_2(\pi^{RR}, y)$ (for an optimal solution (π^{RR}, x, y) to RRPT) is a bit worse than the others.

Computing times Finally, we investigate the computing times for solving the different problems. The results can be found in Tables 8.5 and 8.6. While PESP, SRPT and LRPT can be solved within seconds, the computing times increase tremendously for ARPT and RRPT. This is a huge disadvantage of the recoverable robust approach. However, since timetabling is part of the tactical planning phase, longer computing times might be acceptable. Nevertheless, for large instances there is so far little hope of solving RRPT to optimality. Heuristic approaches, as already considered in Chapter 7, could prove vital to implement recoverable robust timetables in practice. Interestingly, RRPT(α, β) and RRPT$(\bar{f})$ have even worse computing times than RRPT: for several instances the optimal solution was not found within the time limit of one hour. Since these two models cannot achieve a better real travel time than RRPT, they are not very promising.

Scenarios	10	15	20	25	30	35
$f^{\text{nom}}(\pi^{\text{PESP}})$	73184	73184	73184	73184	73184	73184
$f^{\text{nom}}(\pi^{\text{RR}})$	74196	74196	74396	74276	74276	74276
increase (%)	1.38	1.38	1.66	1.49	1.49	1.49
$f^{\text{del}}(\pi^{\text{PESP}})$	30185	30185	30185	30185	30185	30185
$f^{\text{del}}(\pi^{\text{RR}})$	24990	24990	24810	25372	25372	25372
increase (%)	-17.21	-17.21	-17.81	-15.95	-15.95	-15.95
$f^{\text{real}}(\pi^{\text{PESP}})$	103369	103369	103369	103369	103369	103369
$f^{\text{real}}(\pi^{\text{RR}})$	99186	99186	99206	99648	99648	99648
increase (%)	-4.05	-4.05	-4.03	-3.6	-3.6	-3.6

Table 8.4: f^{nom}, f^{del} and f^{real} for RRPT compared to PESP in Experiment 2.

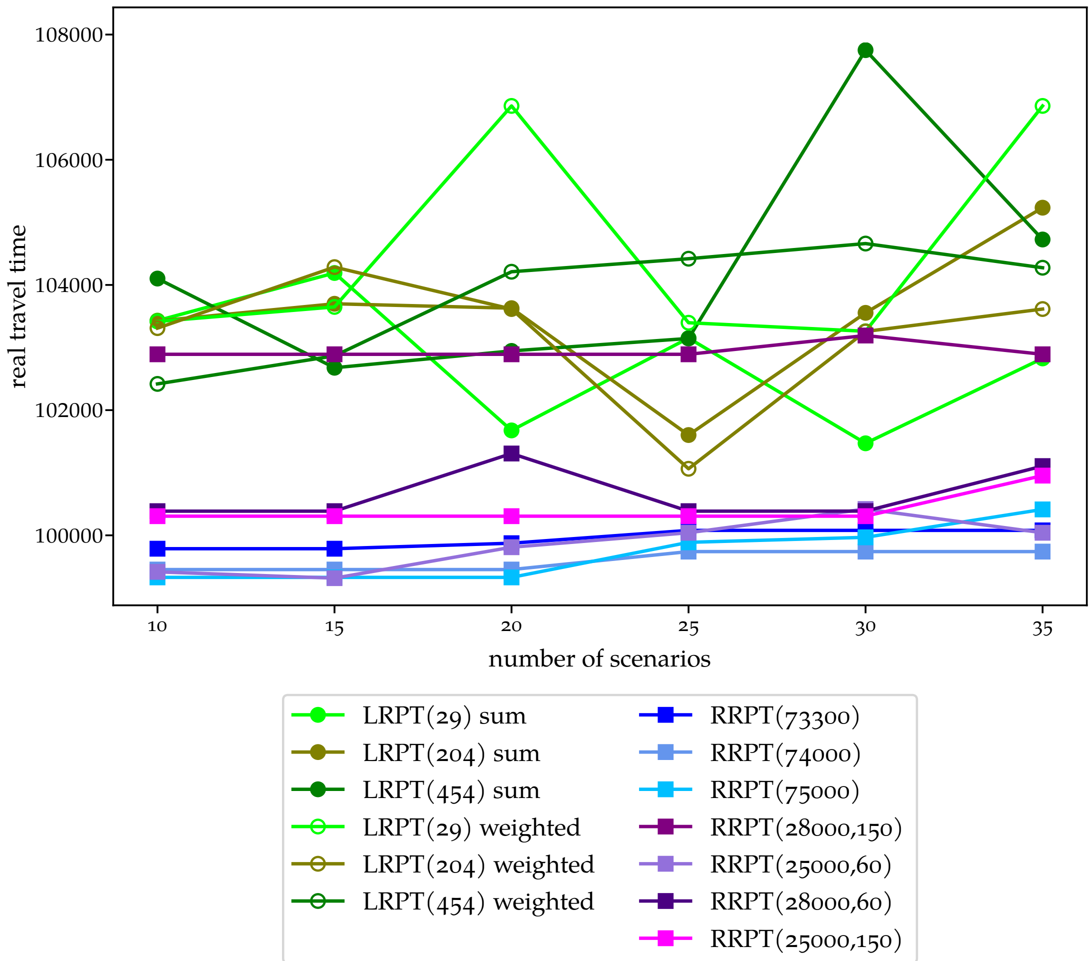


Figure 8.10: The real travel time of RRPT(α, β), RRPT($\bar{f}$) and LRPT for different parameters in Experiment 2.

Scenarios	10	15	20	25	30	35
PESP	0.44	0.49	0.61	0.77	1.02	1.09
RRPT	29.33	141.52	279.28	353.51	665.43	543.06
LRPT(29)	0.94	1.43	1.44	1.9	2.09	2.42
SRPT	0.35	0.48	0.79	0.47	0.72	0.57
ARPT	11.57	59.87	37.5	207.32	190.87	277.04
RRPT(10500, 80)	420.75	204.33	495.18	376.31	492.81	347.82
RRPT(9000, 60)	358.46	103.61	176.89	limit	limit	limit
RRPT(10500, 60)	220.34	185.58	172.92	634.73	1010.97	715.01
RRPT(9000, 80)	1052.87	103.71	327.98	1620.61	2671.88	limit
RRPT(73300)	70.06	579.07	469.95	1199.43	1320.62	1547.01
RRPT(74000)	445.29	1330.52	1045.85	3119.87	limit	limit
RRPT(75000)	573.56	1029.34	1577.27	limit	limit	limit

Table 8.5: Computing times (seconds) in Experiment 1. The time limit was one hour.

Scenarios	10	15	20	25	30	35
PESP	0.68	0.49	0.69	0.88	1.22	1.13
RRPT	166.04	244.66	313.39	548.98	1065.72	2548.53
LRPT(29)	1.14	1.47	1.64	2.09	2.26	2.43
RRPT(28000, 150)	221.24	350.51	382.1	1555.72	812.28	1549.26
RRPT(25000, 60)	904.01	1137.82	1742.77	limit	2227.43	limit
RRPT(28000, 60)	320.39	253.99	840.53	972.05	1210.11	2076.08
RRPT(25000, 150)	658.83	678.57	2271.11	3355.26	2126.36	limit
RRPT(73300)	221.56	458.01	limit	1063.66	1268.75	1795.5
RRPT(74000)	279.31	812.98	609.74	1187.48	1931.34	2931.33
RRPT(75000)	296.14	653.24	987.42	2182.01	3121.89	limit

Table 8.6: Computing times (seconds) in Experiment 2. The time limit was one hour.

CONCLUSION

We have applied strict robustness, light robustness and adjustable robustness to the Periodic Event Scheduling Problem. Putting a focus on the aperiodicity of the source delays, we have seen that some of the derived models, namely the ones using strict or adjustable robustness, are not suited for taking the aperiodicity of the delays into account. We have compared these different models as well as the recoverable robust model from Chapter 5 with respect to the real travel time of the passengers, showing that recoverable robustness is particularly well suited for finding good timetables, especially in networks with a lot of delay. However, in the computational study a big disadvantage of the recoverable robust model has become clear, namely its large computing times.

9 CONCLUSION AND OUTLOOK

In this thesis, we studied robust periodic timetabling in public transport. Recoverable robust timetabling was the main focus of our work since this concept fits the application in a natural way. This becomes apparent in the fact that applying it to timetabling results in the integration of two problems which are important steps in the classical public transport planning process, namely timetabling and delay management. However, some obstacles had to be overcome in the process of developing the robust models. Most strikingly, there is a clash between the periodicity of the timetable and the aperiodicity of the source delays. To resolve this problem, we made a first important step by introducing the problem PTTA in Chapter 4, which translates the periodic timetabling problem to the aperiodic network.

With this preparatory work, we were able to formulate different variants of the Recoverable Robust Periodic Timetabling Problem in Chapter 5. When solving the problem, the choice of the formulation can have a large impact on the computing time, which became evident when we compared three equivalent formulations of RRPT, differing in the way the timetabling subproblem is treated. Surprisingly, it turned out that for the integrated model, the cycle base approach, which is superior for the pure timetabling problem, is not the best choice. A direction for further research could be to further strengthen the MIP formulation.

Also in other aspects there are significant differences between the robust models and PESP, which transpires in the fact that some fundamental and intuitive properties of PESP do not hold for our robust model, as shown in Chapter 6. We also showed that when using an always-wait strategy for the delay management subproblem, the uncertainty set can be reduced to its extreme points.

To be able to compute recoverable robust timetables in practice, one has to resort to heuristic approaches. We made a first step by developing several algorithms for finding timetables with increased robustness by introducing buffer times on selected activities in Chapter 7. These algorithms were indeed able to find timetables with a smaller real travel time. Here is a lot of potential for further research. While we only considered different variants of one type of heuristic, which iteratively solves the timetabling and the delay management subproblems, there are a lot of other paths one could pursue, for example using ideas from the literature on either PESP heuristics or DM heuristics. Another option could be to reduce the scenario set in a reasonable way to decrease the problem size.

Since numerous other robustness concepts exist in the literature, we also applied some of these to the periodic timetabling problem, namely strict robustness, light robustness and adjustable robustness, in Chapter 8. We developed periodic timetabling models using these concepts and saw that the strict and the adjustable robust model are not able to properly take the aperiodicity of the source delays into account. We compared the models with respect to their real travel time, showing that recoverable robust models are indeed superior. However, they come with a big disadvantage: finding an optimal recoverable robust timetable is much more computationally challenging than solving the other considered models, which emphasises the importance of finding good heuristics.

There are a lot of possible extensions to the problem, incorporating more aspects from the real world. For instance, the extensions which were already considered for PESP and DM in the literature could be also applied to RRPT, for example integrating passenger (re-)routing, track choice or vehicle circulations. Of course, this will make the already hard problem even more challenging. Instead of assuming a discrete scenario set, it might also make sense to consider an infinite scenario set.

On the other hand, one could also consider special cases in which the problem might simplify. This could include assuming a certain graph structure or using fixed delay management strategies, such as an always-wait strategy or waiting time rules, instead of doing optimal delay management.

Overall, there is still a lot of work to do in order to be able to use recoverable robust periodic timetables in practice.

A

APPENDIX

A.1 DATASETS

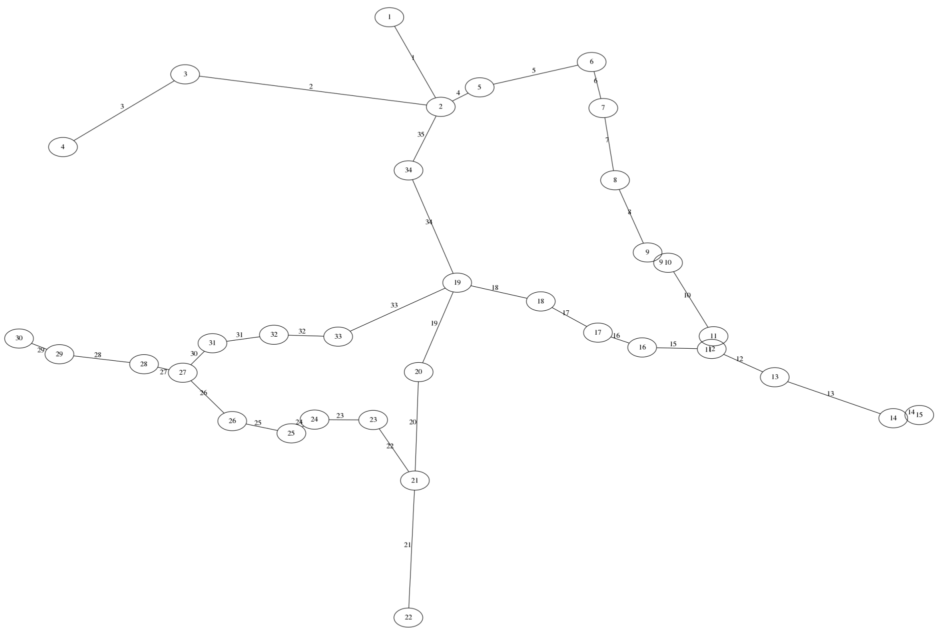


Figure A.1: The PTN of `lowersaxony`.

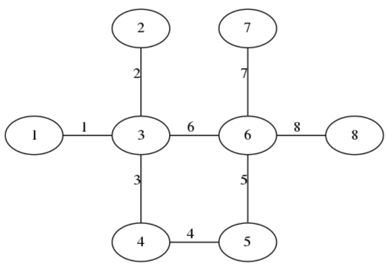


Figure A.2: The PTN of `toy`.

Name	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{A}_{\text{drive}}\rvert$	$\lvert\mathcal{A}_{\text{wait}}\rvert$	$\lvert\mathcal{A}_{\text{transfer}}\rvert$	$\lvert\mathcal{A}_{\text{sync}}\rvert$
`toy`	156	188	78	50	10	50

Table A.1: Size of the EAN of `toy`.

BIBLIOGRAPHY

[Ben+04] A. Ben-Tal, A. Goryashko, E. Guslitzer and A. Nemirovski. 'Adjustable robust solutions of uncertain linear programs'. In: *Mathematical Programming* 99.2 (2004), pp. 351–376. DOI: 10.1007/s10107-003-0454-y.

[BEN09] A. Ben-Tal, L. El Ghaoui and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[BL08] R. Borndörfer and C. Liebchen. 'When Periodic Timetables Are Suboptimal'. In: *Operations Research Proceedings 2007*. Springer Berlin Heidelberg, 2008, pp. 449–454. DOI: 10.1007/978-3-540-77903-2_69.

[BLR20] R. Borndörfer, N. Lindner and S. Roth. 'A concurrent approach to the periodic event scheduling problem'. In: *Journal of Rail Transport Planning & Management* 15 (2020). DOI: 10.1016/j.jrtpm.2019.100175.

[BN00] A. Ben-Tal and A. Nemirovski. 'Robust solutions of Linear Programming problems contaminated with uncertain data'. In: *Mathematical Programming, Series B* 88.3 (2000), pp. 411–424. DOI: 10.1007/PL00011380.

[BN98] A. Ben-Tal and A. Nemirovski. 'Robust convex optimization'. In: *Mathematics of Operations Research* 23.4 (1998), pp. 769–805. DOI: 10.1287/moor.23.4.769.

[BN99] A. Ben-Tal and A. Nemirovski. 'Robust solutions of uncertain linear programs'. In: *Operations Research Letters* 25.1 (1999), pp. 1–13. DOI: 10.1016/S0167-6377(99)00016-4.

[BS14] R. Bauer and A. Schöbel. 'Rules of thumb: Practical online-strategies for delay management'. In: *Public Transport* 6.1-2 (2014), pp. 85–105. DOI: 10.1007/s12469-013-0082-8.

[CCF12] V. Cacchiani, A. Caprara and M. Fischetti. 'A lagrangian heuristic for robustness, with an application to train timetabling'. In: *Transportation Science* 46.1 (2012), pp. 124–133. DOI: 10.1287/trsc.1110.0378.

[CGS17] E. Carrizosa, M. Goerigk and A. Schöbel. 'A biobjective approach to recoverable robustness based on location planning'. In: *European Journal of Operational Research* 261.2 (2017), pp. 421–435. DOI: 10.1016/j.ejor.2017.02.014.

[Cic+08] S. Cicerone, G. Di Stefano, M. Schachtebeck and A. Schöbel. 'Dynamic algorithms for recoverable robustness problems'. In: *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08)*. Vol. 9. Open Access Series in Informatics. 2008, pp. 1–20. DOI: 10.4230/OASIcs.ATMOS.2008.1587.

[Cic+09] S. Cicerone, G. D'Angelo, G. Di Stefano, D. Frigioni and A. Navarra. 'Recoverable robust timetabling for single delay: Complexity and polynomial algorithms for special cases'. In: *Journal of Combinatorial Optimization* 18.3 (2009), pp. 229–257. DOI: 10.1007/s10878-009-9247-4.

[CKL17] G. Caimi, L. Kroon and C. Liebchen. 'Models for railway timetable optimization: Applicability and applications in practice'. In: *Journal of Rail Transport Planning and Management* 6.4 (2017), pp. 285–312. DOI: 10.1016/j.jrtpm.2016.11.002.

[CT12] V. Cacchiani and P. Toth. 'Nominal and robust train timetabling problems'. In: *European Journal of Operational Research* 219.3 (2012), pp. 727–737. DOI: 10.1016/j.ejor.2011.11.003.

[DAn+11] G. D'Angelo, G. Di Stefano, A. Navarra and C. M. Pinotti. 'Recoverable Robust Timetables: An Algorithmic Approach on Trees'. In: *IEEE Transactions on Computers* 60.3 (2011), pp. 433–446. DOI: 10.1109/TC.2010.142.

[Dol+12] T. Dollevoet, D. Huisman, M. Schmidt and A. Schöbel. 'Delay Management with Rerouting of Passengers'. In: *Transportation Science* 46.1 (2012), pp. 74–89. DOI: 10.1287/trsc.1110.0375.

[Dol+15] T. Dollevoet, D. Huisman, L. Kroon, M. Schmidt and A. Schöbel. 'Delay Management Including Capacities of Stations'. In: *Transportation Science* 49.2 (2015), pp. 185–203. DOI: 10.1287/trsc.2013.0506.

[Dol+18] T. Dollevoet, D. Huisman, M. Schmidt and A. Schöbel. 'Delay propagation and delay management in transportation networks'. In: *Handbook of Optimization in the Railway Industry*. Springer, 2018, pp. 285–317. DOI: 10.1007/978-3-319-72153-8_13.

[EL97] L. El Ghaoui and H. Lebret. 'Robust solutions to least-squares problems with uncertain data'. In: *SIAM Journal on Matrix Analysis and Applications* 18 (1997), pp. 1035–1064. DOI: 10.1137/S0895479896298130.

[EOL98] L. El Ghaoui, F. Oustry and H. Lebret. 'Robust solutions to uncertain semidefinite programs'. In: *SIAM Journal on Optimization* 9.1 (1998), pp. 33–52. DOI: 10.1137/S1052623496305717.

[Fli+08] H. Flier, M. Nunkesser, M. Schachtebeck and A. Schöbel. *Integrating Rolling Stock Circulation into the Delay Management Problem*. Report. 2008.

[FM09] M. Fischetti and M. Monaci. 'Light Robustness'. In: *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*. Springer Berlin Heidelberg, 2009, pp. 61–84. DOI: 10.1007/978-3-642-05465-5_3.

[FSZ09] M. Fischetti, D. Salvagnin and A. Zanette. 'Fast approaches to improve the robustness of a railway timetable'. In: *Transportation Science* 43.3 (2009), pp. 321–335. DOI: 10.1287/trsc.1090.0264.

[Gat+05] M. Gatto, R. Jacob, L. Peeters and A. Schöbel. 'The Computational Complexity of Delay Management'. In: *Graph-Theoretic Concepts in Computer Science*. Vol. 3787. Lecture Notes in Computer Science. 2005, pp. 227–238. DOI: 10.1007/11604686_20.

[GL17] 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. Open Access Series in Informatics 12. 2017, pp. 1–12. DOI: 10.4230/OASIcs.ATMOS.2017.12.

[GMT14] V. Gabrel, C. Murat and A. Thiele. 'Recent advances in robust optimization: An overview'. In: *European Journal of Operational Research* 235.3 (2014), pp. 471–483. DOI: 10.1016/j.ejor.2013.09.036.

[Goe12] M. Goerigk. *A Local Search Algorithm for Robust Periodic Timetabling*. Tech. rep. NAM preprint series, Georg-August-Universität Göttingen, 2012.

[Goe15] M. Goerigk. 'Exact and heuristic approaches to the robust periodic event scheduling problem'. In: *Public Transport* 7.1 (2015), pp. 101–119. DOI: 10.1007/s12469-014-0100-5.

[Gov99] R. M. P. Goverde. 'Improving Punctuality and Transfer Reliability by Railway Timetable Optimization'. In: 1999.

[Gro+12] 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*. Vol. 7345. Lecture Notes in Computer Science. 2012, pp. 166–175. DOI: 10.1007/978-3-642-31087-4_18.

[Gro11] P. Großmann. *Polynomial Reduction from PESP to SAT*. Tech. rep. TU Dresden, 2011.

[GS10] M. Goerigk and A. Schöbel. 'An empirical analysis of robustness concepts for timetabling'. In: *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)*. Vol. 14. Open Access Series in Informatics (OASIcs). 2010, pp. 100–113. DOI: 10.4230/OASIcs.ATMOS.2010.100.

[GS13] M. Goerigk and A. Schöbel. 'Improving the modulo simplex algorithm for large-scale periodic timetabling'. In: *Computers and Operations Research* 40.5 (2013), pp. 1363–1370. DOI: 10.1016/j.cor.2012.08.018.

[GS14] M. Goerigk and A. Schöbel. 'Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling'. In: *Computers and Operations Research* 52.PART A (2014), pp. 1–15. DOI: 10.1016/j.cor.2014.06.025.

[GS16] M. Goerigk and A. Schöbel. 'Algorithm engineering in robust optimization'. In: *Algorithm Engineering: Selected Results and Surveys*. Vol. 9220. Lecture Notes in Computer Science. Springer International Publishing, 2016, pp. 245–279. DOI: 10.1007/978-3-319-49487-6_8.

[GS21] V. Grafe and A. Schöbel. 'Solving the Periodic Scheduling Problem: An Assignment Approach in Non-Periodic Networks'. In: *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021)*. Vol. 96. Open Access Series in Informatics (OASIcs). 2021, 9:1–9:16. DOI: 10.4230/OASIcs.ATMOS.2021.9.

[GS23] V. Grafe and A. Schöbel. 'Recoverable Robust Periodic Timetabling'. In: *23rd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2023)*. Vol. 115. Open Access Series in Informatics (OASIcs). 2023, 9:1–9:16. DOI: 10.4230/OASIcs.ATMOS.2023.9.

[GSS22] V. Grafe, A. Schiewe and A. Schöbel. 'Delay Management with Integrated Decisions on the Vehicle Circulations'. In: *22nd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2022)*. Vol. 106. Open Access Series in Informatics (OASIcs). 2022, 7:1–7:18. DOI: 10.4230/OASIcs.ATMOS.2022.7.

[Gur23] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2023. URL: https://www.gurobi.com.

[HK00] H. W. Hamacher and K. Klamroth. *Lineare und Netzwerk-Optimierung / Linear and Network-Optimization: Ein bilinguales Lehrbuch. A bilingual text- book*. Vieweg+Teubner Verlag Wiesbaden, 2000. DOI: 10.1007/978-3-322-91579-5.

[Jov+17] P. Jovanović, P. Kecman, N. Bojović and D. Mandić. 'Optimal allocation of buffer times to increase train schedule robustness'. In: *European Journal of Operational Research* 256.1 (2017), pp. 44–54. DOI: 10.1016/j.ejor.2016.05.013.

[KDV07] L. Kroon, R. Dekker and M. Vromans. 'Cyclic railway timetabling: A stochastic optimization approach'. In: *Algorithmic Methods for Railway Optimization*. Vol. 4359. Lecture Notes in Computer Science. 2007, pp. 41–68. DOI: 10.1007/978-3-540-74247-0_2.

[Kön20] E. König. 'A review on railway delay management'. In: *Public Transport* 12.2 (2020), pp. 335–361. DOI: 10.1007/s12469-020-00233-1.

[KP03] L. G. Kroon and L. W. Peeters. 'A variable trip time model for cyclic railway timetabling'. In: *Transportation Science* 37.2 (2003), pp. 198–212. DOI: 10.1287/trsc.37.2.198.15247.

[Kro+06] L. G. Kroon, G. Maróti, M. J. R. Helmrich, M. J. C. M. Vromans and R. Dekker. 'Stochastic Improvement of Cyclic Railway Timetables'. In: *European Economics: Microeconomics & Industrial Organization eJournal* 42.6 (2006), pp. 553–570. DOI: 10.1016/J.TRB.2007.11.002.

[Kro+09] L. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maróti, A. Schrijver, A. Steenbeek and R. Ybema. 'The New Dutch Timetable: The OR Revolution'. In: *Interfaces* 39.1 (2009), pp. 6–17. DOI: 10.1287/inte.1080.0409.

[KS11] N. Kliewer and L. Suhl. 'A note on the online nature of the railway delay management problem'. In: *Networks* 57.1 (2011), pp. 28–37. DOI: 10.1002/net.20381.

[Küm+15] M. Kümmling, P. Großmann, K. Nachtigall, J. Opitz and R. Weiß. 'A state-of-the-art realization of cyclic railway timetable computation'. In: *Public Transport* 7.3 (2015), pp. 281–293. DOI: 10.1007/s12469-015-0108-5.

[Lie+09] C. Liebchen, M. E. Lübbecke, R. H. Möhring and S. Stiller. 'The concept of recoverable robustness, linear programming recovery, and railway applications'. In: *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*. Vol. 5868. Lecture Notes in Computer Science.

Springer Berlin Heidelberg, 2009, pp. 1–27. DOI: 10.1007/978-3-642-05465-5_1.

[Lie+10] C. Liebchen, M. Schachtebeck, A. Schöbel, S. Stiller and A. Prigge. 'Computing delay resistant railway timetables'. In: *Computers & Operations Research* 37 (2010), pp. 857–868. DOI: 10.1016/j.cor.2009.03.022.

[Lie03] C. Liebchen. 'Finding short integral cycle bases for cyclic timetabling'. In: *Algorithms - ESA 2003*. Vol. 2832. Lecture Notes in Computer Science. 2003, pp. 715–726. DOI: 10.1007/978-3-540-39658-1_64.

[Lie06] C. Liebchen. 'Periodic timetable optimization in public transport'. PhD thesis. TU Berlin, 2006.

[Lie08] C. Liebchen. 'The First Optimized Railway Timetable in Practice'. In: *Transportation Science* 42.4 (2008), pp. 420–435. DOI: 10.1287/trsc.1080.0240.

[Lin00] T. Lindner. 'Train Schedule Optimization in Public Rail Transport'. PhD thesis. Technische Universität Braunschweig, 2000.

[LLB18] R. M. Lusby, J. B. Larsen and S. H. Bull. 'A survey on robustness in railway planning'. In: *European Journal of Operational Research* 266 (2018), pp. 1–15. DOI: 10.1016/j.ejor.2017.07.044.

[LM07] C. Liebchen and R. Möhring. 'The modeling power of the periodic event scheduling problem: Railway timetables - and beyond'. In: *Algorithmic Methods for Railway Optimization*. Vol. 4359. Lecture Notes in Computer Science. 2007, pp. 117–150. DOI: 10.1007/978-3-540-74247-0_1.

[LP02] C. Liebchen and L. Peeters. *On Cyclic Timetabling and Cycles in Graphs*. Tech. rep. TU Berlin, 2002.

[LP09] C. Liebchen and L. Peeters. 'Integral cycle bases for cyclic timetabling'. In: *Discrete Optimization* 6.1 (2009), pp. 98–109. DOI: 10.1016/j.disopt.2008.09.003.

[Lus+11] R. M. Lusby, J. Larsen, M. Ehrgott and D. Ryan. 'Railway track allocation: Models and methods'. In: *OR Spectrum* 33.4 (2011), pp. 843–883. DOI: 10.1007/s00291-009-0189-0.

[Mar17] G. Maróti. 'A branch-and-bound approach for robust railway timetabling'. In: *Public Transport* 9.1-2 (2017), pp. 73–94. DOI: 10.1007/s12469-016-0143-x.

[MLL23] B. Masing, N. Lindner and C. Liebchen. 'Periodic timetabling with integrated track choice for railway construction sites'. In: *Journal of Rail Transport Planning & Managegement* 28 (2023), p. 100416. DOI: 10.1016/j.jrtpm.2023.100416.

[Mül+21] M. Müller-Hannemann, R. Rückert, A. Schiewe and A. Schöbel. 'Towards improved robustness of public transport by a machine-learned oracle'. In: *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021)*. Vol. 96. Open Access Series in Informatics (OASIcs). 2021, 3:1–3:20. DOI: 10.4230/OASIcs.ATMOS.2021.3.

[Mül+22] M. Müller-Hannemann, R. Rückert, A. Schiewe and A. Schöbel. 'Estimating the robustness of public transport schedules using machine learning'. In: *Transportation Research Part C: Emerging Technologies* 137 (2022), p. 103566. DOI: 10.1016/j.trc.2022.103566.

[Nac96] K. Nachtigall. 'Periodic network optimization with different arc frequencies'. In: *Discrete applied mathematics* 69 (1996), pp. 1–17. DOI: 10.1016/0166-218X(95)00073-Z.

[Nac98] K. Nachtigall. 'Periodic network optimization and fixed interval timetables'. Habilitation. University of Hildesheim, 1998.

[NO08] 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. Open Access Series in Informatics (OASIcs). 2008, pp. 1–15. DOI: 10.4230/OASIcs.ATMOS.2008.1588.

[NV96] K. Nachtigall and S. Voget. 'A genetic algorithm approach to periodic railway synchronization'. In: *Computers and Operations Research* 23.5 (1996), pp. 453–463. DOI: 10.1016/0305-0548(95)00032-1.

[NW88] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988. DOI: 10.1002/9781118627372.

[Odi96] M. A. Odijk. 'A constraint generation algorithm for the construction of periodic railway timetables'. In: *Transportation Research Part B: Methodological* 30.6 (1996), pp. 455–464. DOI: 10.1016/0191-2615(96)00005-7.

[Pät21] J. Pätzold. 'Finding robust periodic timetables by integrating delay management'. In: *Public Transport* 13.2 (2021), pp. 349–374. DOI: 10.1007/s12469-020-00260-y.

[Pee03] L. Peeters. 'Cyclic Railway Timetable Optimization'. PhD thesis. Erasmus Universiteit Rotterdam, 2003.

[PK01] L. Peeters and L. Kroon. 'A Cycle Based Optimization Model for the Cyclic Railway Timetabling Problem'. In: *Computer-Aided Scheduling of Public Transport*. Springer Berlin Heidelberg, 2001, pp. 275–296. DOI: 10.1007/978-3-642-56423-9_16.

[Pol+19] G.-J. Polinder, T. Breugem, T. Dollevoet and G. Maróti. 'An adjustable robust optimization approach for periodic timetabling'. In: *Transportation Research Part B: Methodological* 128 (2019), pp. 50–68. DOI: 10.1016/j.trb.2019.07.011.

[PS16] 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. Open Access Series in Informatics (OASIcs). 2016, pp. 1–15. DOI: 10.4230/OASIcs.ATMOS.2016.1.

[Rob+16] T. Robenek, Y. Maknoon, S. S. Azadeh, J. Chen and M. Bierlaire. 'Passenger centric train timetabling problem'. In: *Transportation Research Part B: Methodological* 89 (2016), pp. 107–126. DOI: 10.1016/j.trb.2016.04.003.

[Rüc+17] R. Rückert, M. Lemnian, C. Blendinger, S. Rechner and M. Müller-Hannemann. 'PANDA: a software tool for improved train dispatching with focus on passenger flows'. In: *Public Transport* 9.1-2 (2017), pp. 307–324. DOI: 10.1007/s12469-016-0140-0.

[Sch+] P. Schiewe, A. Schöbel, S. Jäger, S. Albert, U. Baumgart, C. Biedinger, V. Grafe, S. Roth, A. Schiewe, F. Spühler, M. Stinzendörfer and R. Urban. *LinTim - Integrated Optimization in Public Transportation*. Homepage. URL: https://www.lintim.net.

[Sch+23] P. Schiewe, A. Schöbel, S. Jäger, S. Albert, U. Baumgart, C. Biedinger, V. Grafe, S. Roth, A. Schiewe, F. Spühler, M. Stinzendörfer and R. Urban. *LinTim: An integrated environment for mathematical public transport optimization. Documentation for version 2023.12*. University of Kaiserslautern-Landau. 2023. URL: https://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-75699.

[Sch01] A. Schöbel. 'A model for the delay management problem based on mixed-integer-programming'. In: *Electronic Notes in Theoretical Computer Science* 50.1 (2001), pp. 1–10. DOI: 10.1016/S1571-0661(04)00160-4.

[Sch07] A. Schöbel. 'Integer Programming Approaches for Solving the Delay Management Problem'. In: *Algorithmic Methods for Railway Optimization*. Vol. 4359. Lecture Notes in Computer Science. 2007, pp. 145–170. DOI: 10.1007/978-3-540-74247-0_7.

[Sch09] M. Schachtebeck. 'Delay Management in Public Transportation: Capacities, Robustness, and Integration'. PhD thesis. Georg-August-Universität Göttingen, 2009.

[Sch12] A. Schöbel. 'Line planning in public transportation: Models and methods'. In: *OR Spectrum* 34.3 (2012), pp. 491–510. DOI: 10.1007/s00291-011-0251-6.

[Sch14] A. Schöbel. 'Generalized light robustness and the trade-off between robustness and nominal quality'. In: *Mathematical Methods of Operations Research* 80.2 (2014), pp. 161–191. DOI: 10.1007/s00186-014-0474-9.

[SK09] A. Schöbel and A. Kratz. 'A bicriteria approach for robust timetabling'. In: *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*. Vol. 5868. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 119–144. DOI: 10.1007/978-3-642-05465-5_5.

[Soy73] A. L. Soyster. 'Technical Note—Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming'. In: *Operations Research* 21.5 (1973), pp. 1154–1157. DOI: 10.1287/opre.21.5.1154.

[SS10] M. Schachtebeck and A. Schöbel. 'To wait or not to wait-and who goes first? Delay management with priority decisions'. In: *Transportation Science* 44.3 (2010), pp. 307–321. DOI: 10.1287/trsc.1100.0318.

[SS15] M. Schmidt and A. Schöbel. 'Timetabling with passenger routing'. In: *OR Spectrum* 37.1 (2015), pp. 75–97. DOI: 10.1007/s00291-014-0360-0.

[SS20] P. Schiewe and A. Schöbel. 'Periodic timetabling with integrated routing: Towards applicable approaches'. In: *Transportation Science* 54.6 (2020), pp. 1714–1731. DOI: 10.1287/trsc.2019.0965.

[SU89] P. Serafini and W. Ukovich. 'A Mathematical Model for Periodic Scheduling Problems'. In: *SIAM Journal on Discrete Mathematics* 2 (1989), pp. 550–581. DOI: 10.1137/0402049.

[VV06] P. Vansteenwegen and D. Van Oudheusden. 'Developing railway timetables which guarantee a better service'. In: *European Journal of Operational Research* 173.1 (2006), pp. 337–350. DOI: 10.1016/j.ejor.2004.12.013.

NOTATION

$\mathcal{L}$	line plan, 7
$\underline{\mathcal{N}}$	periodic EAN, 9
$\mathcal{N}$	rolled out EAN, 13
$\underline{\mathcal{E}}$	events in the periodic EAN, 8
$\underline{\mathcal{E}}^{\text{dep}}$	departure events in the periodic EAN, 8
$\underline{\mathcal{E}}^{\text{arr}}$	arrival events in the periodic EAN, 8
$\mathcal{E}$	events in the EAN, 12, 27
$\underline{\mathcal{A}}$	activities in the periodic EAN, 8
$\underline{\mathcal{A}}_{\text{drive}}$	driving activities in the periodic EAN, 8
$\underline{\mathcal{A}}_{\text{wait}}$	waiting activities in the periodic EAN, 8
$\underline{\mathcal{A}}_{\text{transfer}}$	transfer activities in the periodic EAN, 8
$\underline{\mathcal{A}}_{\text{head}}$	headway activities in the periodic EAN, 8
$\underline{\mathcal{A}}_{\text{sync}}$	synchronisation activities in the periodic EAN, 9
$\mathcal{A}$	activities in the EAN, 12, 27
$\mathcal{A}_{\text{train}}$	driving and waiting activities in the EAN, 13
$\mathcal{A}_{\text{out}}$	activities in the EAN leaving the planning horizon, 46
$\mathcal{A}'_{\text{head}}$	headway activities used for the DM subproblem, 46
L	lower bounds for activity durations, 9
U	upper bounds for activity durations, 9
w	passenger weights, 11
T	period length, 8
$I = [t_{\min}, t_{\max}]$	planning horizon, 12
K	number of periods in I, 27
d	source delays, 13
$\mathcal{U}$	uncertainty set, 16
$\text{goi}(x)$	grade of infeasibility, 17
$Z_1^r(\pi, x)$	total event delay weighted with the number of passengers in scenario r when using timetable π and disposition timetable x, 42, 46
$Z_1(\pi, x)$	maximal total event delay weighted with the number of passengers when using timetable π and disposition timetable x, 42
$Z_2^r(\pi, y)$	number of missed tranfers in scenario r when using timetable π and wait/no-wait decisions y, 42

$Z_2(\pi, y)$	maximal number of missed tranfers when using timetable π and wait/no-wait decisions y, 42
f^{PESP}	objective function of PESP, 11
$f^{\text{nom}}(\pi)$	nominal travel time of timetable π, 43
$f^{\text{del}}(\pi)$	worst-case delay of timetable π, 43
$f^{\text{real}}(\pi)$	real travel time of timetable π, 43
π^{RR}	optimal solution to RRPT, 44
$\pi^{\text{RR},\alpha,\beta}$	optimal solution to RRPT(α, β), 45
$\pi^{\text{RR},\bar{f}}$	optimal solution to RRPT$(\bar{f})$, 45
π^{SR}	optimal solution to SRPT, 88
$\pi^{\text{LR},\rho}$	optimal solution to LRPT(ρ), 90
$\pi^{\text{AR},\underline{\mathcal{E}}^2}$	optimal solution to ARPT$(\underline{\mathcal{E}}^2)$, 93

ACRONYMS

ARC	adjustable robust counterpart, 17
ARPT	adjustable robust periodic timetabling problem, 92
DM	delay management problem, 13
DM($\mathcal{U}$)	delay management problem solving DM(r) for all $r \in \mathcal{U}$, 43
DM(r)	delay management problem corresponding to delay scenario r, 42
EAN	event-activity network, 8
LRC	lightly robust counterpart, 17
LRPT	lightly robust periodic timetabling problem, 89
MRRPT	multi-objective recoverable robust periodic timetabling, 43
PESP	periodic event scheduling problem, 10
PTN	public transport network, 7
PTTA	periodic timetabling in aperiodic network, 27
RRPT	recoverable robust periodic timetabling, 44
RRPT(α, β)	recoverable robust periodic timetabling with restricted delay, 45
RRPT($\bar{f}$)	recoverable robust periodic timetabling with restricted nominal travel time, 45
SRC	strictly robust counterpart, 16
SRPT	strictly robust periodic timetabling problem, 88

Vera Grafe

Curriculum Vitae

since 10/2020	Doctoral Studies in Mathematics, RPTU in Kaiserslautern (former TU Kaiserslautern)
since 10/2020	Research Assistant, RPTU in Kaiserslautern (former TU Kaiserslautern)
09/2020	Master of Science in Mathematics, TU Kaiserslautern
10/2019 – 09/2020	Teaching Assistant, TU Kaiserslautern
10/2018 – 09/2020	Master Studies in Mathematics, TU Kaiserslautern
09/2018	Bachelor of Science in Mathematics, TU Kaiserslautern
10/2015 – 09/2018	Bachelor Studies in Mathematics, TU Kaiserslautern
03/2015	Abitur, Are-Gymnasium, Bad Neuenahr-Ahrweiler

Vera Grafe

Wissenschaftlicher Werdegang

seit 10/2020	Promotionsstudium in Mathematik, RPTU in Kaiserslautern (ehemals TU Kaiserslautern)
seit 10/2020	Wissenschaftliche Mitarbeiterin, RPTU in Kaiserslautern (ehemals TU Kaiserslautern)
09/2020	Master of Science in Mathematik, TU Kaiserslautern
10/2019 – 09/2020	Wissenschaftliche Hilfskraft, TU Kaiserslautern
10/2018 – 09/2020	Masterstudium in Mathematik, TU Kaiserslautern
09/2018	Bachelor of Science in Mathematik, TU Kaiserslautern
10/2015 – 09/2018	Bachelorstudium in Mathematik, TU Kaiserslautern
03/2015	Abitur, Are-Gymnasium, Bad Neuenahr-Ahrweiler

COLOPHON

This document was typeset using the LaTeX package `classicthesis` developed by André Miede and Ivo Pletikosić. It is available at:

https://bitbucket.org/amiede/classicthesis/