Mathias Kinder

MODELS FOR PERIODIC TIMETABLING

Diplomarbeit bei Prof. Dr. M. Grötschel

Technische Universität Berlin

Mathias Kinder: *Models for Periodic Timetabling*, 15. Mai 2008

Die selbstständige und eigenhändige Anfertigung versichere ich an Eides Statt.

______________________________ ______________________________

(Ort, Datum) (Unterschrift)

Acknowledgments

I wish to thank a number of people who have provided advice, encouragement, comments, questions, and answers. At the very least, I like to mention the following individuals: Ralf Borndörfer, Stefan Heinz, Christian Liebchen, Marika Neumann, Marc Pfetsch, Markus Reuther, Elmar Swarat, and Roland Wessäly. I am sure that I have forgotton a few people, to whom I apologize.

I am also indepted to the PTV AG for providing their VISUM software and two interesting test instances.

Abstract

We investigate the computation periodic timetables for public transport by mixed integer programming. After introducing the problem, we describe two mathematical models for periodic timetabling, the PERIODIC EVENT SCHEDULING PROBLEM (PESP) and the QUADRATIC SEMI-ASSIGNMENT PROBLEM. Specifically, we give an overview of existing integer programming (IP) formulations for both models.

An important contribution of our work are new IP formulations for the PESP based on time discretization. We provide an analytical comparison of these formulations and describe different techniques that allow a more efficient solution by mixed integer programming.

In a preliminary computational study, on the basis of standard IP solvers, we compare different formulations for computing periodic timetables. Our results justify a further investigation of the time discretization approach.

Typically the timetable is optimized for the current traffic situation. The main difficulty with this approach is that after introducing the new timetable the passengers' travel behavior may differ from that assumed for the computation. Motivated by this problem, we examine an iterative timetabling procedure that is a combination of timetable computation and passenger routing. We discuss the algorithmic issues of the passenger routing and study properties of the computed timetables. Finally, we confirm our theoretical results on the basis of an own implementation.

Zusammenfassung

Wir untersuchen die Berechnung von Taktfahrplänen für den öffentlichen Verkehr mit gemischt-ganzzahliger Programmierung (MIP). Im Anschluss an die Problembeschreibung, stellen wir zwei mathematische Modellierungen vor, das PERIODIC EVENT SCHEDULING PROBLEM (PESP) und das QUADRATIC SEMI-ASSIGNMENT PROBLEM. Wichtiger Bestandteil ist ein Überblick über existierende ganzzahlige Formulierungen beider Modelle.

Wir entwickeln neue ganzzahlige Formulierungen für das PESP auf der Basis von Zeitdiskretisierung. Diese werden analytisch miteinander verglichen und wir

beschreiben verschiedene Techniken, die eine effizientere Lösung der Formulierungen mit gemischt-ganzzahliger Programmierung ermöglichen.

In einer ersten Rechenstudie, unter Verwendung gängiger MIP-Löser, vergleichen wir verschiedene ganzzahlige Formulierungen zur Berechnung von Taktfahrplänen. Unsere Ergebnisse rechtfertigen eine weitere Untersuchen des Zeitdiskretisierungsansatzes.

In der Regel werden Fahrpläne mit Bezug auf die gegenwärtige Verkehrssituation optimiert. Dies birgt jedoch folgendes Problem. Wenn der neue Fahrplan eingeführt wird, ist es möglich, dass die Passagiere ein anderes Fahrverhalten zu Tage legen, als für die Berechnung des Fahrplans angenommen wurde. Vor diesem Hintergrund behandeln wir ein iteratives Verfahren zur Berechnung von Taktfahrplänen. Dieses ist eine Kombination aus Fahrplanberechnung und Passagierrouting. Neben den algorithmischen Details des Passagierroutings untersuchen wir Eigenschaften der berechneten Fahrpläne. Abschließend bestätigen wir unsere theoretischen Ergebnisse auf Grundlage einer eigenen Implementierung des Verfahrens.

Contents

List of Figures

List of Tables

Preface

The last several years have seen a tremendous research activity in the area of mathematical optimization in public transport. This thesis studies existing and new formulations of mathematical models for computing high quality periodic timetables.

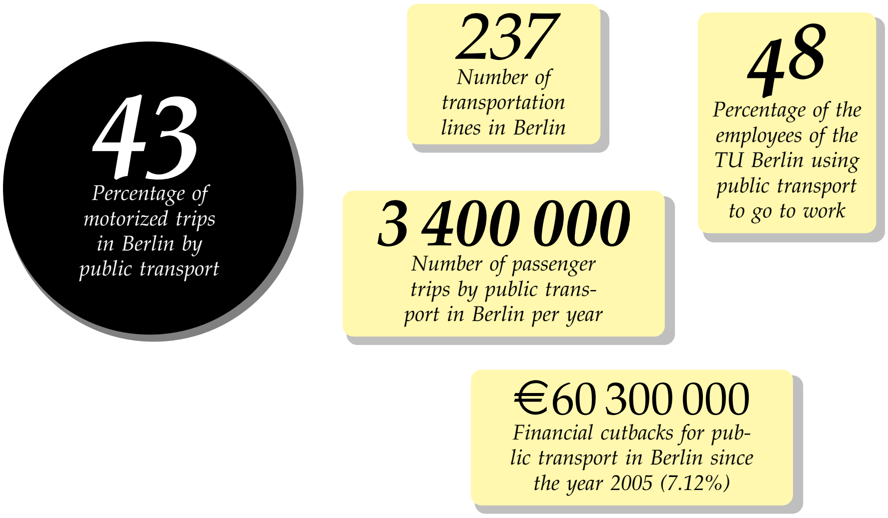
A periodic timetable has the property that certain departure times reoccur in fixed intervals which ensures a very regular service. This is one reason why periodic timetables are widely used for local, regional, and long-distance traffic.

Public transport plays an important role in providing mobility for many people and a customer-friendly service requires reasonable travel times. The timetable is thereby a crucial factor. By coordinating the arrival and departure times of the (public transport) vehicles we can influence the waiting times for transfers. The coordination is a highly complex task, particularly because in practice the timetable underlies several other requirements.

We apply the theory of time discretization to periodic timetabling. The result of this attempt are new integer programming formulations which we compare, analytically and computationally, to existing formulations. Furthermore, we investigate an iterative approach for computing periodic timetables with the aim to incorporate the travel behavior of the passengers.

Overview

In Chapter 1 we establish the relation between timetabling and other planning tasks in public transport, such as line planning and vehicle scheduling. The chapter gives an introduction into periodic timetabling and discusses typical timetable requirements. This chapter covers the most basic material on the topic. Chapter 2 surveys two existing mathematical models for computing periodic timetables. First, we present the PERIODIC EVENT SCHEDULING PROBLEM (PESP) that provides rich modeling capabilities and various integer programming formulations. It follows a discussion of the QUADRATIC SEMI-ASSIGNMENT PROBLEM, a quadratic model that is more specialized and that focusses on the minimization of transfer waiting times. In Chapter 3 we develop new integer programming formulations for the PESP based on time discretization. We provide an analytical comparison of the formulations and investigate promising techniques to reduce computation time when applying mixed integer programming. Chapter 4 studies a new iterative approach for computing periodic timetables. The idea is to further


43
Percentage of motorized trips in Berlin by public transport
237
Number of transportation lines in Berlin
48
Percentage of the employees of the TU Berlin using public transport to go to work
3 400 000
Number of passenger trips by public transport in Berlin per year
€60 300 000
Financial cutbacks for public transport in Berlin since the year 2005 (7.12%)

Berlin public transport in figures.
Sources: Nahverkehrsplan Berlin 2006–2009, Department of Integrated Traffic Planning TU Berlin.

improve the timetable by incorporating the resulting travel behavior of the passengers. Chapter 5 concludes with a computational study that evaluates a large set of integer programming formulations for timetabling on five real-world test instances. It provides results on the speedup techniques developed for the new PESP formulations. Furthermore, we present preliminary results on the new iterative timetabling procedure described in Chapter 4.

1

Timetabling

1.1 The Planning Process in Public Transport

The planning process in public transport is divided into two major stages — strategic planning and operational planning. Due to the high complexity, each of these stages is further divided into a sequence of subtasks. The strategic part concerns network planning, line planning, and timetabling. Important tasks of operational planning are vehicle and crew scheduling. Some authors see timetabling even as a link between the two main planning units.

There are further tasks a public transport company has to deal with. For example, a fare system has to be designed, and a timetable information system needs to be managed. Since these tasks are not directly related to the timetable generation, we will not discuss them here.

1.1.1 Strategic Planning

Strategic planning is made for a planning horizon of 5 – 15 years. In the following, we provide a brief discussion of relevant subtasks.

Travel Demand Estimation
Before strategic planning starts, the travel demand is estimated. This is not only necessary for the dimensioning of the infrastructure of the transportation system, but also for the choice of the transportation routes that are to be offered to the passengers.

The basis for the demand estimation is a subdivision of the region into *traffic zones*. Every traffic zone has a designated *centroid* representing its center (see Figure 1.1). At these centroids traffic originates or terminates. The stations close to a centroid are the entry points to the transportation network. Figure 1.2 illustrates this. A *station* is a regular stopping place, where (public transport) vehicles and passengers arrive and depart.

Travel demand is measured as the number of passengers that want to travel from an origin zone to a destination zone within a fixed period called *demand pe-*

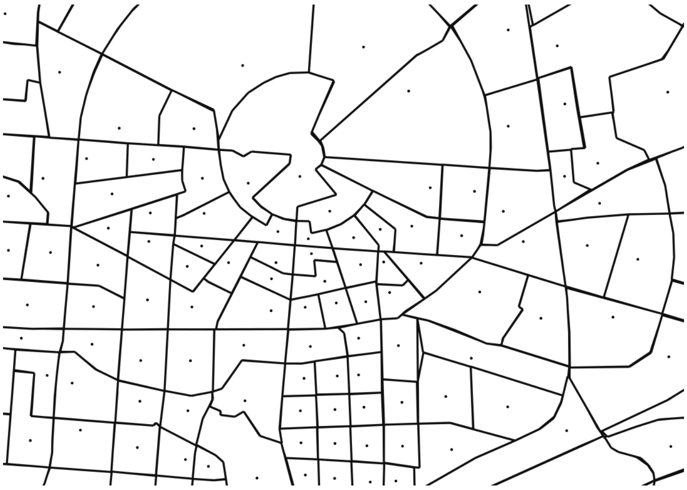


Figure 1.1 City of Karlsruhe: traffic zones with designated centroids (screenshot from VISUM 10.0).

riod. The demand estimates for all origin-destination pairs are commonly summarized in a so-called *origin-destination matrix* (or OD-matrix for short).

Standard approaches for the travel demand estimation are interviews of customers, evaluation of ticket sales, and various statistical methods on the basis of automated passenger counts. All these techniques are very costly and time consuming.

Despite the fact that OD-matrices are widely used for recording of travel demand, they also have drawbacks. The resulting estimates highly depend on the size and the distribution of the traffic zones. Furthermore, the OD-matrix captures only a snapshot of the travel demand. Therefore, it is not always clear how well the real situation is reflected in this type of data.

Network Planning

Network planning focuses on the infrastructure of the transportation system. This includes streets for road traffic as well as tracks for rail traffic. Typically an existing network has to be modified due to changes of the travel demand or new capacity requirements. The usual objective is to minimize the construction cost while still ensuring the expected travel demand.

Line Planning

Once the infrastructure of the transportation system is determined, lines have to be defined and associated with individual frequencies. A *line* is a transportation route

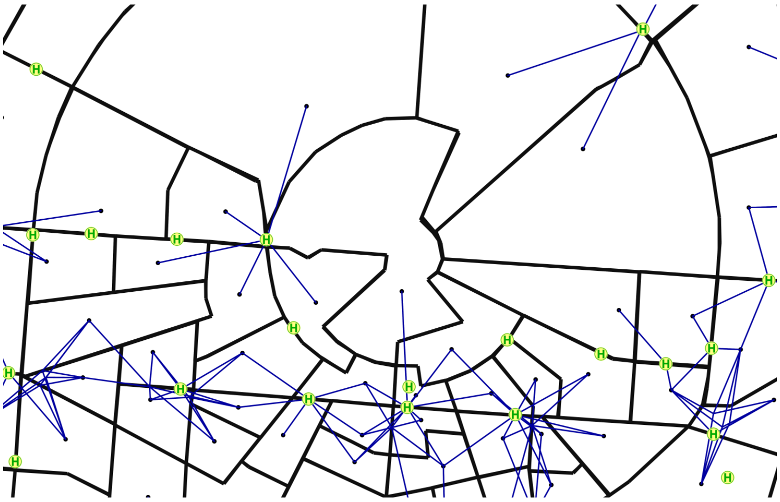


Figure 1.2 City of Karlsruhe: traffic zones with designated centroids linked to close by stations (screenshot from VISUM 10.0).

between two designated terminal stations in the transportation network (see Figure 1.3 for an example). A line has a *transportation mode*, such as subway, regional railway, or intercity railway. Figure 1.4 shows a selection of lines of the city of Karlsruhe. The *frequency* specifies the number of times the line is served depending on the demand period. This could be ten times an hour in rush-hour traffic and two times an hour in off-hour traffic.

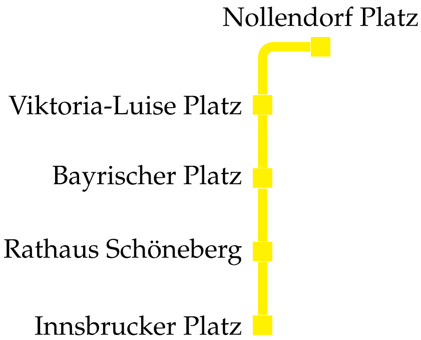


Figure 1.3 Network diagram of the Berlin subway line U4 as of May 2008.

During the line planning process, several constraints are taken into account. In particular, the final line plan has to meet the travel demand and respect existing track capacities.

On the one hand, it is important to offer a service that is attractive for passengers. This includes reasonable travel times and preferably less transfers between the origin and the destination. On the other hand, it is desirable to minimize the lines' operating cost.

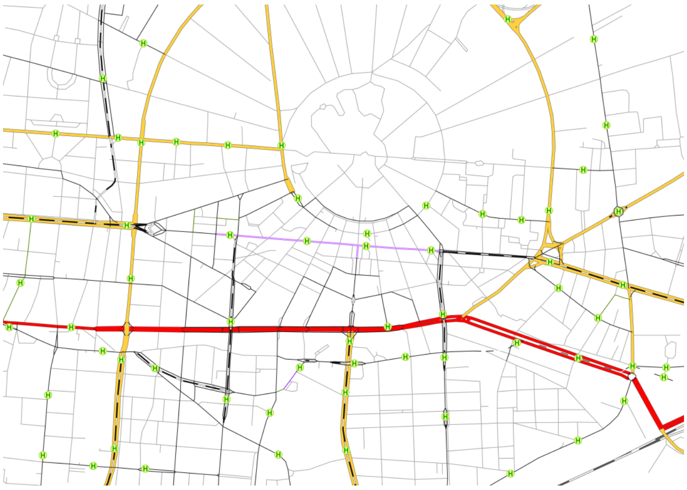


Figure 1.4 City of Karslruhe: transportation lines (screenshot from VISUM 10.0).

Timetabling

The next step is to schedule the trips of each line. A *trip* is the movement of a vehicle between the terminal stations of a line. Scheduling the trips includes to determine the arrival and departure times at the stations in the transportation network.

The resulting timetable is subject to several requirements, e. g. synchronization constraints or safety regulations which we discuss later. These constraints have to be considered, when optimizing for certain objectives like minimizing transfer waiting times or minimizing the number of vehicles required to operate the timetable.

1.1.2 Operational Planning

After timetabling the operational planning starts. We next give a brief exposition of important subtasks that cover a planning horizon of 24h up to 1 year.

Vehicle Scheduling
With the timetable in hand, the task is to combine timetabled trips to so-called *blocks*. Each of these blocks is assigned a vehicle of certain type whose *schedule* is given by the trips in the block.

During the construction of the vehicle schedule one has to keep in mind that not every vehicle is adequate for each trip and that there is only a limited number of vehicles of each type.

One of the main objectives at this stage is to minimize the number of blocks in the peak hours which has direct influence on the required size of the rolling stock and thus on the operating cost. In addition, it is desirable to minimize the period between two linked trips to give standard working conditions.

Crew Scheduling
Following the vehicle scheduling, blocks or pieces of blocks of the vehicle schedule have to be combined to blocks of duties. The goal is to define a set of legal shifts that cover all blocks of the vehicle schedule. The focus is on the construction of a short-term schedule for the crews considering a planning horizon of 24 to 48 hours.

Typical constraints are work regulations like restrictions on the total working time or maximum working hours per day. The difficulty is to accommodate the minimization of the number of duties required to cover the vehicle schedule and the crew satisfaction.

1.2 Periodic Timetabling

We are interested in computing periodic timetables. Therefore, let us discuss this planning step in detail.

From the line plan we obtain the lines together with their frequencies for different demand periods. The task of timetabling is now to schedule the trips of all lines which yields the timetable. Scheduling a trip of a line requires to define the arrival and departure times at each of its stations.

1.2.1 Timetable Requirements

In order that a timetable is operable in practice, it has to fulfill certain requirements. These range from elementary constraints concerning the dwell time at stations to more sophisticated constraints as the coordination of different lines to en-able transfers. We restrict our attention to those timetable requirement that are relevant for the thesis.

❏ **Trip and dwell time constraints.** After a vehicle arrives at a station passengers have to have enough time to board and alight, i. e., a minimum dwell time of the vehicle is required. On the other hand, it may also be useful to limit the dwell time within the station. This is because every waiting minute adds to the total travel time of the passengers who are already on the vehicle and a station has a limited capacity. Finally, the running times of the vehicles between stations have to be respected.

❏ **Synchronizing lines.** Coordinating the trips of different lines is desirable for several reasons: to offer customer-friendly transfer relations, to guarantee a balanced service on tracks operated by several lines, and for combining vehicles to reduce network utilization and operating cost.

If passengers travel between locations in the transportation network for which no direct connection exists, transfers become inevitable. In order to prevent the passengers from unnecessary waiting times, the departure of the connecting vehicle should be shortly after the arrival at the transfer station. However, the time required for possible platform changes has to be respected.

In a transportation network different lines may share sections of route. Without synchronization, the vehicles may arrive at the common stations in quick succession which surely does not yield a very balanced service. Ideally we ensure a nearly equal *headway time* (interval between the vehicles).

It is also possible to combine vehicles of lines on the common part of their routes. Although, this does not increase the service frequency, more direct connections can be offered without increasing the network utilization (otherwise two vehicles occupy the track with a certain headway). Furthermore, operating one vehicle requires less personnel on the common part that can be saved or used for customer service. However, for the combination, both vehicles must be at the station which adds additional constraints to the timetable and the procedure requires some time (e. g. for break test, and other checks).

❏ **Turnarounds at termini.** If a vehicle reaches the end of the route, it turns around and either serves the back direction of the line or another connection. Before that, some time is required, e. g. for shunting, cleaning the vehicle, or idle time of the driver. In either case, a feasible timetable has to respect this minimum turnaround time. If, however, the vehicle dwells too long at the terminus, additional vehicles may be required to operate the timetable (additional costs, e. g. for crew).

❑ **Hierarchical planning.** Transportations systems are typically classified by priority, e. g., inter city trains are superior to local trains. The planning process then follows a top down approach by first scheduling the lines of high priority. Therefore, the construction of timetables for lines of low-level transportation systems has to be done without altering already existing timetables of higher level lines. Similar restrictions apply to international trains whose departures at the border have to be coordinated with transportation systems of the neighboring country.

❑ **Safety requirements.** In a transportation network, many vehicles are in transit simultaneously and often they use the same tracks. To prevent collisions and overtaking, minimum headway times between the vehicles have to be ensured. This especially applies to single tracks operated in both directions.

In practice it is not sufficient when a timetable just satisfies the above requirements. Depending on the point of view, there are several criteria that make a timetable reasonable.

For customer satisfaction the total travel time is a crucial factor. Therefore, we are interested in minimizing additional dwell and transfer times. Furthermore, a viable timetable is expected to be robust against delays of vehicles and requires the minimum number of vehicles.

1.2.2 Timetable Construction

In practice, timetabling is mainly a human planning process supported with computer-aided design tools. Examples for such tools are MICROBUS of the IVU AG and the Siemens route management software ROMAN.

A timetable is not always constructed from scratch. Often an existing timetable has to be modified, because of changes in the line plan. Suppose, for example, that some lines were introduced or others removed.

There are mainly two timetabling strategies. One possibility is to schedule the trips of a line individually without any structure. A standard approach, however, is to ensures a fixed interval between them. The latter strategy is called *fixed interval timetabling* or *periodic timetabling*. Obviously, the resulting *periodic timetables* are easier noticeable. A main advantage, however, lies in the fact that due to the very regular service even spontaneous trips involve a reasonable travel time. Therefore, many transportation companies use periodic timetables today. In Figure 1.5 we give a brief historical overview.

London subway	1863	
	1896	*Budapest subway*
Paris subway	1900	
	1902	*Berlin subway*
Nederlandse Sporwegen Netherlands	1939	
	1974	*DSB Denmark*
"Jede Stunde, jede Klasse" — DB Germany	1979	
	1982	*"Jede Stunde ein Zug" — SBB Switzerland*
Belgium and Austria	1991	

Figure 1.5 History of the periodic timetable. 1863 to the present.

The fixed interval between the trips of a line is called the *period time* and will be denoted by T. The period time depends on the frequency of the line which may vary over the day. An example of a periodic timetable is shown in Table 1.1. In rush-hour traffic, from 6:00 am to 8:20 am, the line is operated 28 times. This corresponds to a frequency of 28/140 min. The period time is given as the reciprocal of the frequency, i. e. $T = 5\,\text{min}$.

An important point to note is that the construction of timetables can be supported by discrete optimization algorithms. In 2005, the first optimized periodic timetable was successfully put into daily operation for the Berlin subway (see Liebchen (2006)).

Table 1.1 Station timetable of the Berlin subway line U4 at Nollendorf Platz valid from 18 April to 1 November 2008.

h	Mondays to Fridays											
⋮												
05	07	17	27	37	47	57						
06	02	07	12	17	22	27	32	37	42	47	52	57
07	02	07	12	17	22	27	32	37	42	47	52	57
08	02	07	12	17	27	37	47	57				
09	07	17	27	37	47	57						
⋮												

The next chapters answer the question how periodic timetabling can be expressed by mathematical models. The objective is to automatically compute high-quality periodic timetables.

1.3 Prerequisites

As for prerequisites, the reader is expected to be familiar with elementary graph theory (see e. g. Schrijver (2003)). The reader with no exposure to mathematical programming is referred to Nemhauser and Wolsey (1999) for a good introduction. The few facts about $\mathcal{NP}$-completeness used here are covered in Garey and Johnson (1979).

2 Models for Periodic Timetabling

In the preceding chapter we presented timetable requirements that typically arise in practice. Now, we give an exposition of two important models for creating periodic timetables that cover a large set of these requirements. After defining the general scope in Section 2.1, we present in Section 2.2 the PERIODIC EVENT SCHEDULING PROBLEM. In Section 2.3 we proceed with the study of the QUADRATIC SEMI-ASSIGNMENT PROBLEM and illustrate its application for periodic timetabling.

2.1 Model Assumptions

This section defines the general conditions under which we investigate the models for periodic timetabling.

Our purpose is to compute a periodic timetable for a single demand period, e. g. the morning peak hours, during which the period times of all lines are assumed be fixed. The full-day timetable is then obtained by gluing together the timetables of all demand periods. This may require minor changes at the transitions.

In the following, we need the set of stations $\mathcal{S}$ and the line plan that specifies the set of operated lines $\mathcal{R}$ and their frequencies. Recall from Chapter 1 that a line $R \in \mathcal{R}$ is a transportation route between to designated terminal stations in the transportation network. From now on, we consider for each line $R \in \mathcal{R}$ a forward and a backward direction. Each direction is a *directed line* and we omit the prefix "directed" when no confusion can arise. Let us denote by $\mathcal{L}$ the set of directed lines. In addition, the timetable computation requires information on the minimum time duration of trips, stops, turnarounds, transfers, etc.

We also need a *passenger flow* that corresponds to the estimated travel demand for the demand period. If $L \in \mathcal{L}$ is a directed line, then the passenger flow has to define

❏ for any two consecutive stations S and S' on the route of line L, the total number of passengers involved in trips of line L between S and S', and

❏ for any station S on the route of line L, the total number of passengers involved in stops of line L at station S, where these numbers apply to the whole demand period. Moreover, the passenger flow has to specify how many passengers use a certain transfer connection. We define a *transfer connection* as a 3-tuple (L, L', S) which means that passengers can change between the lines L and L' at station S. This is for simplicity. In practice, a transfer connection may require to change between stations.

With this information, we intend to construct a periodic timetable that meets the requirements discussed in Section 1.2.1. In particular, we have to determine the arrival and departure times for the lines at all stations.

2.2 The Periodic Event Scheduling Problem (PESP)

This section reviews the PERIODIC EVENT SCHEDULING PROBLEM, a model that has extensively been studied for constructing periodic timetables. The first applications of the PESP, however, were not concerned with periodic timetabling at all. Introduced by Serafini and Ukovich (1989a) as a model for a periodic job shop, it has also been used for traffic light scheduling (Serafini and Ukovich, 1989b) and airline scheduling (Gertsbakh and Serafini, 1991). To simplify the presentation, we investigate the basic model which requires identical line frequencies, i. e., we construct the timetable for a uniform period time $T \in \mathbb{N}$. This condition is not necessary for the EXTENDED PERIODIC EVENT SCHEDULING PROBLEM for which details can be found in Serafini and Ukovich (1989a) and Nachtigall (1996b).

The PESP can be formulated in many ways. The following seems most appropriate for our presentation.

Periodic Event Scheduling Problem (PESP)

Instance: Directed graph $D = (V, A)$, vectors $\ell, u \in \mathbb{Q}^A$, and an integer T.

Question: Is there a vector $\pi \in [0, T)^V$ such that

$$(\pi_w - \pi_v - \ell_a) \bmod T \leq u_a - \ell_a \tag{2.1}$$

for all $a = (v, w) \in A$?

In the above description, T denotes the period time and mod stands for the modulo operator that is defined as

$$a \bmod b = a - b \cdot \max\{z \in \mathbb{Z} \mid a - b \cdot z \geq 0\} \in [0, b - 1),$$

for $a, b \in \mathbb{R}$. The timetabling requirements are expressed by the constraints (2.1) which we call *periodic interval constraints*.

Before we give a complete interpretation of the PESP in terms of periodic timetabling, we need to discuss some preliminaries.

An arrival or departure of a directed line at a station is called an *event*. We use the notation $arr(L, S)$ for the arrival of line L at station S, and $dep(L, S)$ for the departure of line L from station S. Let V denote the set of all events in the transportation network. Then, the vector $\pi \in [0, T)^V$ defines the arrival and departure times in the timetable. We call π_v the *timing* of the event v.

A pair of events $(v, w) \in V \times V$ is referred to as an *activity* and we write A for the set of all activities. Each activity $a \in A$ is associated with a minimum and maximum allowable time duration, denoted by ℓ_a and u_a, respectively. The interval $[\ell_a, u_a]$ is referred to as the *time window* of activity a.

The following are examples of activities. The pair of events

- $(dep(L, S), arr(L, S'))$ is a *trip activity*, where S and S' denote two consecutive stations on the route of line L,
- $(arr(L, S), dep(L, S))$ is a *dwell activity*, where S denotes a station on the route of line L,
- $(arr(L, S), dep(\overline{L}, S))$ is a *turn activity*, where L and $\overline{L}$ denote two opposite directed lines and S is the terminal station of line L, and
- $(arr(L, S), dep(L', S))$ is a *transfer activity*, where S denotes a station where passengers can change between the lines L and L'.

We summarize trip, dwell, and turn activities under the term *vehicle activities* as they are performed by vehicles. In the following $X \subseteq A$ denotes the subset of vehicle activities.

The unit in which timetables are typically published are integer minutes. For internal use it may be also required to have a discretization of seconds. In either case, the arrival and departure times are conveniently integer values. Therefore, it is desirable to find an integer solution $\pi \in \{0, \ldots, T-1\}^V$ for the PESP.

Theorem 2.1 (Odijk, 1994) Every feasible instance (D, ℓ, u, T) of the PESP with ℓ and u integer-valued has a solution π that is integer.

On account of the above remarks, we will assume that every activity has a minimum and maximum allowable time duration that is integer. An instance of the PESP with ℓ and u integer-valued will be referred to as *integer PESP instance*.

Periodic interval constraints

Every timetable requirement discussed in Section 1.2.1 can be modeled by periodic interval constraints (2.1). These ensure that the timetable defines the timings of all events in such a way that for each activity $a = (v, w)$, the resulting time duration is contained in its time window $[\ell_a, u_a]$.

Let use investigate the application of periodic interval constraints by example. First, consider a trip activity $(dep(L, S), arr(L, S'))$, and suppose the minimum and maximum allowable time duration for the trip are given by $\ell_{trip(L,S,S')}$ and $u_{trip(L,S,S')}$, respectively. To model this requirement, we simply put the information into the general periodic interval constraint (2.1) to obtain

$$(\pi_{arr(L,S')} - \pi_{dep(L,S)} - \ell_{trip(L,S,S')}) \bmod T \leq u_{trip(L,S,S')} - \ell_{trip(L,S,S')}.$$

Imposing constrains on dwell activities works by the same method. Let the minimum dwell time of line L at station S be given by $\ell_{dwell(L,S)}$. If this value should not be exceeded by more than n minutes, we define a maximum allowable duration of $u_{dwell(L,S)} = \ell_{dwell(L,S)} + n$ to obtain

$$(\pi_{dep(L,S)} - \pi_{arr(L,S)} - \ell_{dwell(L,S)}) \bmod T \leq n.$$

Slightly more complicated is the modeling of safety and synchronization constraints in case we allow variable running times. However, we will not develop this point here. A more complete discussion and further examples can be found in Peeters (2003). See also Lindner (2000) and Kroon and Peeters (2003) who investigate the PESP with variable running times in detail.

Remark 2.2 We can certainly assume that $\ell_a \geq 0$ and $\ell_a \leq u_a$. Notice that $(\pi_w - \pi_v - \ell_a) \bmod T < T$. Therefore, every activity with $u_a - \ell_a \geq T$ imposes no condition on the timetable. From now on, we will therefore assume that $u_a - \ell_a < T$ for all activities $a \in A$. Furthermore, we need the assumption that $\ell_a < T$. This condition is not particular restrictive, due to the periodicity of the timetable. Indeed, if $\ell_a \geq T$, we compute z_a such that $\ell_a = Tz_a + (\ell_a \bmod T)$ and set $\ell'_a = \ell_a - Tz_a$ and $u'_a = u_a - Tz_a$. In some cases there is no explicit maximum allowable time duration for an activity a given. This can be handled by setting $u_a = \ell_a + T$.

The event-activity graph

It is possible to encode the set of periodic interval constraints imposed on the timetable in an *event-activity graph* $D = (V, A)$ with node set V and arc set A. The idea is to represent events by nodes, and activities by arcs in the graph. Every arc $a \in A$ is associated with the time window $[\ell_a, u_a]$ of its belonging activity. For convenience, we pass the naming of the activities on the arcs in the event-activity graph (i. e. trip arc, dwell arc etc). Depending on the context we refer to $v \in V$ as a node or an event. In the same manner we handle arcs and activities.

We have thus established the relation between the PERIODIC EVENT SCHEDULING PROBLEM and periodic timetabling. The event-activity graph $D = (V, A)$, the vectors $\ell, u \in \mathbb{Q}^A$, and the period time T together define a PESP instance. A solution $\pi \in [0, T)^V$ to this instance yields the periodic timetable.

Managing the modulo operator

Solving the PERIODIC EVENT SCHEDULING PROBLEM is rather difficult with the modulo operator as part of the constraints (2.1). Since we aim at applying integer programming techniques, we consider a reformulation of the problem that uses integer variables instead of the modulo operator to express the periodic interval constraints.

Proposition 2.3 (Serafini and Ukovich, 1989a) Suppose $\mathcal{I} = (D, \ell, u, T)$ is an instance of the PESP. Then, a vector π is a solution for $\mathcal{I}$ if and only if for every arc $a = (v, w) \in A$ there exists a unique integer $p_a \in \mathbb{Z}$ such that

$$\ell_a \leq \pi_w - \pi_v + T\, p_a \leq u_a. \tag{2.2}$$

Proof. The proof is based on Liebchen (2006). Recall the definition of the modulo operator. From this it follows that

$$(\pi_w - \pi_v - \ell_a) \bmod T = \pi_w - \pi_v - \ell_a - T \cdot \max\{z \in \mathbb{Z} \mid \pi_w - \pi_v - \ell_a - Tz \geq 0\}.$$

Rewriting inequality (2.2) to

$$0 \leq \pi_w - \pi_v - \ell_a + T\, p_a \leq u_a - \ell_a$$

we conclude that

$$p_a = -\max\{z \in \mathbb{Z} \mid \pi_w - \pi_v - \ell_a - Tz \geq 0\} \in \mathbb{Z},$$

where the uniqueness of p_a follows from our assumption that $u_a - \ell_a < T$. ■

The integer p_a is called *periodical offset* (or *modulo parameter*) of the activity a.

2.2.1 Complexity of the PESP

The complexity of the PESP has been extensively studied in the literature. Without loss of generality we announce the results for integer PESP instances only.

Theorem 2.4 The PESP is $\mathcal{NP}$-complete for fixed $T \geq 3$.

A first proof of the above theorem was given by Serafini and Ukovich (1989a). They used a reduction from the HAMILTONIAN CYCLE PROBLEM (HCP) which is $\mathcal{NP}$-complete. However, several authors proclaim that this proof is problematic, because the period time of the resulting PESP instance depends on the size of the HCP instance. In particular, if $G = (V, E)$ denotes the undirected graph of the HAMILTONIAN CYCLE PROBLEM, then $T = |V|$. Therefore, Nachtigall (1996a) provides an alternative proof that fixes this drawback. Further proofs can be found using reductions from the K-VERTEX COLORABILITY PROBLEM (Odijk, 1994), and the LINEAR ORDERING PROBLEM (Liebchen and Peeters, 2001).

It is worth pointing out that for $T = 2$, the PERIODIC EVENT SCHEDULING PROBLEM is polynomially solvable. A possible solution algorithm can be found in Peeters (2003).

2.2.2 Cost Optimization

The PERIODIC EVENT SCHEDULING PROBLEM is a feasibility problem. If a solution has been found, no information on its *quality* is available. In order to compute timetables that are provably better than those constructed "by hand", we introduce a cost optimization scheme for the PESP.

Every activity in the transportation network has a minimum time duration (see Section 2.2). Exceeding this time duration affects the quality of the timetable. In case of dwell and transfer activities, passenger may have to face longer travel times. For turnaround activities, the timetable may require additional vehicles. Therefore, we want to penalize the *slack time* $(\pi_w - \pi_v - \ell_a) \bmod T$ for every activity $a = (v, w) \in A$ with a non-negative weight $w_a \in \mathbb{Q}_{\geq 0}$. This weight represents the importance of the related activity. In case of transfer and dwell activities, the weight is typically chosen proportional to the number of affected passengers which can be obtained from the given passenger flow. If an activity a is not relevant for optimization, we set $w_a = 0$.

The objective is to minimize the weighted sum of the slack times. To this end, we consider the following linear objective function

$$\text{Minimize} \sum_{a=(v,w)\in A} w_a\,(\pi_w - \pi_v - \ell_a) \bmod T. \tag{2.3}$$

Note that we do not directly minimize the total travel time in the transportation network, but we can influence factors like transfer waiting time.

An instance of the PESP with an objective function as (2.3) is called PESP *optimization instance* and denoted by (D, ℓ, u, w, T).

2.2.3 Redundancy in the PESP Solution Space

The solution space of the PESP contains many solutions that correspond to the same periodic timetables. They can be transformed into each other by a *periodic shift*. This is stated in the following lemma.

Lemma 2.5 Let $\mathcal{I} = (D, \ell, u, w, T)$ denote an optimization instance of the PESP, and let $\pi \in [0, T)^V$ be an optimal solution to $\mathcal{I}$. Then for every $\beta \in [0, T)$, the vector $\overline{\pi} \in [0, T)^V$ defined by $\overline{\pi}_v = (\pi_v + \beta) \bmod T$ for $v \in V$ is optimal for the instance $\mathcal{I}$, too.

Proof. The proof is straightforward. We first show that the vector $\overline{\pi}$ satisfies the set of periodic interval constraints (2.1). To this end, let $a = (v, w) \in A$. Then,

$$\begin{aligned}(\overline{\pi}_w - \overline{\pi}_v - \ell_a) \bmod T &= ((\pi_w + \beta) \bmod T - (\pi_v + \beta) \bmod T - \ell_a) \bmod T\\ &= (\pi_w - \pi_v - \ell_a) \bmod T\\ &\leq u_a - \ell_a,\end{aligned}$$

where the last inequality holds, due to the feasibility of the vector π. Finally, we conclude from (2.3) that the vectors π and $\overline{\pi}$ have the same objective value, which proves the lemma. ■

The lemma shows that we can set $\pi_v = 0$ for an arbitrary $v \in V$ without "loosing" the optimal solution. Liebchen (2006) calls this the *scaling property*.

2.2.4 Integer Programming Formulations

In the following we consider optimization instances of the PESP. Recall that such instances are given by a directed graph $D = (V, A)$, vectors $\ell, u, w \in \mathbb{Q}^A$, and an integer T. For a better understanding, let us restate the meaning of the required symbols.

π_v	timing of event $v \in V$
ℓ_a	minimum allowable time duration of activity $a \in A$
u_a	maximum allowable time duration of activity $a \in A$
w_a	weight of activity $a \in A$
p_a	periodical offset of activity $a \in A$
T	period time

We can model the problem of finding an optimal timetable for the PESP instance (D, ℓ, u, w, T) as the mixed integer program

PESP *basic* (2.4)

$$\text{Minimize} \quad \sum_{a=(v,w)\in A} w_a \left(\pi_w - \pi_v + T\, p_a - \ell_a\right)$$

subject to

$$\begin{aligned}
\pi_w - \pi_v + T\, p_a &\geq \ell_a && \text{for all } a = (v,w) \in A && \text{(2.4a)}\\
\pi_w - \pi_v + T\, p_a &\leq u_a && \text{for all } a = (v,w) \in A && \text{(2.4b)}\\
0 \leq \pi_v &\leq T - 1 && \text{for all } v \in V && \text{(2.4c)}\\
\pi_v &\in \mathbb{Z} && \text{for all } v \in V && \text{(2.4d)}\\
p_a &\in \mathbb{Z} && \text{for all } a \in A. && \text{(2.4e)}
\end{aligned}$$

According to Proposition 2.3, we may identify a feasible solution for the instance (D, ℓ, u, w, T) with a point (π, p) in $\{0, \ldots, T-1\}^V \times \mathbb{Z}^A$ that satisfies the constraints (2.4a) and (2.4b). An optimal solution minimizes the objective function $\sum_{a=(v,w)\in A} w_a \left(\pi_w - \pi_v + T\, p_a - \ell_a\right)$ which is the sum of the weighted slack times over all activities $a \in A$ (see Section 2.2.2).

Remark 2.6 If ℓ and u are integer vectors, we can relax the integrality constraints on the π variables, if the periodical offsets p_a are still required to be integer (see Peeters (2003)).

Lemma 2.7 The optimal solution value to the LP relaxation of PESP *basic* (2.4) is zero.

Proof. For every $\pi \in \{0, \dots, T-1\}^V$, set $p_a = (-\pi_w + \pi_v + \ell_a)/T$. ■

Lemma 2.8 (Liebchen, 2006) In PESP *basic* (2.4) we can assume that $p_a \in [0, \overline{p}_a] \cap \mathbb{Z}$, where

$$\overline{p}_a = \begin{cases} 1, & \text{if } u_a \leq T, \\ 2, & \text{otherwise.} \end{cases} \tag{2.5}$$

This involves no loss of generality.

Proof. Let us restate the proof from Liebchen (2006). Let $a \in A$, and let $\pi \in \{0, \dots, T-1\}^V$ be a vector that satisfies the constraints (2.4a)–(2.4b) for activity a, i. e. $\ell_a \leq \pi_w - \pi_v + T\, p_a \leq u_a$. Depending on the value of p_a, we have

$$\pi_w - \pi_v + T\, p_a \in \begin{cases} \{-2T+1, \dots, \quad -1\}, & \text{if } p_a = -1, \\ \{\ -T+1, \dots, \ T-1\}, & \text{if } p_a = \ \ 0, \\ \{\qquad 1, \dots, 2T-1\}, & \text{if } p_a = \ \ 1, \\ \{\quad T+1, \dots, 3T-1\}, & \text{if } p_a = \ \ 2, \text{ and} \\ \{\ 2T+1, \dots, 4T-1\}, & \text{if } p_a = \ \ 3. \end{cases}$$

On account of Remark 2.2, we have $\ell_a \geq 0$. Thus, we can exclude $p_a \leq -1$. Further, we assumed $\ell_a < T$ and $u_a - \ell_a < T$, such that $u_a < 2T$. Thus, we cannot have $p_a \geq 3$. Finally, in the case $u_a \leq T$, we can even forbid $p_a = 2$. ■

The lemma shows that we can reformulate PESP *basic* (2.4) with box constraints on the periodical offsets. This gives

PESP *box* (2.6)

$$\text{Minimize} \quad \sum_{a=(v,w)\in A} w_a \left(\pi_w - \pi_v + T\, p_a - \ell_a\right)$$

subject to

$$\begin{aligned}
\pi_w - \pi_v + T\, p_a &\geq \ell_a && \text{for all } a = (v,w) \in A && \text{(2.4a)}\\
\pi_w - \pi_v + T\, p_a &\leq u_a && \text{for all } a = (v,w) \in A && \text{(2.4b)}\\
0 \leq \pi_v &\leq T - 1 && \text{for all } v \in V && \text{(2.4c)}\\
0 \leq p_a &\leq \overline{p}_a && \text{for all } a \in A && \text{(2.6a)}\\
\pi_v &\in \mathbb{Z} && \text{for all } v \in V && \text{(2.4d)}\\
p_a &\in \mathbb{Z} && \text{for all } a \in A. && \text{(2.4e)}
\end{aligned}$$

Notation 2.9 The convex hull of all solutions to PESP *box* (2.6) will be denoted by

$$\mathcal{P}_{\text{IP}}(\text{PESP } box) = \text{conv}\{(\pi, p) \in \mathbb{Q}^V \times \mathbb{Q}^A \mid (\pi, p) \text{ satisfies (2.4a)–(2.4e), and (2.6a)}\}$$

Moreover, we write

$$\mathcal{P}_{\text{LP}}(\text{PESP } box) = \{(\pi, p) \in \mathbb{Q}^V \times \mathbb{Q}^A \mid (\pi, p) \text{ satisfies (2.4a)–(2.4c), and (2.6a)}\}$$

for the rational polyhedron of the LP relaxation of PESP *box* (2.6).

Remark 2.10 It is even possible to transform a PESP instance such that the periodical offset variables may take only binary values, i. e., $p_a \in \{0, 1\}$ for every activity $a \in A$. An appropriate transformation was given by Peeters (2003). From the proof of Lemma 2.8 it follows that only activities with a maximum allowable time duration larger than T may cause non-binary periodical offsets, and therefore require special treatment. According to Peeters (2003), we can split every activity a with $u_a \geq T$ into two activities. The time windows of the new activities are so chosen that they are equivalent to that of activity a.

In case the maximum allowable time duration u_a of activity $a = (v, w)$ is strictly smaller than T, the periodical offset p_a indicates a sequence of the events v and

w in the interval $[0, T-1]$. To see this, consider the periodic interval constraint of activity a

$$\ell_a \leq \pi_w - \pi_v + T\,p_a \leq u_a.$$

According to Remark 2.2, we have $\ell_a \geq 0$, and thus $[\ell_a, u_a] \subseteq [0, T)$. Since $-T + 1 \leq \pi_w - \pi_v \leq T - 1$, the following can be concluded. If $\pi_v \leq \pi_w$, we have $p_a = 0$. Else, if $\pi_v > \pi_w$, then $p_a = 1$.

Lemma 2.11 **(Peeters, 2003)** Every PESP instance $\mathcal{I} = (D, \ell, u, T)$ with periodic interval constraints of the form (2.2) can be transformed such that $p_a \in \{0, 1\}$ for all $a \in A$.

Another interesting point is that we can predetermine the periodical offsets for certain activities of a feasible PESP instance $\mathcal{I} = (D, \ell, u, w, T)$. Let $F \subseteq A$ induce an arbitrary spanning tree of the underlying undirected graph $G = (V, A)$ of D. Let (π, p) be a feasible solution to $\mathcal{I}$, and let $p^* \in \mathbb{Z}^F$ be a fixed vector of periodical offsets (e. g. $p^* \equiv 0$). Then, there exists a feasible solution (π', p') such that for all $(v, w) \in V \times V$, we have $\pi_w - \pi_v \bmod T = \pi'_w - \pi'_v \bmod T$, and $p'_a = p^*_a$ for every activity $a \in F$. However, this fixation may require π variables to take values not contained in the set $\{0, \ldots, T-1\}$. More details can be found in Liebchen (2006). We obtain

PESP *tree* (2.7)

$$\text{Minimize} \quad \sum_{a=(v,w)\in A} w_a \left(\pi_w - \pi_v + T\,p_a - \ell_a\right)$$

subject to

$$\pi_w - \pi_v + T\,p_a \geq \ell_a \qquad \text{for all } a = (v,w) \in A \qquad (2.4a)$$

$$\pi_w - \pi_v + T\,p_a \leq u_a \qquad \text{for all } a = (v,w) \in A \qquad (2.4b)$$

$$\pi_v \in \mathbb{Z} \qquad \text{for all } v \in V \qquad (2.7a)$$

$$p_a \in \mathbb{Z} \qquad \text{for all } a \in A \backslash F \qquad (2.7b)$$

$$p_a = 0 \qquad \text{for all } a \in F. \qquad (2.7c)$$

In comparison with PESP *box* (2.6), we reduce the number of p variables from $|A|$ to $|A| - |V| + 1$ by fixing the periodical offsets on the spanning tree F. This, however, requires a larger range for the π variables. As a consequence, we need to compute every π_v in a solution to the above linear program modulo T to obtain a vector π that is feasible for the PESP instance.

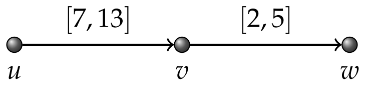


Figure 2.1 Illustration of Remark 2.12.

Remark 2.12 The PESP formulations we have seen so far are based on event timings. This makes the modeling easy, but it is difficult to define reasonable bounds on the variables. We illustrate this by the PESP instance shown in Figure 2.1. Suppose the period time $T = 10$, and consider the following two solutions:

a) $\pi_u = 0$, $\pi_v = 7$, and $\pi_w = 9$, and

b) $\pi_u = 4$, $\pi_v = 1$, and $\pi_w = 3$.

Both solutions are optimal, but the timings of the events are completely different. Hence, the values of the π variables are no direct indicator for the quality of the solution.

This motivates us to present another type of integer programming formulations for the PESP that use variables for the time durations of the activities. We are led to the theory of tensions.

Tensions

The vector $\pi \in [0, T)^V$ in the PESP description can be interpreted as a *potential* of the event-activity graph $D = (V, A)$. Then, a vector $y \in \mathbb{Z}^A$ is called *tension* if $y(\pi)_a = \pi_w - \pi_v$ for every activity $a = (v, w) \in A$. Our goal is to develop a PESP formulation based on tensions. To this end, we replace any occurrence of $\pi_w - \pi_v$ in PESP *box* (2.6) by the tension variable y_a of activity $a = (v, w)$. This gives $\ell_a \leq y_a + T\,p_a \leq u_a$ for the periodic interval constraint of activity a and $w_a\,(y_a + T\,p_a - \ell_a)$ for the weighted slack time. The transformation, however, is only allowed if we ensure that the vector $y \in \mathbb{Z}^A$ defines a tension. Theoretically, we have to impose $|A|$ constraints of the form $y_a = \pi_w - \pi_v$, one for every activity $a \in A$. However, by utilizing cycle bases of directed graphs, we show how to manage with $|A| - |V| + 1$ constraints and without the π variables. In what follows, we consider *oriented cycles*, i. e. cycles that have an orientation and that may contain forward and backward directed arcs. Let us introduce the notation C^+ (C^-) for the forward (backward) directed arcs of the oriented cycle C. Then, the incidence vector $\gamma_C \in \{-1, 0, 1\}^A$ of the cycle C is given by

$$\gamma_{C_a} = \begin{cases} 1, & \text{if } a \in C^+, \\ -1, & \text{if } a \in C^-, \text{ and} \\ 0, & \text{if } a \notin C. \end{cases}$$

If $D = (V, A)$ is a directed graph, we call

$$\mathcal{C}(D) = \text{span}(\{\gamma_C \mid C \text{ oriented cycle in } D\}) \subset \mathbb{Q}^A \tag{2.8}$$

the *cycle space* of D. A set $C_1, \ldots, C_k$ of oriented cycles is called a *cycle basis of the directed graph D*, if the following conditions hold: The vectors $\gamma_{C_1}, \ldots, \gamma_{C_k}$ are linearly independent and $\text{span}(\{\gamma_{C_1}, \ldots, \gamma_{C_k}\}) = \mathcal{C}$.

The following theorem provides the conditions by which we can ensure that a vector defines a tension for a directed graph.

Theorem 2.13 **(Bollobás, 1998)** Let $D = (V, A)$ be a directed graph, and let $y \in \mathbb{Q}^A$. Then, the following conditions are equivalent:

1. y is a tension,
2. for every oriented cycle C there holds $\sum_{a \in C^+} y_a - \sum_{a \in C^-} y_a = 0$, and
3. for every cycle basis $\mathcal{B} = \{C_1, \ldots, C_\nu\}$ we have $\sum_{a \in C_i^+} y_a - \sum_{a \in C_i^-} y_a = 0$ along every cycle $C_i \in \mathcal{B}$.

The theorem states that it suffices to introduce constraints for the cycles in a cycle basis in order to enforce a tension on the arcs of the event-activity graph. Thus, only $|A| - |V| + 1$ constraints are required which is the dimension of the cycle space of a directed graph (e. g. cf. Liebchen, 2006).

Suppose $\mathcal{I} = (D, \ell, u, w, T)$ denotes a PESP optimization instance and $\mathcal{B}$ is a cycle basis of the event-activity graph D. We obtain

PESP *y-box* (2.9)

$$\text{Minimize} \quad \sum_{a \in A} w_a \left(y_a + T\, p_a - \ell_a \right)$$

subject to

$$\begin{aligned}
y_a + T\, p_a &\geq \ell_a && \text{for all } a \in A && (2.9a)\\
y_a + T\, p_a &\leq u_a && \text{for all } a \in A && (2.9b)\\
\sum_{a \in C^+} y_a - \sum_{a \in C^-} y_a &= 0 && \text{for all } C \in \mathcal{B} && (2.9c)\\
0 \leq p_a &\leq \overline{p}_a && \text{for all } a \in A && (2.6a)\\
-T + 1 \leq y_a &\leq T - 1 && \text{for all } a \in A && (2.9d)\\
p_a &\in \mathbb{Z} && \text{for all } a \in A && (2.4e)\\
y_a &\in \mathbb{Z} && \text{for all } a \in A. && (2.9e)
\end{aligned}$$

For the transformation between tensions and potentials we refer the reader to Liebchen (2006).

Periodic Tensions

We now develop integer programming formulations for the PESP that do not require the periodical offset variables p_a. This can be achieved by using a special type of tensions.

Let $\pi \in [0, T)^V$ denote a potential of the event-activity graph $D = (V, A)$. Then, a vector $x \in \mathbb{Q}^A$ is a *periodic tension*, if there exists a vector $p \in \mathbb{Z}^A$ such that $x(\pi)_a = \pi_w - \pi_v + T\, p_a$ for every activity $a = (v, w) \in A$. Note that the periodic tension x_a is exactly the time duration of the activity a.

As in the previous section, we cannot simply replace the term $\pi_w - \pi_v + T\, p_a$ in PESP *box* (2.6) by the variable x_a to obtain a PESP formulation based on periodic tensions. We need to add further constraints which require the following definition.

Definition 2.14 Let $D = (V, A)$ be a directed graph, and let C be an oriented cycle of D. A vector $x \in \mathbb{Z}^A$ satisfies the *cycle periodicity property* for C with period time T, if there exists an integer z_C such that

$$\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T\, z_C. \tag{2.10}$$

Theorem 2.15 (Peeters, 2003) Let $D = (V, A)$ be a directed graph. A vector $x \in \mathbb{Z}^A$ is a periodic tension if and only if, for each cycle $C \in D$, it satisfies the cycle periodicity property for period time T

$$\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T\, z_C.$$

Again, we can reduce the number of required constraints with the help of cycle bases, in particular, integral cycle bases.

Definition 2.16 A cycle basis $\mathcal{B}$ of a directed graph D is called *integral cycle basis* if every non-basis cycle of D is an integer linear combination of the cycles in $\mathcal{B}$.

The next theorem states that it suffices to require the cycle periodicity property for an integral cycle basis of the event-activity graph to obtain a periodic tension.

Theorem 2.17 (Peeters, 2003) Let $D = (V, A)$ be a directed graph, and let $x \in \mathbb{Z}^A$. If x satisfies the cycle periodicity property $\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T\, z_C$ for every cycle C in an integral cycle basis $\mathcal{B}$ of D, then it does for all cycles of D.

Let (D, ℓ, u, w, T) denote a PESP instance, and suppose $\mathcal{B}$ is an integral cycle basis of the graph D. Then, we get

PESP x (2.11)

$$\begin{aligned}
&\text{Minimize} \quad \sum_{a \in A} w_a \left(x_a - \ell_a\right) \\
&\quad \text{subject to} \\
&& x_a \geq \ell_a && \text{for all } a \in A && \text{(2.11a)} \\
&& x_a \leq u_a && \text{for all } a \in A && \text{(2.11b)} \\
&& \sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a - T z_C = 0 && \text{for all } C \in \mathcal{B} && \text{(2.11c)} \\
&& x_a \in \mathbb{Z} && \text{for all } a \in A && \text{(2.11d)} \\
&& z_C \in \mathbb{Z} && \text{for all } C \in D && \text{(2.11e)}
\end{aligned}$$

The last IP formulation is obtained by replacing the periodic tension variables x_a with the periodic slack variables $\tilde{x}_a = x_a - \ell_a$. According to Remark 2.2 and the periodic interval constraints, we may assume that $\tilde{x}_a \in [0, T)$. This gives

PESP $\tilde{x}$ (2.12)

$$\begin{aligned}
&\text{Minimize} \quad \sum_{a \in A} w_a \tilde{x}_a \\
&\quad \text{subject to} \\
&& \tilde{x}_a \leq u_a - \ell_a && \text{for all } a \in A && \text{(2.12a)} \\
&& \sum_{a \in C^+} \tilde{x}_a - \sum_{a \in C^-} \tilde{x}_a - T z_C = - \sum_{a \in C^+} \ell_a + \sum_{a \in C^-} \ell_a && \text{for all } C \in \mathcal{B} && \text{(2.12b)} \\
&& 0 \leq \tilde{x}_a \leq T - 1 && \text{for all } a \in A && \text{(2.12c)} \\
&& \tilde{x}_a \in \mathbb{Z} && \text{for all } a \in A && \text{(2.12d)} \\
&& z_C \in \mathbb{Z} && \text{for all } C \in \mathcal{B}. && \text{(2.12e)}
\end{aligned}$$

In principle, we may choose an arbitrary integral cycle basis for the formulations PESP x (2.11) and PESP $\tilde{x}$ (2.12). Certain cycle bases, however, yield stricter cycle inequalities and thus better bounds on the z variables. How to determine such cycle bases is extensively studied in Liebchen (2006).

Finally, note that also the PESP formulations on periodic tensions have poor LP relaxations.

Lemma 2.18 The optimal solution value to the LP relaxation of PESP $\tilde{x}$ (2.12) is zero.

Proof. Set $\tilde{x}_a = 0$ for every activity $a \in A$, and choose

$$z_C = (\sum_{a \in C^+} \ell_a - \sum_{a \in C^-} \ell_a)/T$$

for every cycle $C \in \mathcal{B}$. ■

Valid inequalities for the PESP formulations

There has been enormous effort to develop valid inequalities for the mixed integer programs of the PESP. Interesting results can be found in, for example, Odijk (1994), Nachtigall (1998), and Lindner (2000). A detailed discussion of the theory is outside the scope of this thesis, but let us shortly investigate the cycle inequalities proposed by Odijk (1994).

Theorem 2.19 (Odijk, 1994) An instance (D, ℓ, u, T) of the PESP is feasible if and only if there exists an integer vector p such that for every cycle C of D,

$$\underline{p}_C \leq \sum_{a \in C^+} p_a - \sum_{a \in C^-} p_a \leq \overline{p}_C, \tag{2.13}$$

where

$$\underline{p}_C = \left\lceil \frac{1}{T} \left(\sum_{a \in C^+} \ell_a - \sum_{a \in C^-} u_a \right) \right\rceil, \text{ and } \overline{p}_C = \left\lfloor \frac{1}{T} \left(\sum_{a \in C^+} u_a - \sum_{a \in C^-} \ell_a \right) \right\rfloor.$$

Remark 2.20 The cycle inequalities (2.13) can be used to obtain bounds on the integer variables z_C in PESP x (2.11) and PESP $\tilde{x}$ (2.12). Indeed, the combination of equalities (2.9c) and (2.11c) for a cycle C of the same cycle basis of the graph D gives

$$z_C = \frac{1}{T} \left(\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a \right) = \sum_{a \in C^+} p_a - \sum_{a \in C^-} p_a,$$

and from inequality (2.13) we can conclude

$$\left\lceil \frac{1}{T}\left(\sum_{a\in C^+} \ell_a - \sum_{a\in C^-} u_a \right) \right\rceil \leq z_C \leq \left\lfloor \frac{1}{T}\left(\sum_{a\in C^+} u_a - \sum_{a\in C^-} \ell_a \right) \right\rfloor . \tag{2.14}$$

Example 2.21 In Figure 2.2 we give a small example for Remark 2.20. Let C denote the displayed cycle, and suppose $T = 10$. Then, the cycle inequalities (2.14) yield

$$\left\lceil \frac{1}{10}(4+3-7) \right\rceil \leq z_C \leq \left\lfloor \frac{1}{10}(7+6-2) \right\rfloor ,$$

and hence, $0 \leq z_C \leq 1$.

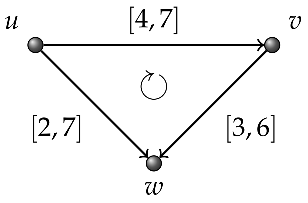


Figure 2.2 Illustration of Example 2.21.

2.2.5 Further Solution Approaches

A straightforward approach is to solve the PESP by integer programming techniques. However, there may be situations in which other solution methods are more appropriate. On the one hand, the mixed integer programs resulting from real-world timetabling instances are likely hard to solve. From this point of view, heuristic approaches as the *Modulo Network Simplex Method* proposed by Nachtigall and Opitz (2007) gain in interest. On the other hand, a PESP instance may be infeasible if there are conflicting timetable requirements. Hence, we need to know which of the initial constraints have to be relaxed. Since this information cannot be provided by a standard MIP solver, special algorithms are needed. A good overview of this topic is provided by Lindner (2000).

2.3 The Quadratic Semi-Assignment Problem

In this section we discuss the QUADRATIC SEMI-ASSIGNMENT PROBLEM (QSAP), a quadratic model for periodic timetabling. Several researchers have studied exact and heuristic solution approaches for the QSAP, including Domschke (1989), Klemt and Stemme (1987), and Daduna and Voß (1993). Our purpose is to investigate mixed integer programming formulations.

The main focus of the model is the minimization of transfer waiting times in the transportation network. Like the PESP, the QSAP supports individual line frequencies and we can easily incorporate timetables of superior transportation systems. However, there are some modeling limitations. The QSAP requires fixed running times for the lines and does not allow to exceed the minimum dwell times. Thus, the departure time of a line at its initial station is the only degree of freedom.

Before stating the model, let us first outline its idea. We determine the timetable by selecting a departure time for every directed line $i \in \mathcal{L}$ within its period time T_i, i. e. from the set $\{0, \ldots, T_i - 1\}$. By our assumptions, the departure times of the lines at their initial stations already define the arrival times at the transfer stations. Therefore, we are able precompute the transfer waiting times that arise for transfers between two lines based on their initial departure times. This is the basic concept behind the minimization of transfer waiting time in the QSAP.

In the model description we use the following parameters.

$\mathcal{L}$ set of (directed) lines

T_i period time of line $i \in \mathcal{L}$; to shorten notation, we sometimes let $\mathcal{T}_i$ stand for the set $\{0, \ldots, T_i - 1\}$ of possible departure times of line i at its initial station

w_{ijkl} total waiting time that arises for transfers from line i to line k, or vice versa, if line i (k) leaves its initial station at time $j \in \mathcal{T}_i$ ($l \in \mathcal{T}_k$)

c_{ijkl} product of w_{ijkl} and the number of passengers who change between the lines i and j, and vice versa

x_{ij} binary variable whose value is 1 if line $i \in \mathcal{L}$ is assigned the departure time $j \in \mathcal{T}_i$ at its initial station, and 0 otherwise

Quadratic Semi-Assignment Problem (QSAP)

Instance: Set $\mathcal{L} = \{1, \dots, n\}$, positive integers T_i ($i \in \mathcal{L}$), and cost coefficients c_{ijkl} ($i, k \in \mathcal{L} : i < k;\ j \in \mathcal{T}_i;\ l \in \mathcal{T}_k$).

Question: Is there a vector x with components $x_{ij} \in \{0, 1\}$ ($i \in \mathcal{L};\ j \in \mathcal{T}_i$) such that $\sum_{j=0}^{T_i-1} x_{ij} = 1$ for all $i \in \mathcal{L}$ and which minimizes

$$Z(x) = \sum_{i=1}^{n} \sum_{j=0}^{T_i-1} \sum_{k=i+1}^{n} \sum_{l=0}^{T_k-1} c_{ijkl}\, x_{ij}\, x_{kl} \quad ?$$

The problem is $\mathcal{NP}$ hard (Sahni and Gonzalez, 1976). For every line $i \in \mathcal{L}$, the constraint $\sum_{j=0}^{T_i-1} x_{ij} = 1$ requires to select a departure time for line i at its initial station. If line i departs at time $j \in \mathcal{T}_i$ and line k departs at time $l \in \mathcal{T}_k$, then the product $x_{ij}\, x_{kl}$ is 1 which gives a weighted transfer waiting time of c_{ijkl}. An optimal solution corresponds to an assignment x that minimizes the total weighted transfer waiting time $Z(x)$.

It is also possible to consider an alternative objective function to prevent the passengers from extreme transfer waiting times,

$$\begin{aligned} Z(x, \lambda) = \lambda \sum_{i=1}^{n} \sum_{j=0}^{T_i-1} \sum_{k=i+1}^{n} \sum_{l=0}^{T_k-1} & c_{ijkl}\, x_{ij}\, x_{kl} \\ & + (1 - \lambda) \max\{c_{ijkl}\, x_{ij}\, x_{kl} \mid i, k \in \mathcal{L},\ j \in \mathcal{T}_i,\ l \in T_k\}, \end{aligned}$$

where $\lambda \in [0, 1]$ is a respective weight parameter.

A graph theoretic interpretation

The above given interpretation of the QSAP for periodic timetabling is not the only possible. Klemt and Stemme (1987) introduce the problem on the basis of the *transfer graph*.

This graph has the following structure. For each line $i \in \mathcal{L}$, the transfer graph contains a node set V_i of dimension T_i. Each node $v \in V_i$ represents a possible departure time of line i at its initial station within the period time T_i. There are no arcs between nodes in the set V_i, and the node sets V_i and V_j are linked if there is a transfer relation between the lines i and j. The number of arcs between the node sets V_i and V_j equals the product of T_i and T_j, while a node in the set V_i has T_j outgoing arcs with an end node in the set V_j. Precisely one arc joins a node in the set V_i with a node in the set V_j. Each arc in the transfer graph is associated with an arc weight equal to c_{ijkl}.

In this interpretation a permissible timetable corresponds to subgraph of the transfer graph that contains exactly one node from each of the V_i and the arcs between them. The cost of the timetable is then given by the sum of the arc weights of the subgraph.

Solving the periodic timetabling problem is thus equal to the task of determining such a subgraph while minimizing the total cost.

2.3.1 Integer Programming Formulations

In this section we have compiled some integer programming formulations of the QUADRATIC SEMI-ASSIGNMENT PROBLEM. We begin with a standard quadratic formulation and derive two possible linearizations of it.

A quadratic integer programming formulation

The QUADRATIC SEMI-ASSIGNMENT PROBLEM directly translates into the following quadratic integer programming formulation.

QSAP (2.15)

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=0}^{T_i-1} \sum_{k=i+1}^{n} \sum_{l=0}^{T_k-1} c_{ijkl}\, x_{ij}\, x_{kl}$$

subject to

$$\sum_{j=0}^{T_i-1} x_{ij} = 1 \qquad \text{for all } i \in \mathcal{L} \tag{2.15a}$$

$$0 \leq x_{ij} \leq 1 \qquad \text{for all } i \in \mathcal{L},\ j \in \mathcal{T}_i \tag{2.15b}$$

$$x_{ij} \in \mathbb{Z} \qquad \text{for all } i \in \mathcal{L},\ j \in \mathcal{T}_i \tag{2.15c}$$

Since the objective function of QSAP (2.15) is quadratic and in general nonconvex, a direct solution of this formulation is evidently hard. Due to this issue, we will next review possible linearizations of the problem.

Standard linearization of the Quadratic Semi-assignment Problem

We first demonstrate how to obtain a mixed integer linear program from the quadratic formulation QSAP (2.15) using the standard textbook linearization due to Fortet (1960).

Without loss of generality we can assume that the products $x_{ij}\,x_{kl}$ are ordered such that $i < k$. For each such product $x_{ij}\,x_{kl}$, we introduce a non-negative variable $y_{ijkl} \in \mathbb{Q}_{\geq 0}$ and extend the formulation by the following linearization constraints

$$y_{ijkl} \leq x_{ij} \tag{2.16}$$
$$y_{ijkl} \leq x_{kl} \tag{2.17}$$
$$y_{ijkl} \geq x_{ij} + x_{kl} - 1, \tag{2.18}$$

where $i,k \in \mathcal{L} : i < k,\ j \in \mathcal{T}_i,\ l \in \mathcal{T}_k$. It is easy to check that $y_{ijkl} = 1$ if and only if $x_{ij}\,x_{kl} = 1$. If either x_{ij} or x_{kl} is zero, then by the non-negativity of y_{ijkl} and inequality (2.16) or (2.17), we have $y_{ijkl} = 0$. If $x_{ij} = x_{kl} = 1$, then the constraints (2.18) and (2.16) imply $y_{ijkl} = 1$.

In our case, we can omit the constraints (2.16) and (2.17) since all cost coefficients c_{ijkl} are non-negative. It follows that the resulting linear QSAP formulation requires $\sum_{i=1}^{n} \sum_{k=i+1}^{n} T_i\,T_k$ additional variables and the same number of additional constraints. We get

QSAP *basic* (2.19)

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=0}^{T_i-1} \sum_{k=i+1}^{n} \sum_{l=0}^{T_k-1} c_{ijkl}\, y_{ijkl}$$

subject to

$$\sum_{j=0}^{T_i-1} x_{ij} = 1 \qquad \text{for all } i \in \mathcal{L} \tag{2.15a}$$
$$x_{ij} + x_{kl} - y_{ijkl} \leq 1 \qquad \text{for all } i,k \in \mathcal{L} : i < k,\ j \in \mathcal{T}_i,\ l \in \mathcal{T}_k \tag{2.19a}$$
$$0 \leq x_{ij} \leq 1 \qquad \text{for all } i \in \mathcal{L},\ j \in \mathcal{T}_i \tag{2.15b}$$
$$x_{ij} \in \mathbb{Z} \qquad \text{for all } i \in \mathcal{L},\ j \in \mathcal{T}_i \tag{2.15c}$$
$$y_{ijkl} \in \mathbb{Q}_{\geq 0} \qquad \text{for all } i,k \in \mathcal{L} : i < k,\ j \in \mathcal{T}_i,\ l \in \mathcal{T}_k \tag{2.19b}$$

Notation 2.22 Let us introduce the sets $I = \{(i,j) \mid i \in \mathcal{L}, j \in \mathcal{T}_i\}$ and $J = \{(i,j,k,l) \in I \times I \mid i < k\}$. Then, we write

$$\mathcal{P}_{\text{IP}}(\text{QSAP } \textit{basic}) = \text{conv}\{(x,y) \in \mathbb{Q}^I \times \mathbb{Q}^J \mid (x,y) \text{ satisfies } (2.15\text{a})\text{–}(2.15\text{c}), \\ (2.19\text{a})\text{ , and } (2.19\text{b})\},$$

for the convex hull of the solutions to QSAP *basic* (2.19), and we denote by

$$\mathcal{P}_{\text{LP}}(\text{QSAP } \textit{basic}) = \{(x,y) \in \mathbb{Q}^I \times \mathbb{Q}^J \mid (x,y) \text{ satisfies } (2.15\text{a}),(2.15\text{b}), \\ (2.19\text{a}), \text{ and } (2.19\text{b})\}$$

the rational polyhedron of the LP relaxation to QSAP *basic* (2.19).

A downside of this formulation is that the newly introduced linearization constraints (2.19a) are of Big-M type and typically lead to a poor LP relaxation bound. We therefore provide another linearization of QSAP (2.15) on the same new variables y_{ijkl}. This formulation circumvents the Big-M difficulties of the standard linearization and manages with less additional constraints.

Compact linearization of the Quadratic Semi-assignment Problem

The next mixed integer program for the QSAP is based on a general linearization technique for binary quadratic programs presented in Liberti (2007). The method can be summarized as follows. We multiply the semi-assignment constraints under (2.15a) by each problem variable which yields further quadratic constraints of the form

$$\sum_{j=0}^{T_i-1} x_{ij}\, x_{kl} = x_{kl} \qquad \text{for } i,k \in \mathcal{L},\ l \in \mathcal{T}_k.$$

As before, we replace each product $x_{ij}\, x_{kl}$ in the formulation by the belonging variable y_{ijkl}. Note that y_{ijkl} and y_{klij} with $i < k$ stand for the same variable. This gives

QSAP *plus* (2.20)

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=0}^{T_i-1} \sum_{k=i+1}^{n} \sum_{l=0}^{T_k-1} c_{ijkl}\, y_{ijkl}$$

subject to

$$\sum_{j=0}^{T_i-1} x_{ij} = 1 \qquad \text{for all } i \in \mathcal{L} \qquad (2.15a)$$

$$\sum_{j=0}^{T_i-1} y_{ijkl} - x_{kl} = 0 \qquad \text{for all } i,k \in \mathcal{L} : i \neq k,\ l \in \mathcal{T}_k \qquad (2.20a)$$

$$0 \leq x_{ij} \leq 1 \qquad \text{for all } i \in \mathcal{L},\ j \in \mathcal{T}_i \qquad (2.15b)$$

$$x_{ij} \in \mathbb{Z} \qquad \text{for all } i \in \mathcal{L},\ j \in \mathcal{T}_i \qquad (2.15c)$$

$$y_{ijkl} \in \mathbb{Q}_{\geq 0} \qquad \text{for all } i,k \in \mathcal{L} : i < k,\ j \in \mathcal{T}_i,\ l \in \mathcal{T}_k. \qquad (2.19b)$$

Notation 2.23 Recall the definitions of the sets I and J from QSAP *basic* (2.19). Then, we introduce the following notation.

$$\mathcal{P}_{\text{IP}}(\text{QSAP } \textit{plus}) = \text{conv}\{(x,y) \in \mathbb{Q}^I \times \mathbb{Q}^J \mid (x,y) \text{ satisfies (2.15a)–(2.15c), (2.20a), and (2.19b)}\}$$

$$\mathcal{P}_{\text{LP}}(\text{QSAP } \textit{plus}) = \{(x,y) \in \mathbb{Q}^I \times \mathbb{Q}^J \mid (x,y) \text{ satisfies (2.15a),(2.15b), (2.20a), and (2.19b)}\}$$

The above formulation has $\sum_i \sum_{k \neq i} T_k$ additional constraints with respect to the formulation QSAP (2.15). We wish to compare this number with that of the standard linearization. For convenience we make the analysis for $T = \max_{i \in \mathcal{L}} T_i$. It follows that the standard linearization adds $\mathcal{O}(T^2)$ constraints to the formulation. For the compact linearization, on the other hand, only $\mathcal{O}(T)$ further constraints are required.

That QSAP *plus* (2.20) is indeed a valid linearization of QSAP (2.15), shows the following lemma.

Lemma 2.24 If $(x, y) \in \mathcal{P}_{\mathrm{IP}}(\mathrm{QSAP}\ \textit{plus})$, then

$$y_{ijkl} = 1 \Leftrightarrow x_{ij}\, x_{kl} = 1,$$

for all $i, k \in \mathcal{L} : i < k,\ j \in \mathcal{T}_i,\ l \in \mathcal{T}_k$.

Proof. We show that the constraints (2.20a) imply the standard linearization constraints (2.16)–(2.18). Recall that y_{ijkl} and y_{klij} for $i < k$ denote the same variable. From constraints (2.20a) it then follows immediately that $y_{ijkl} \leq x_{ij}$ and $y_{ijkl} \leq x_{kl}$. Thus, the only point remaining concerns the verification of inequality (2.18).

We have $y_{ijkl} \leq x_{ij}$, which gives

$$\sum_{m \neq j} y_{imkl} \leq \sum_{m \neq j} x_{im}.$$

Adding zero to the left-hand side, yields

$$\sum_m y_{imkl} - y_{ijkl} - \sum_{m \neq j} x_{im} \leq 0,$$

and according to the constraints (2.20a) we can replace $\sum_m y_{imkl}$ by x_{kl} to obtain

$$x_{kl} - y_{ijkl} - \sum_{m \neq j} x_{im} \leq 0.$$

Doing the zero-trick for the right-hand side as well, we get

$$x_{kl} - y_{ijkl} - \sum_m x_{im} - x_{ij} \leq 0.$$

But $\sum_m x_{im} = 1$ by constraints (2.15a), and hence,

$$x_{kl} - y_{ijkl} - 1 - x_{ij} \leq 0.$$

■

The next corollary shows that the LP relaxation of QSAP *plus* (2.20) is at least as tight as that of QSAP *basic* (2.19).

Corollary 2.25

$$\mathcal{P}_{\mathrm{LP}}(\text{QSAP } \textit{plus}) \subseteq \mathcal{P}_{\mathrm{LP}}(\text{QSAP } \textit{basic}).$$

Proof. Notice that the proof of Lemma 2.3.1 does not require the variables x_{ij} to be binary. Thus, by the same reasoning we can see that every point in the polyhedron $\mathcal{P}_{\mathrm{LP}}$(QSAP *plus*) is also contained in the polyhedron $\mathcal{P}_{\mathrm{LP}}$(QSAP *basic*). This establishes the corollary. ■

Finally, we have to note that the formulation QSAP (2.15) and, in particular, its linearizations QSAP *basic* (2.19) and QSAP *plus* (2.20) are, contrary to the PESP, very sensitive against further discretization of the period time. Every refinement would yield a drastic increase of the problem size.

3

Time Discretization Applied to PESP

The theme of this chapter is to develop a set of new integer programming formulations for the PERIODIC EVENT SCHEDULING PROBLEM that arise from time discretization. The basic idea is to consider an expanded event-activity graph which encodes the set of feasible timetables.

A similar approach has successfully been applied by Caprara *et al.* (2000) for a special scenario. They aim at computing periodic timetables for a set of railway lines that are operated on parts of the same route between two major stations. In order that the trains does not violate existing track capacities, minimum headway times are required. Their objective is to minimize waiting time caused by overtaking. Caprara *et al.* (2000) define a time-expanded graph as the basis for their IP formulations. In this graph a node models the arrival/departure at a station at a certain point in time.

We extend this method to general transportation networks. Section 3.1 introduces the expanded event-activity graph from which we derive the new PESP formulations in Section 3.2. The chapter closes with a discussion on speedup techniques that can be applied when solving the formulations by mixed integer programming.

3.1 *Expanding the PESP Event-Activity Graph*

We have defined the PERIODIC EVENT SCHEDULING PROBLEM on the basis of events (departures or arrivals) and activities in the transportation network. From this point of view, a feasible timetable is obtained by determining the timings of all events such that a given set of periodic interval constraints is satisfied.

We now describe how to encode the set of feasible solutions of a PESP instance in an *expanded event-activity graph*. In this graph a node represents an event at a point in time. The problem of finding a feasible timetable is then reduced to determine a special subgraph of this expanded event-activity graph.

Suppose $\mathcal{I} = (D, \ell, u, w, T)$ is an optimization instance of PESP. Recall that the instance $\mathcal{I}$ comprises the following components.

D	event-activity graph, where $D = (V, A)$
ℓ	minimum allowable time durations of the activities
u	maximum allowable time durations of the activities
w	weights of the activities
T	period time

Let us denote by $\boldsymbol{D} = (\boldsymbol{V}, \boldsymbol{A})$ the expanded event-activity graph, where $\boldsymbol{V}$ and $\boldsymbol{A}$ stand for the node set and the arc set of $\boldsymbol{D}$, respectively. For every event $v \in V$ and point in time $t \in \{0, \dots, T-1\}$, the expanded event-activity graph $\boldsymbol{D}$ contains a node $v[t] \in \boldsymbol{V}$. From this we obtain the node set of $\boldsymbol{D}$ which thus contains a total of $|\boldsymbol{V}| = |V| \cdot T$ nodes. In the following, we will write $\boldsymbol{V}(v)$ for the subset of $\boldsymbol{V}$ belonging to the event v.

Every activity is modeled by a set of arcs in the expanded event-activity graph. In fact, if $a = (v, w)$ is an activity in the event-activity graph $D = (V, A)$ of a PESP instance and has the time window $[\ell_a, u_a]$, then it is represented by the arc set

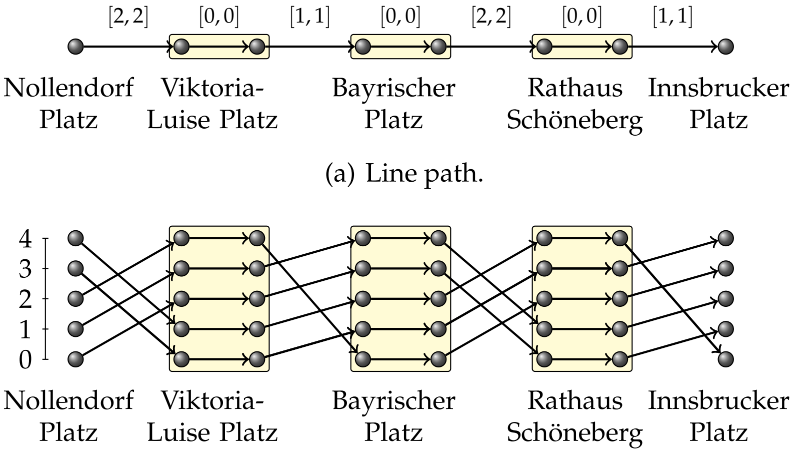
$$\boldsymbol{A}(a) = \{(v[t], w[t']) \in \boldsymbol{V} \times \boldsymbol{V} \mid t, t' \in \{0, \dots, T-1\},\ (t' - t - \ell_a) \bmod T \leq u_a - \ell_a\}.$$

For activities $a \in A$ with a fixed time duration this yields a set of T arcs. Notice that every arc in the set $\boldsymbol{A}(a)$ implies a feasible timing for the events related to activity a. The union of $\boldsymbol{A}(a)$ over all activities a in A gives the arc set $\boldsymbol{A}$ of the expanded event-activity graph. We associate with each arc $\boldsymbol{a} = (v[t], w[t'])$ in $\boldsymbol{A}$ costs $c_{\boldsymbol{a}}$ which are defined as $c_{\boldsymbol{a}} = w_a\,(t' - t - \ell_a) \bmod T$. Recall from Section 2.2.2 that this is precisely the slack time of the activity $a \in A$ implied by the arc $\boldsymbol{a} \in \boldsymbol{A}(a)$.

A feasible solution to the given PESP instance $\mathcal{I}$ corresponds to a subgraph $\boldsymbol{D'} = (\boldsymbol{V'}, \boldsymbol{A'})$ of the expanded event-activity graph $\boldsymbol{D}$ with the following properties.

1. For every activity $a \in A$, the set $\boldsymbol{A'}$ contains precisely one arc from the set $\boldsymbol{A}(a)$.
2. For every event $v \in V$, the set $\boldsymbol{V'}$ contains precisely one node from the set $\boldsymbol{V}(v)$.

The subgraph $\boldsymbol{D'}$ defines an optimal solution if, in addition, the sum of the weights of the arcs in the set $\boldsymbol{A'}$ is minimal.



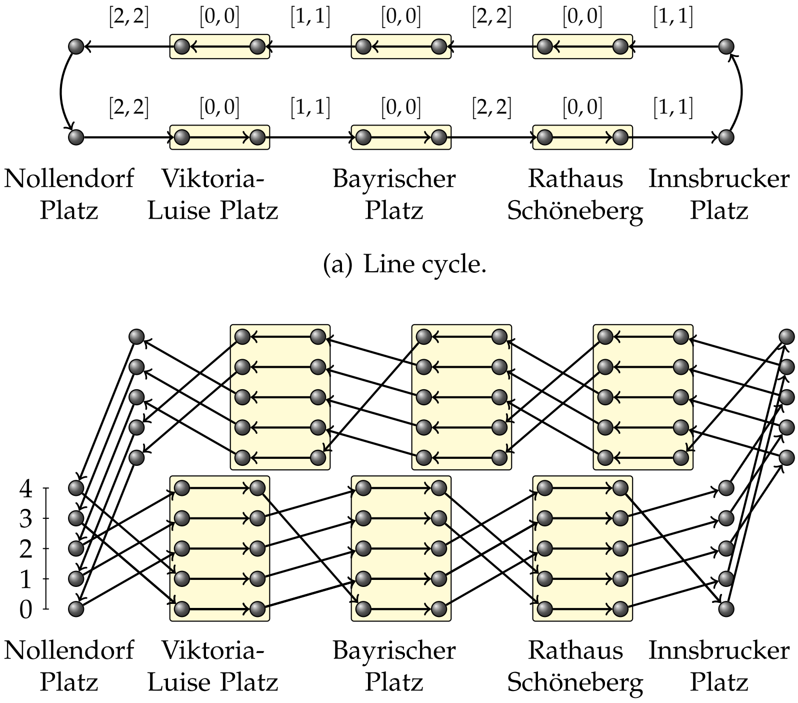
(a) Line path.

(b) Expanded line path.

Figure 3.1 A line path in the event-activity graph and the corresponding expanded line path in the expanded event-activity graph ($T = 5$). The arc labels denote the time windows of the related activities.

In the remainder of this chapter we will need the notion of a *line cycle* in the event-activity graph. Consider a line $R \in \mathcal{R}$ with forward direction L and backward direction $\overline{L}$. Taking each direction of line R on its own, the trip and dwell arcs of both lines define separate *line paths* in the event-activity graph (see Figure 3.1(a)). Each path starts and ends with a trip arc, and in between trip and dwell arcs alternate. At the end points the line paths of L and $\overline{L}$ are connected through turnaround arcs which form the line cycle of line R (see Figure 3.2(a)).

Line path and line cycle have an equivalent in the expanded event-activity graph which we shall call *expanded line path* and *expanded line cycle* (see Figures 3.1(b) and 3.2(b), respectively).



(a) Line cycle.

(b) Expanded line cycle.

Figure 3.2 A line cycle in the event-activity graph and the belonging expanded line cycle in the expanded event-activity graph ($T = 5$). Note that only a subset of the expanded turnaround arcs is displayed.

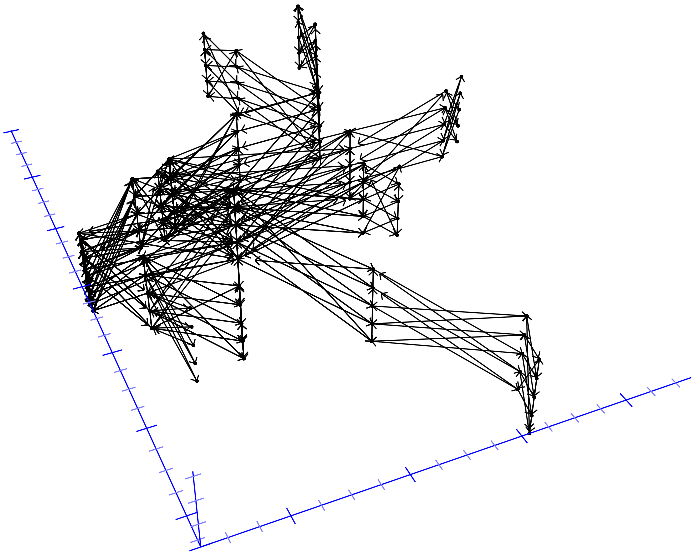


Figure 3.3 Expanded event-activity graph of a sample instance.

3.2 X-PESP Formulations

As indicated in the section above, we can solve the PERIODIC EVENT SCHEDULING PROBLEM by determining a special subgraph of the expanded event-activity graph. We now derive a set of new integer programming formulations for PESP based on this observation.

Again, consider an optimization instance $\mathcal{I} = (D, \ell, u, w, T)$ of the PESP and let $\boldsymbol{D} = (\boldsymbol{V}, \boldsymbol{A})$ denote its belonging expanded event-activity graph. We need two types of decision variables. One for the arcs of vehicle activities $a \in X$, and the other for the arcs of non-vehicle activities $a \in A \backslash X$ (see Section 2.2). With the notation $\boldsymbol{X} = \bigcup_{a \in X} \boldsymbol{A}(a)$ for the arcs belonging to vehicle activities, we define these variables by

$$x_{\boldsymbol{a}} = \begin{cases} 1, & \text{if arc } \boldsymbol{a} \in \boldsymbol{X} \text{ is selected,} \\ 0, & \text{otherwise, and} \end{cases} \qquad y_{\boldsymbol{a}} = \begin{cases} 1, & \text{if arc } \boldsymbol{a} \in \boldsymbol{A} \backslash \boldsymbol{X} \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

For abbreviation, we use the notations $\delta_{\boldsymbol{X}}(\boldsymbol{v}) = \delta(\boldsymbol{v}) \cap \boldsymbol{X}$ and $\delta_a(\boldsymbol{v}) = \delta(\boldsymbol{v}) \cap \boldsymbol{A}(a)$ for $\boldsymbol{v} \in \boldsymbol{V}$. In the following, let a_R be an arbitrary fixed vehicle activity of line $R \in \mathcal{R}$, where $\mathcal{R}$ denotes the set of operated lines. We get

X-PESP *basic* (3.1)

$$\text{Minimize} \qquad \sum_{\boldsymbol{a} \in \boldsymbol{X}} c_{\boldsymbol{a}}\, x_{\boldsymbol{a}} + \sum_{\boldsymbol{a} \in \boldsymbol{A} \setminus \boldsymbol{X}} c_{\boldsymbol{a}}\, y_{\boldsymbol{a}}$$

subject to

$$\begin{array}{rll}
\displaystyle\sum_{\boldsymbol{a} \in \boldsymbol{A}(a_R)} x_{\boldsymbol{a}} = 1 & \text{for all } R \in \mathcal{R} & (3.1a) \\
\displaystyle\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(\boldsymbol{v})} x_{\boldsymbol{a}} - \sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(\boldsymbol{v})} x_{\boldsymbol{a}} = 0 & \text{for all } \boldsymbol{v} \in \boldsymbol{V} & (3.1b) \\
\displaystyle\sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} y_{\boldsymbol{a}} = 1 & \text{for all } a \in A \setminus X & (3.1c) \\
\displaystyle\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(\boldsymbol{v})} x_{\boldsymbol{a}} - y_{\boldsymbol{b}} \geq 0 & \text{for all } \boldsymbol{b} = (\boldsymbol{v}, \boldsymbol{w}) \in \boldsymbol{A} \setminus \boldsymbol{X} & (3.1d) \\
\displaystyle\sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(\boldsymbol{w})} x_{\boldsymbol{a}} - y_{\boldsymbol{b}} \geq 0 & \text{for all } \boldsymbol{b} = (\boldsymbol{v}, \boldsymbol{w}) \in \boldsymbol{A} \setminus \boldsymbol{X} & (3.1e) \\
0 \leq x_{\boldsymbol{a}} \leq 1 & \text{for all } \boldsymbol{a} \in \boldsymbol{X} & (3.1f) \\
0 \leq y_{\boldsymbol{a}} \leq 1 & \text{for all } \boldsymbol{a} \in \boldsymbol{A} \setminus \boldsymbol{X} & (3.1g) \\
x_{\boldsymbol{a}} \in \mathbb{Z} & \text{for all } \boldsymbol{a} \in \boldsymbol{X} & (3.1h) \\
y_{\boldsymbol{a}} \in \mathbb{Z} & \text{for all } \boldsymbol{a} \in \boldsymbol{A} \setminus \boldsymbol{X} & (3.1i)
\end{array}$$

Notation 3.1 We denote the convex hull of all feasible solutions to the formulation X-PESP *basic* (3.1) by

$$\mathcal{P}_{\text{IP}}(\text{X-PESP } \textit{basic}) = \operatorname{conv}\{(x, y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A} \setminus \boldsymbol{X}} \mid (x, y) \text{ satisfies (3.1a)–(3.1i)}\}.$$

Furthermore, we write

$$\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{basic}) = \{(x, y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A} \setminus \boldsymbol{X}} \mid (x, y) \text{ satisfies (3.1a)–(3.1g)}\}$$

for the rational polyhedron of its LP relaxation.

Let us first restrict our attention to the constraints (3.1a) and (3.1b) that are exclusively defined on arcs belonging to vehicle activities. If we consider the value $x_{\boldsymbol{a}}$ as a flow on arc $\boldsymbol{a}$, then the constraints (3.1b) require flow conservation in every node $\boldsymbol{v} \in \boldsymbol{V}$. Now, the constraints (3.1a) say that for every line $R \in \mathcal{R}$ there must be a flow of value 1 over the cut $\boldsymbol{A}(a_R)$. In combination with the integrality constraints (3.1h) on the x variables, this yields an integral circulation of value 1 along the expanded line cycle of every line $R \in \mathcal{R}$ (see Figure 3.4). If there were only vehicle activities, then the constraints (3.1a)–(3.1c) as well as (3.1f) and (3.1h) would already lead to a feasible timetable. The departure and arrival times of all lines are determined and meet the conditions on trip, dwell, and turn activities.

The rest of the constraints is used to incorporate non-vehicle activities. By the constraints (3.1c), we ensure that for every non-vehicle activity $a \in A \backslash X$ precisely one arc $\boldsymbol{a}$ in $\boldsymbol{A}(a)$ is selected. But this condition is not sufficient to fulfill the periodic interval constraints of all activities, since the arc $\boldsymbol{a}$ may imply other timings for its related events than those defined by the arcs selected for vehicle activities. Roughly speaking, we need to enforce a coupling between selected arcs of non-vehicle activities and the circulations along the expanded line cycles. This is done by the constraints (3.1d) and (3.1e) which we shall therefore call *coupling constraints*. The objective function represents the cost of the timetable and is given as the sum of the weights of all arcs selected from the set $\boldsymbol{A}$.

As we will see later, we may relax the integrality constraints on the y variables.

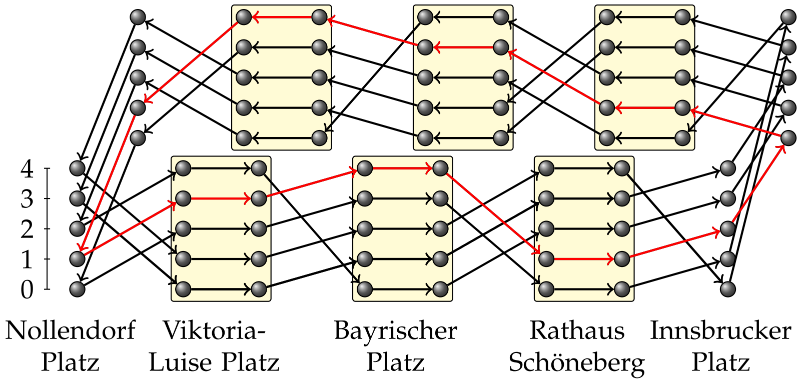


Figure 3.4 Integral circulation in an expanded line cycle.

The next proposition will be needed in the remainder of this section.

Proposition 3.2 Let $\mathcal{I} = (D, \ell, u, w, T)$ denote a PESP optimization instance, and let $\boldsymbol{D} = (\boldsymbol{V}, \boldsymbol{A})$ be the belonging expanded event-activity graph. If a vector $(x, y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A} \setminus \boldsymbol{X}}$ satisfies the constraints (3.1a)–(3.1b), then

$$\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(\boldsymbol{V}(v))} x_{\boldsymbol{a}} = \sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(\boldsymbol{V}(v))} x_{\boldsymbol{a}} = 1$$

for all $v \in V$.

Proof. Let $\mathcal{I}$, $\boldsymbol{D}$, and $(x, y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A} \setminus \boldsymbol{X}}$ satisfy the hypothesis of the proposition, and let $v \in V$. The node v is contained in the line cycle of some line $R \in \mathcal{R}$. That $\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(\boldsymbol{V}(v))} x_{\boldsymbol{a}} = \sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(\boldsymbol{V}(v))} x_{\boldsymbol{a}}$ follows from the constraints (3.1b). To see that both sums have the value 1, observe that there is a unique vehicle activity $a \in X$ with the property that $\boldsymbol{A}(a) = \delta^-_{\boldsymbol{X}}(\boldsymbol{V}(v))$. If $a = a_R$, then the proposition follows directly from the constraints (3.1a). If, on the other hand, $a \neq a_R$, we invoke the constraints (3.1b) in the same way as before. By induction, for every arc a' that follows a_R on the line cycle, we have $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a')} x_{\boldsymbol{a}} = 1$, and the proposition follows. ■

We are now in a position to show why we can relax the integrality constraints for the y variables in X-PESP *basic* (3.1).

Lemma 3.3 If a vector $(x, y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A} \setminus \boldsymbol{X}}$ satisfies the constraints (3.1a)–(3.1h), then the vector y is integer.

Proof. Let $b = (v, w) \in A \setminus X$. By Proposition 3.2, we have $\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(\boldsymbol{V}(v))} x_{\boldsymbol{a}} = 1$ and $\sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(\boldsymbol{V}(w))} x_{\boldsymbol{a}} = 1$. Since x is integral, there is precisely one node $\boldsymbol{v} \in \boldsymbol{V}(v)$ with $\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(\boldsymbol{v})} x_{\boldsymbol{a}} = 1$, while for all other nodes $\boldsymbol{v}' \in \boldsymbol{V}(v)$ we have $\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(\boldsymbol{v}')} x_{\boldsymbol{a}} = 0$. For the same reason, there is a unique node $\boldsymbol{w} \in \boldsymbol{V}(w)$ such that $\sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(\boldsymbol{w})} x_{\boldsymbol{a}} = 1$ and $\sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(\boldsymbol{w}')} x_{\boldsymbol{a}} = 0$ for all nodes $\boldsymbol{w}' \in \boldsymbol{V}(w)$ with $\boldsymbol{w}' \neq \boldsymbol{w}$. The coupling constraints (3.1d) and (3.1e) now imply that $y_{\boldsymbol{b}} = 0$ for all arcs $\boldsymbol{b} \in \boldsymbol{A}(b)$ with $\boldsymbol{b} \neq (\boldsymbol{v}, \boldsymbol{w})$. Finally, we deduce from the constraints (3.1c) that $y_{\boldsymbol{b}} = 1$ for $\boldsymbol{b} = (\boldsymbol{v}, \boldsymbol{w})$. Hence, all y variables are integer. ■

Lemma 3.4 There is a one-to-one correspondence between the integer solutions of the PESP and the solutions to X-PESP *basic* (3.1).

Proof. The proof is straightforward. Consider a feasible optimization instance $\mathcal{I} = (D, \ell, u, w, T)$ of the PESP. For every integer solution to the instance $\mathcal{I}$, we show how to construct the corresponding solution to X-PESP *basic* (3.1), and vice versa.

Let $\pi \in \{0,\dots,T-1\}^V$ be a solution to $\mathcal{I}$. The corresponding solution to the formulation X-PESP *basic* (3.1) is obtained as follows: Let $a = (v,w) \in X$. Since the vector π is feasible for the instance $\mathcal{I}$, we have $(\pi_w - \pi_v - \ell_a) \bmod T \leq u_a - \ell_a$. By construction, there is precisely one arc $\boldsymbol{a} \in \boldsymbol{A}(a)$ with $\boldsymbol{a} = (v[\pi_v], w[\pi_w])$ in the expanded event-activity graph that models this choice of timings for the events v and w. Set $x_{\boldsymbol{a}} = 1$. An activity $a = (v,w) \in A \backslash X$ is handled in much the same way. The only difference is that we set $y_{\boldsymbol{a}} = 1$ for the respective arc $\boldsymbol{a} = (v[\pi_v], w[\pi_w]) \in \boldsymbol{A}(a)$. This defines the PESP solution in terms of a vector $(x,y) \in \mathbb{Q}^{\boldsymbol{A}}$, and the task is now to prove that (x,y) is feasible to X-PESP *basic* (3.1).

- ❏ **Proof of the constraints** (3.1a)**.** Let $R \in \mathcal{R}$. By definition, we have $a_R \in X$. Following the above remarks, there is precisely one arc $\boldsymbol{a} \in \boldsymbol{A}(a_R)$ with $x_{\boldsymbol{a}} = 1$. This gives $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a_R)} x_{\boldsymbol{a}} = 1$.

- ❏ **Proof of the constraints** (3.1b)**.** Let $v[t] \in \boldsymbol{V}$. If $\pi_v \neq t$, then all arcs in $\delta^-_{\boldsymbol{X}}(v[t]) \cup \delta^+_{\boldsymbol{X}}(v[t])$ are set to zero, and hence the constraints are obviously satisfied. The interesting case is $\pi_v = t$. Since the node v is part of the line cycle of some line $R \in \mathcal{R}$, there exist distinguished vehicle activities $a, a' \in X$ with $\{a\} = \delta^-_X(v)$ and $\{a'\} = \delta^+_X(v)$. By the definition of x, there are unique arcs $\boldsymbol{a} = (\boldsymbol{u}, v[t]) \in \boldsymbol{A}(a)$ and $\boldsymbol{a}' = (v[t], \boldsymbol{w}) \in \boldsymbol{A}(a')$ with $x_{\boldsymbol{a}} = 1 = x_{\boldsymbol{a}'}$. Together with $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} x_{\boldsymbol{a}} = 1 = \sum_{\boldsymbol{a} \in \boldsymbol{A}(a')} x_{\boldsymbol{a}}$ this implies flow conservation in $v[t]$.

- ❏ **Proof of the constraints** (3.1c)**.** Consider an activity $a \in A \backslash X$. By the definition of the vector y, there is precisely one arc $\boldsymbol{a} \in \boldsymbol{A}(a)$ with $y_{\boldsymbol{a}} = 1$, which is the desired conclusion.

- ❏ **Proof of the constraints** (3.1d) **and** (3.1e)**.** We give the proof only for the constraints under (3.1d); the remaining constraints can be handled in the same way. Let $\boldsymbol{b} = (v[t], w[t']) \in \boldsymbol{A} \backslash \boldsymbol{X}$. We first turn to the case $\pi_v \neq t$ or $\pi_w \neq t'$. It follows immediately from the definition of the vector y that $y_{\boldsymbol{b}} = 0$. Thus, inequality (3.1d) holds. What is left is the case $\pi_v = t$ and $\pi_w = t'$. According to the proof of the constraints (3.1b), we have $\sum_{\boldsymbol{a} \in \delta^-_{\boldsymbol{X}}(v[t])} x_{\boldsymbol{a}} = 1$. Since $y_{\boldsymbol{b}} \in \{0,1\}$ by the constraints (3.1i), this proves the claim.

- ❏ **Proof of the constraints** (3.1f)–(3.1i)**.** Immediately from the definition of the vector (x,y).

Now, consider a solution $(x,y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A} \backslash \boldsymbol{X}}$ that is feasible to the formulation X-PESP *basic* (3.1). The corresponding PESP solution $\pi \in \{0,\dots,T-1\}^V$ is defined as follows. Let $v \in V$. Then, there is a unique vehicle activity $a = (v,w) \in X$ and, according to the proof of Proposition 3.2, we have $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} x_{\boldsymbol{a}} = 1$. Since the vector x is integral, there is only one arc in $\boldsymbol{A}(a)$, say $\boldsymbol{a} = (v[t], w[t'])$, with $x_{\boldsymbol{a}} = 1$. Set $\pi_v = t$.

The proof is completed by showing that the vector π as defined above satisfies the periodic interval constraints (2.1) for all activities in A. First, suppose $a \in A \backslash X$. Applying the constraints (3.1c) and (3.1i), we can assert that there is unique arc $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}(a)$ with $y_{\boldsymbol{a}} = 1$. As all arcs in $\boldsymbol{A}(a)$, the arc $\boldsymbol{a}$ respects the periodic interval constraint of activity a in the sense that $(t' - t - \ell_a) \bmod T \leq u_a - \ell_a$. The coupling constraints (3.1d)–(3.1e) now imply that $\pi_v = t$ and $\pi_w = t'$, which finishes the proof for the case $a \in A \backslash X$.
The same argumentation works for $a \in X$. We only need to show that $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} x_{\boldsymbol{a}} = 1$. Each arc $a \in X$ belongs to the line cycle of some line $R \in \mathcal{R}$. According to the above remarks, we have an integral circulation around the corresponding expanded line cycle , and so for one arc $\boldsymbol{a} \in \boldsymbol{A}(a)$ the value $x_{\boldsymbol{a}} = 1$, while for the remaining arcs in $\boldsymbol{A}(a)$ we have $x_{\boldsymbol{a}} = 0$. This is our claim. ■

The coupling constraints (3.1d)–(3.1e) leave room for improvement. It is possible to strengthen these constraints and with the notation $\mathcal{T} = \{0, \ldots, T-1\}$, we obtain

X-PESP *plus* (3.2)

$$\text{Minimize} \quad \sum_{a \in X} c_a\, x_a + \sum_{a \in A \setminus X} c_a\, y_a$$

subject to

$$\begin{aligned}
\sum_{a \in A(a_R)} x_a &= 1 && \text{for all } R \in \mathcal{R} && \text{(3.1a)}\\
\sum_{a \in \delta_X^-(v)} x_a - \sum_{a \in \delta_X^+(v)} x_a &= 0 && \text{for all } \boldsymbol{v} \in \boldsymbol{V} && \text{(3.1b)}\\
\sum_{a \in A(a)} y_a &= 1 && \text{for all } a \in A \setminus X && \text{(3.1c)}\\
\sum_{a \in \delta_X^-(v[t])} x_a - \sum_{a \in \delta_a^+(v[t])} y_a &\geq 0 && \text{for all } a = (v,w) \in A \setminus X,\ t \in \mathcal{T} && \text{(3.2a)}\\
\sum_{a \in \delta_X^+(w[t])} x_a - \sum_{a \in \delta_a^-(w[t])} y_a &\geq 0 && \text{for all } a = (v,w) \in A \setminus X,\ t \in \mathcal{T} && \text{(3.2b)}\\
0 \leq x_a &\leq 1 && \text{for all } \boldsymbol{a} \in \boldsymbol{X} && \text{(3.1f)}\\
0 \leq y_a &\leq 1 && \text{for all } \boldsymbol{a} \in \boldsymbol{A} \setminus \boldsymbol{X} && \text{(3.1g)}\\
x_a &\in \mathbb{Z} && \text{for all } \boldsymbol{a} \in \boldsymbol{X} && \text{(3.1h)}\\
y_a &\in \mathbb{Z} && \text{for all } \boldsymbol{a} \in \boldsymbol{A} \setminus \boldsymbol{X}. && \text{(3.1i)}
\end{aligned}$$

Notation 3.5 We define the following rational polyhedra.

$$\mathcal{P}_{\text{IP}}(\text{X-PESP } \textit{plus}) = \operatorname{conv}\{(x,y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A \setminus X}} \mid (x,y) \text{ satisfies (3.1a)–(3.1c), (3.2a), (3.2b), and (3.1f)–(3.1i)}\}$$

$$\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{plus}) = \{(x,y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A \setminus X}} \mid (x,y) \text{ satisfies (3.1a)–(3.1c), (3.2a), (3.2b), (3.1f), and (3.1g)}\}.$$

Lemma 3.6

$$\mathcal{P}_{\text{IP}}(\text{X-PESP } \textit{plus}) = \mathcal{P}_{\text{IP}}(\text{X-PESP } \textit{basic}), \tag{3.3}$$

and

$$\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{plus}) \subseteq \mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{basic}). \tag{3.4}$$

Proof. We begin by proving (3.3). The idea is to show that every solution to the formulation X-PESP *basic* (3.1) is feasible to X-PESP *plus* (3.2). Since both formulations only differ in the coupling constraints, it suffices to verify (3.2a) and (3.2b). Suppose $(x, y) \in \mathcal{P}_{\text{IP}}(\text{X-PESP } \textit{basic})$. Let $a \in A \backslash X$, and let $t \in \{0, \ldots, T-1\}$. By the constraints (3.1c), and (3.1i), there is precisely one arc in $\boldsymbol{A}(a)$, say $\boldsymbol{a} = (v[t_1], w[t_2])$, with $y_{\boldsymbol{a}} = 1$. If $t \neq t_1$, then $\sum_{\boldsymbol{a} \in \delta_a^+(v[t])} y_{\boldsymbol{a}} = 0$ and the constraint (3.2a) is clearly satisfied. On the other hand, if $t = t_1$, we have $\sum_{\boldsymbol{a} \in \delta_a^+(v[t])} y_{\boldsymbol{a}} = 1$. In this case we can write

$$\sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^-(v[t])} x_{\boldsymbol{a}} - \sum_{\boldsymbol{a} \in \delta_a^+(v[t])} y_{\boldsymbol{a}} = \sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^-(v[t])} x_{\boldsymbol{a}} - y_{\boldsymbol{a}} \geq 0,$$

where the last inequality follows from the constraints (3.1d). In the same manner we see that the vector (x, y) satisfies the constraints (3.2b).

It remains to prove the relation (3.4). First, observe that the constraints (3.1d) and (3.1e) are dominated by the ones under (3.2a)–(3.2b). To see this, consider an arc $\boldsymbol{b} = (v[t], w[t']) \in \boldsymbol{A} \backslash \boldsymbol{X}$. Let $a \in A \backslash X$ with $\boldsymbol{b} \in \boldsymbol{A}(a)$. Applying the constraints (3.2a)–(3.2b) we obtain

$$\sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^-(v[t])} x_{\boldsymbol{a}} \geq \sum_{\boldsymbol{a} \in \delta_a^+(v[t])} y_{\boldsymbol{a}} \geq y_{\boldsymbol{b}}$$

and

$$\sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^+(w[t'])} x_{\boldsymbol{a}} \geq \sum_{\boldsymbol{a} \in \delta_a^-(w[t])} y_{\boldsymbol{a}} \geq y_{\boldsymbol{b}}.$$

This gives $\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{plus}) \subseteq \mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{basic})$. ■

We later show that, under certain conditions, the polyhedron $\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{plus})$ is even a proper subset of $\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{basic})$.

Starting from X-PESP *plus* (3.2) we are able to provide another formulation in which the coupling constraints are stated as equality constraints. It will turn out that this makes the constraints (3.1c) unnecessary. Again, we use $\mathcal{T}$ as a shorthand for $\{0,\dots,T-1\}$ and obtain

X-PESP *pro* (3.5)

$$\begin{aligned}
&\text{Minimize} \quad \sum_{a\in X} c_a\, x_a + \sum_{a\in A\setminus X} c_a\, y_a \\
&\text{subject to} \\
&\sum_{a\in A(a_R)} x_a = 1 && \text{for all } R\in\mathcal{R} && (3.1a)\\
&\sum_{a\in\delta_X^-(v)} x_a - \sum_{a\in\delta_X^+(v)} x_a = 0 && \text{for all } \boldsymbol{v}\in \boldsymbol{V} && (3.1b)\\
&\sum_{a\in\delta_X^-(v[t])} x_a - \sum_{a\in\delta_a^+(v[t])} y_a = 0 && \text{for all } a=(v,w)\in A\setminus X,\ t\in\mathcal{T} && (3.5a)\\
&\sum_{a\in\delta_X^+(w[t])} x_a - \sum_{a\in\delta_a^-(w[t])} y_a = 0 && \text{for all } a=(v,w)\in A\setminus X,\ t\in\mathcal{T} && (3.5b)\\
&0\le x_a\le 1 && \text{for all } \boldsymbol{a}\in \boldsymbol{X} && (3.1f)\\
&0\le y_a\le 1 && \text{for all } \boldsymbol{a}\in \boldsymbol{A}\setminus\boldsymbol{X} && (3.1g)\\
&x_a\in\mathbb{Z} && \text{for all } \boldsymbol{a}\in \boldsymbol{X} && (3.1h)\\
&y_a\in\mathbb{Z} && \text{for all } \boldsymbol{a}\in \boldsymbol{A}\setminus\boldsymbol{X}. && (3.1i)
\end{aligned}$$

Notation 3.7 Let us introduce the following notation.

$$\mathcal{P}_{\text{IP}}(\text{X-PESP } \textit{pro}) = \operatorname{conv}\{(x,y)\in\mathbb{Q}^X\times\mathbb{Q}^{A\setminus X} \mid (x,y) \text{ satisfies (3.1a), (3.1b),} \\ \text{(3.5a), (3.5b), and (3.1f)–(3.1i)}\}$$

$$\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{pro}) = \{(x,y)\in\mathbb{Q}^X\times\mathbb{Q}^{A\setminus X} \mid (x,y) \text{ satisfies (3.1a), (3.1b), (3.5a),} \\ \text{(3.5b), (3.1f), and (3.1g)}\}.$$

Remark 3.8 Proposition 3.2 gives the reason why we can omit the constraints (3.1c) in X-PESP *pro* (3.5). In fact, if $a=(v,w)\in A\setminus X$, then the combination of the constraints (3.5a) belonging to activity a, together with Proposition 3.2, give

$$\sum_{a\in A(a)} y_a = \sum_{a\in\delta_X^-(V(v))} x_a = 1.$$

Lemma 3.9

$$\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{pro}) = \mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{plus}).$$

Proof. The idea of the proof is to show that the equality constraints under (3.5a) and (3.5b) are already implied by X-PESP *plus* (3.2).

Let $(x, y) \in \mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{plus})$, and let $a \in A\backslash X$. Then, the constraints (3.2a) state

$$\sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^{-}(v[t])} x_{\boldsymbol{a}} - \sum_{\boldsymbol{a} \in \delta_{a}^{+}(v[t])} y_{\boldsymbol{a}} \geq 0$$

for all $t \in \{0, \ldots, T-1\}$. Combining these inequalities we obtain

$$\sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^{-}(\boldsymbol{V}(v))} x_{\boldsymbol{a}} - \sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} y_{\boldsymbol{a}} \geq 0.$$

Proposition 3.2 and the constraints (3.1c) now give $\sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^{-}(\boldsymbol{V}(v))} x_{\boldsymbol{a}} = \sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} y_{\boldsymbol{a}} = 1$, and hence we must have

$$\sum_{\boldsymbol{a} \in \delta_{\boldsymbol{X}}^{-}(v[t])} x_{\boldsymbol{a}} - \sum_{\boldsymbol{a} \in \delta_{a}^{+}(v[t])} y_{\boldsymbol{a}} \leq 0$$

for all $t \in \{0, \ldots, T-1\}$. Thus, the constraints (3.2a) are satisfied by equality. The constraints (3.2b) can be handled in the same way. ■

We know from Lemma 2.7 that the integer programming formulations for the PESP presented in Section 2.2.4 have a very poor LP relaxation. Unfortunately, the same applies to the X-PESP formulations.

Lemma 3.10 The optimal solution value to the LP relaxation of X-PESP *pro* (3.5) is zero.

Proof. The proof is by construction. We define a vector $(x, y) \in \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A}\backslash\boldsymbol{X}}$ that is satisfies the constraints (3.1a), (3.1b), (3.5a),(3.5b), (3.1f), and (3.1g) and has zero cost.

Let $\mathcal{I} = (D, \ell, u, w, T)$ be feasible optimization instance of PESP, and let $\boldsymbol{D}$ be the belonging expanded event-activity graph. Then, for each activity $a \in A$ and point in time t in $\{0, \ldots, T-1\}$, there is precisely one arc $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}(a)$

satisfying $(t' - t) \bmod T = \ell_a$ (cf. Section 3.1). We set $x_{\boldsymbol{a}} = 1/T$ if $a \in X$ and $y_{\boldsymbol{a}} = 1/T$ otherwise. The remaining variables are set to zero.

We first show that the resulting vector (x, y) is feasible to the LP relaxation of X-PESP *pro* (3.5).

❑ **Proof of the constraints (3.1a).** Let $R \in \mathcal{R}$. According to the definition of the vector (x, y), there are precisely T arcs in the set $\boldsymbol{A}(a_R)$ for which the vector x defines a value of $1/T$. The components of x belonging to the remaining arcs in $\boldsymbol{A}(a_R)$ are zero. Hence, summing over the arcs in the set $\boldsymbol{A}(a_R)$ gives $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a_R)} x_{\boldsymbol{a}} = 1$, which is the desired conclusion.

❑ **Proof of the constraints (3.1b).** Let $\boldsymbol{v} \in \boldsymbol{V}$. Then, there exists an event $v \in V$ such that $\boldsymbol{v} \in \boldsymbol{V}(v)$ and there are unique vehicle activities $a,\ a' \in X$ with $\delta^-_{\boldsymbol{X}}(\boldsymbol{V}(v)) = \boldsymbol{A}(a)$, and $\delta^+_{\boldsymbol{X}}(\boldsymbol{V}(v)) = \boldsymbol{A}(a')$. Notice that both $\boldsymbol{A}(a)$ and $\boldsymbol{A}(a')$ contain precisely T arcs that are set to $1/T$ in the x vector, and that the node set $\boldsymbol{V}(v)$ has cardinality T. If we show that neither $\boldsymbol{A}(a)$ nor $\boldsymbol{A}(a')$ contains two arcs with common end point, then flow is conserved in $\boldsymbol{v}$.

Let $\boldsymbol{a}_1 = (v[t_1], w[t'_1])$ and $\boldsymbol{a}_2 = (v[t_2], w[t'_2])$ be two distinct arcs in $\boldsymbol{A}(a)$ at value $1/T$ in x. By the definition of x, we have $(t'_1 - t_1) \bmod T = \ell_a$, and $(t'_2 - t_2) \bmod T = \ell_a$. Now suppose $t_1 = t_2$, i. e. $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ share the same start point. It follows that $t_2 = t'_2$, which contradicts the fact that $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are distinct. A similar contradiction arises if we assume that $t'_1 = t'_2$. Drawing the same conclusion for the set $\boldsymbol{A}(a')$ gives equation (3.1b).

❑ **Proof of the constraints (3.5a) and (3.5b).** Let $a = (v, w) \in A \backslash X$, and let $t \in \{0, \ldots, T-1\}$. From what has already been proved, we have $\sum_{a \in \delta^-_{\boldsymbol{X}}(v[t])} x_{\boldsymbol{a}} = 1/T = \sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(v[t])} x_{\boldsymbol{a}}$. Recalling the definition of the vector (x, y) it follows that $\sum_{\boldsymbol{a} \in \delta^+_a} y_{\boldsymbol{a}} = 1/T$, which gives equation (3.5a). Again, no two arcs in the set $\boldsymbol{A}(a)$ have an end point in common, so that $\sum_{\boldsymbol{a} \in \delta^-_a(w[t])} y_{\boldsymbol{a}} = 1/T$, too. This establishes equation (3.5b).

❑ **Proof of the constraints (3.1f) and (3.1g).** The non-negativity of the vector (x, y) follows directly from its definition.

We finally show that the value of the solution is zero. Since we have a minimization problem, this combined with the non-negativity of the cost coefficients and the variables will imply optimality.

Recall from Section 3.1 that $c_{\boldsymbol{a}} = w_a\,(t' - t - \ell_a) \bmod T$ for $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}$. All arcs in $\boldsymbol{A}$ that are assigned a positive value by the vector (x, y) have a slack time $(t' - t - \ell_a)$ of zero. Therefore, we can conclude that the solution value is zero, which completes the proof. ■

The next theorem expresses the relation between the polyhedra of the LP relaxations of the above X-PESP formulations.

Theorem 3.11

$$\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{pro}) = \mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{plus}) \subseteq \mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{basic})$$

Proof. Immediate from Lemmas 3.6 and 3.9. ∎

Remark 3.12 If the considered PESP timetabling instances fulfill some minimal requirements, we can even show that $\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{plus})$ is a proper subset of $\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{basic})$. In particular, we need the following conditions.

- The period time $T \geq 2$.
- The set of events V is not empty.
- The set of vehicle activities X is not empty.
- The set of non-vehicle activities $A \backslash X$ is not empty.
- There exists a non-vehicle activity $a^* = (v, w) \in A \backslash X$ with $\ell_{a^*} \neq u_{a^*}$.

For every PESP instance satisfying these requirements, we can find a feasible solution to the LP relaxation of X-PESP *basic* (3.1) that violates the coupling constraints of X-PESP *plus* (3.2). The construction follows the proof of Lemma 3.10. For all vehicle activities $a \in X$, we set $x_{\boldsymbol{a}} = 1$ for the arc $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}(a)$ that satisfies $(t' - t) \bmod T = \ell_a$. In the same way we define the vector y for all activities $a \in A \backslash X$, except for the activity $a^*(v, w)$.
By our assumption $\ell_{a^*} \neq u_{a^*}$, there are at least two arcs in $\boldsymbol{A}(a)$ entering the node $w[t^*]$ with $t^* \in \{0, \ldots, T-1\}$. For two of them, say $\boldsymbol{a}_1 = (v[t_1], w[t^*])$ and $\boldsymbol{a}_2 = (v[t_2], w[t^*])$, we set $y_{\boldsymbol{a}_1} = y_{\boldsymbol{a}_2} = 1/T$. In addition, for every $t \in \{0, \ldots, T-1\}$ with $t \neq t_1$ and $t \neq t_2$, we choose an arbitrary arc $\boldsymbol{a} \in \boldsymbol{A}(a)$ leaving $v[t]$ and set $y_{\boldsymbol{a}} = 1/T$.

It follows that the so-defined solution is feasible to X-PESP *basic* (3.1), but violates the constraints (3.2b) at least for the node $w[t^*]$. The conflict is given by $1/T - 2/T \neq 0$. Figure 3.5 illustrates this for the case $[\ell_{a^*}, u_{a^*}] = [0, 1]$.

We are now interested in an analytical comparison of X-PESP *pro* (3.5) with the formulation PESP *box* (2.6). To this end, consider a PESP optimization instance (D, ℓ, u, w, T) and its belonging expanded event-activity graph $\boldsymbol{D} = (\boldsymbol{V}, \boldsymbol{A})$.

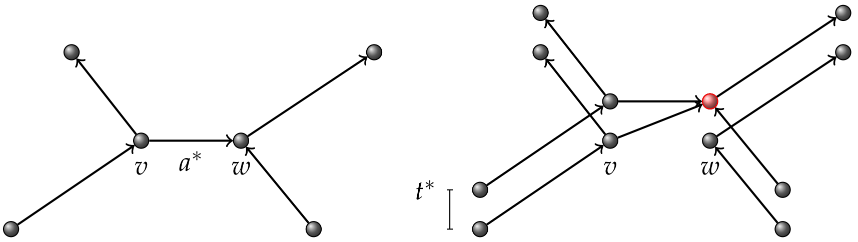


Figure 3.5 Illustration of Remark 3.12. Part of a PESP instance and the belonging expanded event-activity graph. All activities different from (v, w) are vehicle activities. Displayed arcs in the expanded event-activity graph have the value $1/T$. Arcs with zero value have been omitted.

Theorem 3.13 There is a linear transformation $\psi : \mathbb{Q}^{\boldsymbol{A}} \to \mathbb{Q}^{V+A}$ with the property that

$$\psi(\mathcal{P}_{\text{IP}}(\text{X-PESP } \textit{pro})) = \mathcal{P}_{\text{IP}}(\text{PESP } \textit{box}). \tag{3.6}$$

and

$$\psi(\mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{pro})) \subseteq \mathcal{P}_{\text{LP}}(\text{PESP } \textit{box}), \tag{3.7}$$

Proof. The proof is by construction. According to Lemma 2.11, we may assume, without loss of generality, that $u_a \leq T$ for all activities $a \in A$. We define the linear transformation ψ by

$$\psi : \mathbb{Q}^{\boldsymbol{X}} \times \mathbb{Q}^{\boldsymbol{A} \setminus \boldsymbol{X}} \to \mathbb{Q}^{V} \times \mathbb{Q}^{A}, \qquad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \left[\begin{array}{c|c} \Pi & \mathbf{0} \\ \hline \multicolumn{2}{c}{P} \end{array} \right] \begin{pmatrix} x \\ y \end{pmatrix}, \tag{3.8}$$

where

$$\begin{aligned}
&\Pi \in \mathbb{Q}^{V \times \boldsymbol{X}}, && \Pi = SR, \\
&R \in \mathbb{Q}^{\boldsymbol{V} \times \boldsymbol{X}}, && R_{\boldsymbol{v},\boldsymbol{a}} = \begin{cases} t, & \text{if } \boldsymbol{a} = (v[t], w[t']) \in \delta^+_{\boldsymbol{X}}(\boldsymbol{v}), \\ 0, & \text{otherwise,} \end{cases} \\
&S \in \{0,1\}^{V \times \boldsymbol{V}}, && S_{v,\boldsymbol{v}} = \begin{cases} 1, & \text{if } \boldsymbol{v} \in \boldsymbol{V}(v), \\ 0, & \text{otherwise,} \end{cases} \\
&P \in \mathbb{Q}^{A \times \boldsymbol{A}}, && P_{a,\boldsymbol{a}} = \begin{cases} 0, & \text{if } \boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}(a) \text{ and } t \leq t', \\ 1, & \text{otherwise.} \end{cases}
\end{aligned}$$

Note the definition of the matrix P. For an activity $a = (v, w) \in A$, every arc $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}(a)$ implies a sequence of the events v and w in the interval

$[0, T)$. The matrix entry $P_{a,\boldsymbol{a}}$ equals the periodical offset $p_a \in \{0, 1\}$ that is needed in the periodic interval constraint

$$\ell_a \leq t' - t + T\, p_a \leq u_a$$

according to Remark 2.10.

Let $(x, y) \in \mathcal{P}_{\text{LP}}(\text{X-PESP } \textit{pro})$, and let $(\pi, p) = \psi((x, y))$. We first show that (π, p) is a feasible solution to the LP relaxation of PESP *box* (2.6) which then implies statement (3.7).

❑ **Proof of the constraints** (2.4a) **and** (2.4b) **for** $a \in X$**.** We have to show that $\ell_a \leq \pi_w - \pi_v + T\, p_a \leq u_a$ for all $a = (v, w) \in X$. To this end, let $a \in X$. Then, for every arc $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}(a)$, we have

$$\ell_a \leq t' - t + T\, P_{a,\boldsymbol{a}} \leq u_a.$$

We multiply this relation by $x_{\boldsymbol{a}}$ which gives

$$\ell_a\, x_{\boldsymbol{a}} \leq (t' - t + T\, P_{a,\boldsymbol{a}})\, x_{\boldsymbol{a}} \leq u_a\, x_{\boldsymbol{a}},$$

keeping in mind that $x_{\boldsymbol{a}}$ is non-negative. Summing over all arcs in $\boldsymbol{A}(a)$ yields

$$\ell_a \sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} x_{\boldsymbol{a}} \leq \sum_{\substack{\boldsymbol{a} \in \boldsymbol{A}(a)\\ \boldsymbol{a} = (v[t], w[t'])}} (t' - t + T\, P_{a,\boldsymbol{a}})\, x_{\boldsymbol{a}} \leq u_a \sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} x_{\boldsymbol{a}}.$$

From the proof of Proposition 3.2 we conclude $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} x_{\boldsymbol{a}} = 1$, and hence

$$\ell_a \leq \sum_{\substack{\boldsymbol{a} \in \boldsymbol{A}(a)\\ \boldsymbol{a} = (v[t], w[t'])}} (t' - t + T\, P_{a,\boldsymbol{a}})\, x_{\boldsymbol{a}} \leq u_a,$$

which can be rewritten to

$$\ell_a \leq \sum_{\substack{\boldsymbol{a} \in \boldsymbol{A}(a)\\ \boldsymbol{a} = (v[t], w[t'])}} t'\, x_{\boldsymbol{a}} - \sum_{\substack{\boldsymbol{a} \in \boldsymbol{A}(a)\\ \boldsymbol{a} = (v[t], w[t'])}} t\, x_{\boldsymbol{a}} + T \sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\, x_{\boldsymbol{a}} \leq u_a.$$

Definition (3.8) now gives $p = P\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$ and, in particular, $p_a = \sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\, x_{\boldsymbol{a}}$, since $a \in X$. Therefore, the proof is completed by showing that

$$\sum_{\substack{\boldsymbol{a} \in \boldsymbol{A}(a)\\ \boldsymbol{a} = (v[t], w[t'])}} t'\, x_{\boldsymbol{a}} = \pi_w, \qquad \text{and} \qquad \sum_{\substack{\boldsymbol{a} \in \boldsymbol{A}(a)\\ \boldsymbol{a} = (v[t], w[t'])}} t\, x_{\boldsymbol{a}} = \pi_v.$$

First, consider the equation on the left-hand side. We derive

$$\sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t'\,x_{\boldsymbol{a}} = \sum_{w[t']\in\boldsymbol{V}(w)} \Big(t' \sum_{\boldsymbol{a}\in\delta^-_{\boldsymbol{X}}(w[t'])} x_{\boldsymbol{a}}\Big) \overset{(3.1b)}{=} \sum_{w[t']\in\boldsymbol{V}(w)} \Big(t' \sum_{\boldsymbol{a}\in\delta^+_{\boldsymbol{X}}(w[t'])} x_{\boldsymbol{a}}\Big) \overset{(3.8)}{=} \pi_w,$$

where for the first equality we have applied that a is the only arc in X that enters the node w, so that $\delta^-_a(w[t']) = \delta^-_{\boldsymbol{X}}(w[t'])$.

We now apply the argumentation again, with $\delta^+_a(v[t]) = \delta^+_{\boldsymbol{X}}(v[t])$, to obtain

$$\sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t\,x_{\boldsymbol{a}} = \sum_{v[t]\in\boldsymbol{V}(v)} \Big(t \sum_{\boldsymbol{a}\in\delta^+_{\boldsymbol{X}}(v[t])} x_{\boldsymbol{a}}\Big) \overset{(3.8)}{=} \pi_v.$$

❑ **Proof of the constraints** (2.4a) **and** (2.4b) **for** $a \in A\backslash X$**.** The task is now to show that $\ell_a \leq \pi_w - \pi_v + T\,p_a \leq u_a$ for all $a = (v,w) \in A\backslash X$. Let $a = (v,w) \in A\backslash X$. Then, by the definition of the matrix P, we have $\ell_a \leq t' - t + T\,P_{a,\boldsymbol{a}} \leq u_a$ for every arc $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{A}(a)$. Summing over all arcs in $\boldsymbol{A}(a)$ gives

$$\ell_a \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} y_{\boldsymbol{a}} \leq \sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} (t' - t + T\,P_{a,\boldsymbol{a}})\,y_{\boldsymbol{a}} \leq u_a \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} y_{\boldsymbol{a}}.$$

Keeping in mind Remark 3.8, we derive

$$\ell_a \leq \sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t'\,y_{\boldsymbol{a}} - \sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t\,y_{\boldsymbol{a}} + T \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\,y_{\boldsymbol{a}} \leq u_a.$$

If we recall that $p = P\left(\begin{smallmatrix}x\\y\end{smallmatrix}\right)$, and hence $p_a = \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\,y_{\boldsymbol{a}}$, then we are reduced to proving that

$$\sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t'\,y_{\boldsymbol{a}} = \pi_w, \qquad \text{and} \qquad \sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t\,y_{\boldsymbol{a}} = \pi_v.$$

The equation on the left-hand side results from

$$\sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t'\,y_{\boldsymbol{a}} = \sum_{w[t']\in\boldsymbol{V}(w)} \Big(t' \sum_{\boldsymbol{a}\in\delta^-_{a}(w[t'])} y_{\boldsymbol{a}}\Big) \overset{(3.5b)}{=} \sum_{w[t']\in\boldsymbol{V}(w)} \Big(t' \sum_{\boldsymbol{a}\in\delta^+_{\boldsymbol{X}}(w[t'])} x_{\boldsymbol{a}}\Big) \overset{(3.8)}{=} \pi_w.$$

That on the right-hand side is shown by

$$\sum_{\substack{\boldsymbol{a}\in\boldsymbol{A}(a)\\ \boldsymbol{a}=(v[t],w[t'])}} t\, y_{\boldsymbol{a}} = \sum_{v[t]\in\boldsymbol{V}(v)} \Big(t \sum_{\boldsymbol{a}\in\delta_a^+(v[t])} y_{\boldsymbol{a}}\Big) \overset{(3.5a)}{=} \sum_{v[t]\in\boldsymbol{V}(v)} \Big(t \sum_{\boldsymbol{a}\in\delta_{\boldsymbol{X}}^-(v[t])} x_{\boldsymbol{a}}\Big) \overset{(3.1b),(3.8)}{=} \pi_v.$$

❑ **Proof of the constraints** (2.4c)**.** We have to prove that $\pi_v \in [0, T-1]$ for all $v \in V$. By the definition of ψ, we have $\pi = \Pi x = SRx$ and, in particular,

$$\pi_v = \sum_{v\in\boldsymbol{V}(v)} \sum_{\substack{\boldsymbol{a}\in\delta_{\boldsymbol{X}}^+(\boldsymbol{v})\\ \boldsymbol{a}=(v[t],w[t'])}} t\, x_{\boldsymbol{a}}$$

for $v \in V$. Of course $\pi_v \geq 0$, since t and $x_{\boldsymbol{a}}$ are non-negative. Therefore, the only point remaining is to show that $\pi_v \leq T-1$. To this end, observe the following

$$\begin{aligned}\pi_v = \sum_{v\in\boldsymbol{V}(v)} \sum_{\substack{\boldsymbol{a}\in\delta_{\boldsymbol{X}}^+(v)\\ \boldsymbol{a}=(v[t],w[t'])}} t\, x_{\boldsymbol{a}} &\leq (T-1) \sum_{v\in\boldsymbol{V}(v)} \sum_{\boldsymbol{a}\in\delta_{\boldsymbol{X}}^+(v)} x_{\boldsymbol{a}}\\ &= (T-1) \sum_{\boldsymbol{a}\in\delta_{\boldsymbol{X}}^+(\boldsymbol{V}(v))} x_{\boldsymbol{a}}\\ &= (T-1),\end{aligned}$$

where for the last equality we have applied Proposition 3.2.

❑ **Proof of the constraints** (2.6a)**.** Our next claim is that $p_a \in [0,1]$ for all $a \in A$. Let $a \in A$. From Definition (3.8) it follows that $p = P\left(\begin{smallmatrix} x\\ y\end{smallmatrix}\right)$. If $a \in X$, this gives $p_a = \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\, x_{\boldsymbol{a}}$. Since $P_{a,\boldsymbol{a}} \in \{0,1\}$, we have

$$p_a = \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\, x_{\boldsymbol{a}} \leq \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} x_{\boldsymbol{a}} = 1,$$

where the last equality follows from the proof of Proposition 3.2. By Remark 3.8, the same conclusion can be drawn for the case $a \in A\backslash X$, where

$$p_a = \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\, y_{\boldsymbol{a}} \leq \sum_{\boldsymbol{a}\in\boldsymbol{A}(a)} y_{\boldsymbol{a}} = 1.$$

Finally, as the vectors x, y and the matrix P are non-negative, we can conclude $p_a \geq 0$.

We have thus proved that $\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } \textit{pro}) \subseteq \mathcal{P}_{\mathrm{LP}}(\text{PESP } \textit{box})$. What is left is the second assertion, i. e. $\mathcal{P}_{\mathrm{IP}}(\text{X-PESP } \textit{pro}) = \mathcal{P}_{\mathrm{IP}}(\text{PESP } \textit{box})$.

First, observe that if we apply the transformation ψ on an integer vector $(x,y) \in \mathbb{Z}^{\boldsymbol{X}} \times \mathbb{Z}^{\boldsymbol{A}\setminus\boldsymbol{X}}$, the result is an integer vector $\psi((x,y)) \in \mathbb{Z}^V \times \mathbb{Z}^A$. This is because ψ is defined by a matrix that contains only integer entries. Hence, it is sufficient to show that for every $(\pi, p) \in \mathcal{P}_{\mathrm{IP}}(\text{PESP } box)$, there exists a vector $(x,y) \in \mathcal{P}_{\mathrm{IP}}(\text{X-PESP } pro)$ such that $(\pi, p) = \psi((x,y))$.

Let $(\pi, p) \in \mathcal{P}_{\mathrm{IP}}(\text{PESP } box)$. Then, define the vector x by

$$x_{\boldsymbol{a}} = \begin{cases} 1, & \text{if } \boldsymbol{a} = (v[\pi_v], w[\pi_w]) \in \boldsymbol{A}(a) \\ 0, & \text{otherwise,} \end{cases}$$

for all $a = (v,w) \in X$ and set the vector y to

$$y_{\boldsymbol{a}} = \begin{cases} 1, & \text{if } \boldsymbol{a} = (v[\pi_v], w[\pi_w]) \in \boldsymbol{A}(a) \\ 0, & \text{otherwise,} \end{cases}$$

for all $a = (v,w) \in A\setminus X$. Denote by $\Pi_{v,\cdot}$ the row of the matrix Π belonging to the event v. Then, it follows that

$$\Pi_{v,\cdot} \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right) = \sum_{v[t] \in \boldsymbol{V}(v)} \Big(t \sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(v[t])} x_{\boldsymbol{a}}\Big) = \pi_v \sum_{\boldsymbol{a} \in \delta^+_{\boldsymbol{X}}(v[\pi_v])} x_{\boldsymbol{a}} = \pi_v,$$

for all $v \in V$. Now, let $a \in X$ and denote by $\boldsymbol{b} \in \boldsymbol{A}(a)$ the unique arc with $x_{\boldsymbol{b}} = 1$. If we write $P_{a,\cdot}$ for the row of the activity a in the matrix P, then

$$P_{a,\cdot} \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right) = \sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} P_{a,\boldsymbol{a}}\, x_{\boldsymbol{a}} = P_{a,\boldsymbol{b}}\, x_{\boldsymbol{b}} = P_{a,\boldsymbol{b}} = p_a,$$

where the last equality follows from the definition of the matrix P. The same conclusion can be drawn for $a \in A\setminus X$. ■

Remark 3.14 In general, the polyhedrons $\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } pro)$ and $\mathcal{P}_{\mathrm{LP}}(\text{PESP } box)$ are *not* equal. There exist PESP instances for which we can find solutions in the polyhedron $\mathcal{P}_{\mathrm{LP}}(\text{PESP } box)$ that have no counterpart in $\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } pro)$. Consider, for example, the instance in Figure 3.6. Suppose the period time $T = 5$. An easy computation shows that the vector π defined by $\pi_{\mathit{dep}(L,A)} = 4$, $\pi_{\mathit{arr}(L,B)} = 3$, $\pi_{\mathit{dep}(\overline{L},B)} = 2$, and $\pi_{\mathit{arr}(\overline{L},A)} = 1$ together with the vector $p \equiv 1/5$ is feasible to the LP relaxation of PESP *box* (2.6). However, every feasible solution to the LP relaxation of X-PESP *pro* (3.5) is a convex combination of the cycles in the displayed expanded event-activity graph. Hence, there is no corresponding solution in $\mathcal{P}_{\mathrm{LP}}(\text{X-PESP } pro)$.

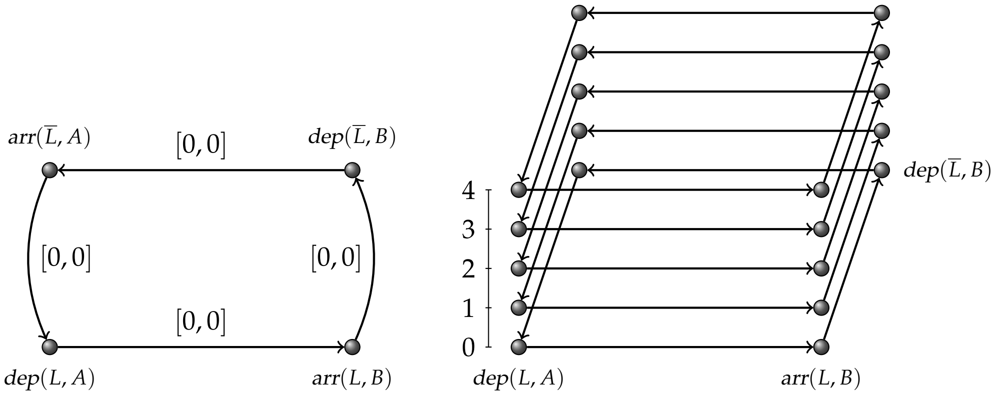
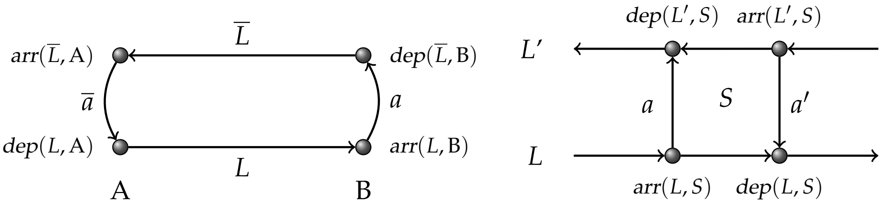


Figure 3.6 Illustration of Remark 3.14.

The polyhedron $\mathcal{P}_{\text{LP}}(\text{X-PESP } pro)$ is, in general, also no proper subset of the polyhedron $\mathcal{P}_{\text{LP}}(\text{PESP } box)$. To see this, consider again the PESP instance with the belonging expanded event-activity graph in Figure 3.6 and suppose now the period time $T = 1$. In this case, the event-activity graph and the expanded event-activity graph are identical. It follows that $\pi \equiv 0$, $p \equiv 0$ is the only feasible solution to the LP relaxation of PESP *box* (2.6). It corresponds to the only solution $x \equiv 1$ in $\mathcal{P}_{\text{LP}}(\text{X-PESP } pro)$ (note that $A \backslash X = \emptyset$). That is, both polyhedrons contain exactly one point.

3.2.1 Strong Valid Equalities for the X-PESP Formulations



(a) Line cycle with fixed dwell times.

(b) Crosswise transfer relation of the lines L and L' at station S.

Figure 3.7 Structures in the event-activity graph that yield valid equalities for the X-PESP formulations.

We now show how to derive a class of equalities that are valid for the above X-PESP formulations in case the running and dwell times of all lines are fixed. First, we consider two opposite directed lines L and $\overline{L}$ operated between the stations A and B (see Figure 3.7(a)). Let ℓ_{AB} denote the minimum travel time from A to B, i. e. the sum of the lower bounds of the trip and dwell activities of line L. The crucial observation is that every feasible timetable satisfies

$$\begin{aligned} \pi_{arr(\overline{L},\mathrm{A})} &= (\pi_{dep(\overline{L},\mathrm{B})} + \ell_{\mathrm{BA}}) \bmod T, \text{ and} \\ \pi_{dep(L,\mathrm{A})} &= (\pi_{arr(L,\mathrm{B})} - \ell_{\mathrm{AB}}) \bmod T. \end{aligned}$$

We now profit from expanding the event-activity graph. Let $\boldsymbol{a} \in \boldsymbol{A}(a)$ denote a turnaround arc of line L with $\boldsymbol{a} = (v[t_v], w[t_w])$. If there exists a turnaround arc $\overline{\boldsymbol{a}} = (\overline{v}[t_{\overline{v}}], \overline{w}[t_{\overline{w}}]) \in \boldsymbol{A}(\overline{a})$ of line $\overline{L}$ with $t_{\overline{v}} = (t_w + \ell_{\mathrm{BA}}) \bmod T$ and $t_{\overline{w}} = (t_v - \ell_{\mathrm{AB}}) \bmod T$, then it follows that $x_{\overline{\boldsymbol{a}}} = 1$ if and only if $x_{\boldsymbol{a}} = 1$. Thus, we can conclude that $x_{\boldsymbol{a}} - x_{\overline{\boldsymbol{a}}} = 0$ is a valid equality for X-PESP *basic* (3.1) and its improved versions X-PESP *plus* (3.2), and X-PESP *pro* (3.5). This also shows that if the turnaround arc $\overline{\boldsymbol{a}}$ does not exist, we must have $x_{\boldsymbol{a}} = 0$, and hence the arc $\boldsymbol{a}$ can be removed from the expanded event-activity graph.

By the same method we can derive equalities of the above type for crosswise transfer relations of a pair of lines. Figure 3.7(b) illustrates the situation for two lines L and L' that meet at the station S. Here, we must have

$$\begin{aligned} \pi_{arr(L',S)} &= (\pi_{dep(L',S)} - \ell_{dwell(L',S)}) \bmod T, \text{ and} \\ \pi_{dep(L,S)} &= (\pi_{arr(L,S)} + \ell_{dwell(L,S)}) \bmod T, \end{aligned}$$

and hence, if $\boldsymbol{a} = (v[t_v], w[t_w]) \in \boldsymbol{A}(a)$ and $\overline{\boldsymbol{a}} = (\overline{v}[t_{\overline{v}}], \overline{w}[t_{\overline{w}}]) \in \boldsymbol{A}(\overline{a})$ are two transfer arcs such that $t_{\overline{v}} = (t_v + \ell_{dwell(L,S)}) \bmod T$ and $t_{\overline{w}} = (t_w - \ell_{dwell(L',S)}) \bmod T$, we may conclude $x_{\boldsymbol{a}} - x_{\overline{\boldsymbol{a}}} = 0$.

Finally, let us briefly investigate the setting in terms of periodic tensions. According to the cycle periodicity property, we have $\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T z_C$ for every cycle C of the event-activity graph. Of course, this also applies to line cycles and those cycles resulting from crosswise transfers. Hence, if C denotes the cycle in Figure 3.7(a), then it follows that $\ell_{\mathrm{AB}} + x_a + \ell_{\mathrm{BA}} + x_{\overline{a}} = T z_C$ for some integer z_C. Now, recall that the cycle inequalities provide bounds on the variable z_C. So, if z_C can be restricted to a single value, we obtain the equation $x_a + x_{\overline{a}} = T z_C - x_{\mathrm{AB}} - x_{\mathrm{BA}}$, where only the tensions x_a and $x_{\overline{a}}$ are unknown. However, in general the variable z_C may take more than one value, as is shown in Example 2.21.

3.3 Speedup Techniques

We are interested in a direct solution of the X-PESP formulations by mixed integer programming. For this purpose we present two approaches that can support this task. The first aims at strengthening the LP relaxation to obtain improved lower bounds. The second is a special branching strategy that can lead to a more balanced division when applying branch-and-bound.

3.3.1 Event Fixation

The raw X-PESP formulations have the problem that their LP relaxations allow for solutions with zero cost. From these we obtain no more than trivial bounds on the optimal solution value of the integer programs. We can certainly assume that every integer solution has a non-zero value, since otherwise we cannot improve the LP bound.

Recall from Section 2.2.3 that, due to the redundancy in the PESP solution space, we can fix the timing of a single event a priori. We call this an *event fixation*. The PESP formulations cannot benefit from such a fixation, as is immediately seen from the proof of Lemma 2.7. For X-PESP, however, this is the way to strengthen the LP relaxation.

The X-PESP formulations are defined on arc variables. For this reason, the fixation of an event has to be done indirectly. Every event $v \in V$ is related to a trip activity, where there are two possible scenarios depending if v is an arrival event or a departure event. Since the following argumentation works for both cases, we may assume the latter and denote the trip arc by $a = (v, w)$. If we wish to fix the event v to the timing t, we have to impose the condition

$$\sum_{\substack{\boldsymbol{a} \in \boldsymbol{A}(a) \\ \boldsymbol{a} = (v[t], w[t'])}} x_{\boldsymbol{a}} = 1.$$

The effect of an event fixation is that no LP solution is allowed to select arcs that are adjacent to nodes in the set $\boldsymbol{V}(v) \setminus \{v[t]\}$. Hence, we avoid solutions that only select zero cost arcs, as is the case for the solution constructed in the proof of Lemma 3.10. Thus, the event fixation can be interpreted as a kind of symmetry breaking for the linear program. The expected increase in the objective value depends on the weight of the activities related to the fixed event. Therefore, we may choose the event $v^* \in V$ with the property that

$$v^* = \arg\max_{v \in V} \sum_{a \in \delta(v)} w_a.$$

The departure event of a high frequented line at some major transfer station may be a proper candidate.

Now, suppose the running times as well as the dwell times of all lines are fixed. Then, by fixing an event of line we automatically fix all events of that line. In this case, we speak of a *line fixation*. According to our remarks on the event fixation, it seems reasonable to fix the line $L^* \in \mathcal{L}$ with

$$L^* = \arg\max_{L \in \mathcal{L}} \sum_{v \in V(L)} \sum_{a \in \delta(v)} w_a, \tag{3.9}$$

where $V(L) \subseteq V$ denotes the set of events belonging to line L. We call $w(L) = \sum_{v \in V(L)} \sum_{a \in \delta(v)} w_a$ the *weight* of line L.

Figure 3.8 illustrates the effect of a line fixation by example. Contrary to a solution of the raw LP relaxation, the displayed solution involves non-zero transfer waiting times for all transfer stations along the route of the fixed line and some further stations.

3.3.2 Constraint Branching

The X-PESP formulations may be solved with a branch-and-bound algorithm. In this section we review two possible branching schemes with respect to the expected balance in the branch-and-bound tree. We will assume that the algorithm uses the linear programming relaxation.

Suppose in an LP solution to one of the X-PESP formulations we have $0 < x_{\boldsymbol{a}} < 1$ for some arc $\boldsymbol{a} = (v[t], w[t']) \in \boldsymbol{X}$. Such a solution is infeasible to the integer program. The same applies to fractional y variables. *Variable branching* yields two branches, one with $x_{\boldsymbol{a}} = 1$, and another with $x_{\boldsymbol{a}} = 0$. Setting $x_{\boldsymbol{a}} = 1$ implies that the timings of the events v and w are fixed to t and t', respectively. From our remarks on event fixation we know that this branch is likely to exclude many solutions that are integer infeasible. In the $x_{\boldsymbol{a}} = 0$ branch we only forbid the timing t for the event v and the timing t' for the event w'. Therefore, the feasibility region of the this branch is nearly the same as that of its father. In consequence, we have an imbalance between both branches which lessens the progress in the increase of the lower bound.

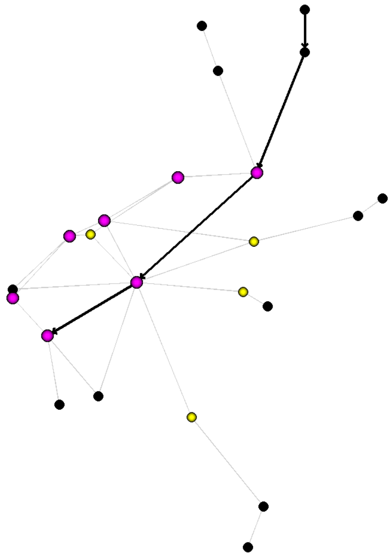


Figure 3.8 Transfers with waiting time for an X-PESP LP solution with line fixation. Black arcs mark the route of the fixed line. Magenta nodes indicate stations with non-zero transfer waiting time and yellow nodes those with zero transfer waiting time. Black nodes symbolize stations without transfers.

The X-PESP formulations ensure that for every activity $a \in A$, exactly one arc from the set $\boldsymbol{A}(a)$ is chosen. That is, every feasible solution satisfies the constraints

$$\sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} x_{\boldsymbol{a}} = 1 \qquad \text{for all } a \in X, \text{ and} \tag{3.10}$$

$$\sum_{\boldsymbol{a} \in \boldsymbol{A}(a)} y_{\boldsymbol{a}} = 1 \qquad \text{for all } a \in A \backslash X. \tag{3.11}$$

These type of constraints are known as *generalized upper bound constraints*, and allow for a special branching strategy called *constraint branching* (or *GUB branching*). Recall the fractional variable $x_{\boldsymbol{a}}$ which belongs to some activity $a \in X$. By the constraints (3.10), for every subset $\boldsymbol{B}$ of $\boldsymbol{A}(a)$, an integer solution either selects an arc from the set $\boldsymbol{B}$ or from the set $\boldsymbol{A}(a) \backslash \boldsymbol{B}$. This gives the idea for the branching rule. For the one branch we require $\sum_{\boldsymbol{a} \in \boldsymbol{B}} x_{\boldsymbol{a}} = 0$ and for the other branch $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a) \backslash \boldsymbol{B}} x_{\boldsymbol{a}} = 0$. For abbreviation, we say "branching on an activity" instead of "branching on the constraint of an activity".

We can apply the branching rule for every subset $\boldsymbol{B}$ for which $\sum_{\boldsymbol{a} \in \boldsymbol{B}} x_{\boldsymbol{a}} > 0$ and $\sum_{\boldsymbol{a} \in \boldsymbol{A}(a) \backslash \boldsymbol{B}} x_{\boldsymbol{a}} > 0$ in the current LP solution. However, we suggest to choose $\boldsymbol{B}$ by the following method. Suppose the arcs in the set $\boldsymbol{A}(a)$ are ordered according to the time values of their start nodes. That means if $\boldsymbol{A}(a) = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$, then for any two arcs $a_i = (v[t_i], w[t'_i])$ and $a_j = (v[t_j], w[t'_j])$ in $\boldsymbol{A}(a)$ with $i < j$, we have $t_i \leq t_j$. Then define the subset $\boldsymbol{B}$ as proposed by Beale and Tomlin (1970) for branching on special-ordered-set constraints. Compute the index $j = \lfloor \sum_{i=1}^{k} i\, x_{\boldsymbol{a}_i} \rfloor$ and set $\boldsymbol{B} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_j\}$.

It follows that in the branch with $\sum_{\boldsymbol{a} \in \boldsymbol{B}} x_{\boldsymbol{a}} = 0$, the event v is restricted to the timings $\{t_j, \ldots, T-1\}$. In the other branch only the timings $\{0, \ldots, t_j\}$ are allowed. In all cases where $t_j \neq 0$ and $t_j \neq T-1$ the division of the feasibility region may be better than by variable branching.

If more than one arc $\boldsymbol{a} \in \boldsymbol{A}$ is chosen fractionally, we propose to branch on the activity

$$a^* = \arg \max_{\substack{b=(v,w)\in A \\ \boldsymbol{A}(b) \cap \boldsymbol{A}_{frac} \neq \emptyset}} \sum_{a \in \delta(v)} w_a,$$

where $\boldsymbol{A}_{frac}$ denotes the set of arcs with fractional value. The reason for this choice is that constraint branching for an activity $a = (v, w)$ is comparable with a partial fixation of the event v.

Branching on an activity $a = (v, w)$ may be less effective if the timings of the events v and w are already restricted by previous branchings on other activities. Therefore, it is conceivable to consider only a subset of activities for constraint branching. Such a selection can be obtained from the *line-activity graph* (see Figure 3.9). In this graph every line $L \in \mathcal{L}$ is represented by a node, and for every activity $a = (v, w)$ there is an arc that joins the nodes of the lines related to the events v and w. Thus, the graph may have parallel arcs. The arc weight for an activity $a = (v, w)$ is set to $\sum_{b \in \delta(v)} w_b$. A possible subset of promising activities is then obtained by determining a maximum matching or a maximum spanning tree in the line-activity graph.

For a more general treatment of branch-and-bound algorithms and branching schemes we refer the reader to Nemhauser and Wolsey (1999).

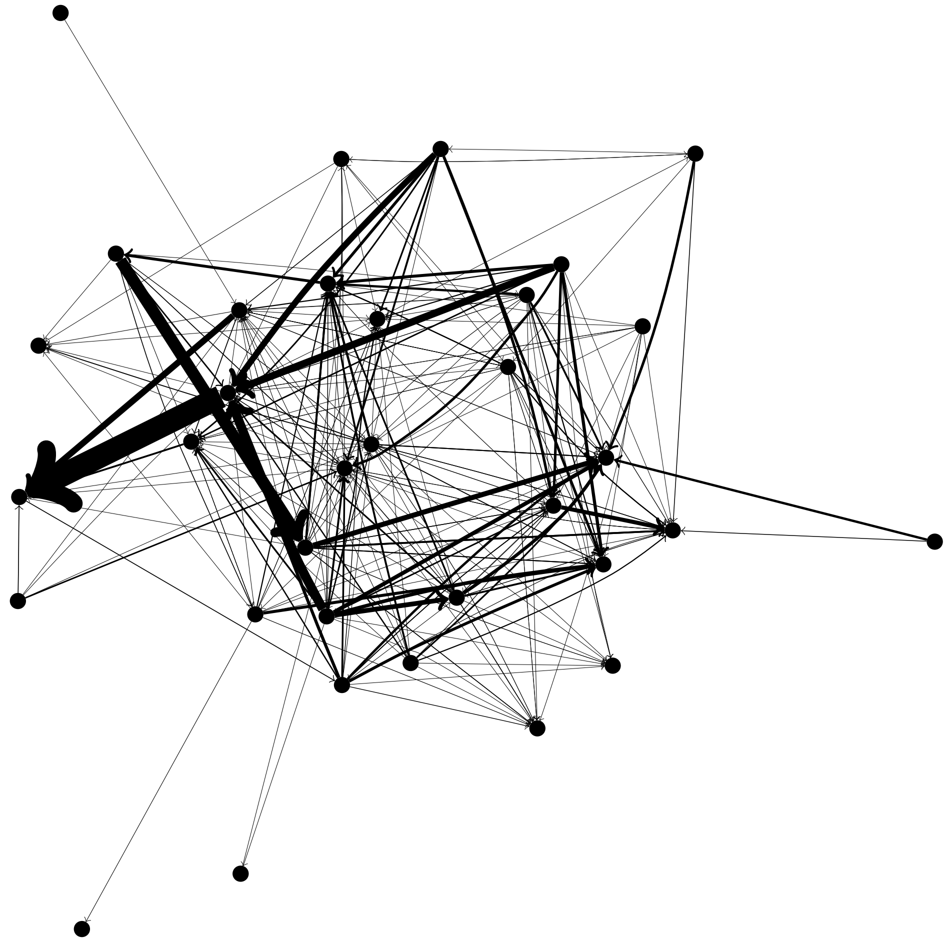


Figure 3.9 Line-activity graph of a sample instance.

4

Re-timetabling

There is a general problem concerning the computation of periodic timetables as indicated in Chapter 2. We required a fixed passenger flow in order to be able to prioritize the timetable requirements and, in particular, the transfer connections. The passenger flow, however, only reflects the passengers' travel behavior based on the service that is currently offered. If the new timetable is introduced, passengers may change their behavior in an unpredictable way. If this is the case, we have computed the timetable for a different traffic situation. The problem has been already pointed out by Lindner (2000). According to him, practitioners try to *simulate* the passengers' travel behavior to estimate the effect of a timetable.

We are interested in analyzing this effect from a mathematical point of view. To this end, we describe a procedure to route the passengers in the transportation network according to the computed timetable. Our goal is to obtain an updated passenger flow and to resolve the timetabling problem. This yields an iterative process, which we shall call *re-timetabling* (see Figure 4.1).

Under certain conditions, we can show that the total travel time in the transportation system is monotone decreasing when applying the re-timetabling procedure

4.1 Model Assumptions

As in the preceding chapters we want to compute a periodic timetable for a single demand period. Therefore, we require the conditions stated for timetabling in Section 4.1. In addition, we need a set of traffic zones $\mathcal{Z}$ and for every traffic zone $Z \in \mathcal{Z}$ a set of *neighboring stations* $\mathcal{S}(Z)$ where journeys to and from the traffic zone Z end. Furthermore, we require an OD-matrix M that specifies the estimated travel demand between all traffic zones during the demand period. Recall that the travel demand is interpreted as the number of passengers that want to travel between two traffic zones during the demand period. In the following, an ordered pair of traffic zones (Z, Z') is called *OD-pair* and we denote by $M_{Z,Z'}$ the related travel demand. If there is travel demand between two traffic zones, we will assume that there exists a possible connection by public transport.

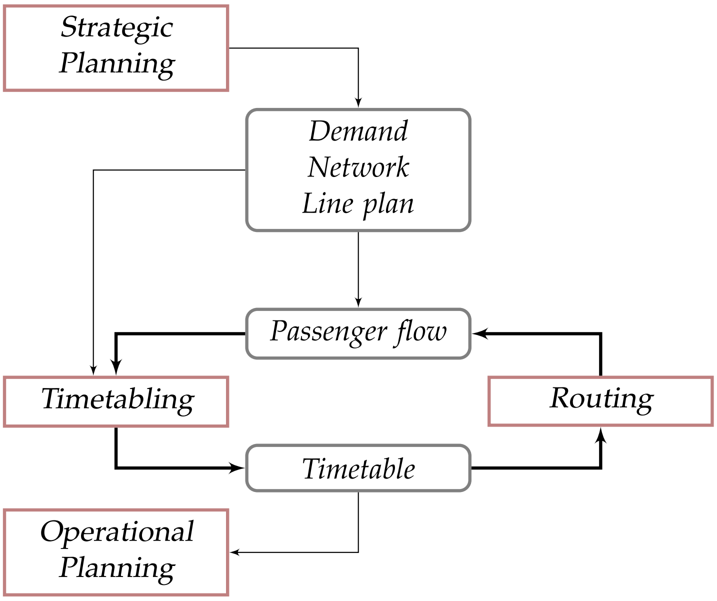


Figure 4.1 Re-timetabling. Iteratively compute a timetable and the resulting passenger flow.

Furthermore, it is required that the travel demand is not affected by the timetable, and we ignore service gaps that occur at the beginning and end of the operating period of a line. Actually, the last constraint can be dropped, but that would result in a more complex presentation.
Finally, let us assume a station specific minimum transfer time, denoted by $\ell_{trans(S)}$, for every station $S \in \mathcal{S}$.

4.2 The Timetable Graph

The aim of this section is to introduce the *timetable graph*, which is the foundation of our passenger flow computation. In each re-timetabling iteration this graph is constructed anew based on the current timetable. A similar graph has previously been used by Müller-Hannemann *et al.* (2007) for timetable information problems.

We start by defining the basic node set of the timetable graph (see Table 4.1 and Figure 4.2). Let T_L stand for the period time of line $L \in \mathcal{L}$ and write T_{lcm} for the least common multiple of all occurring period times. We introduce the notation $t_{arr(L,S)} \in \{0, \ldots, T_L - 1\}$ for the arrival time and $t_{dep(L,S)} \in \{0, \ldots, T_L - 1\}$ for the departure time of line L at station S according to the timetable. Let $L \in \mathcal{L}$, and define $m_L = T_{lcm}/T_L$. Then, for every station S on the route of line L, and time value $t = t_{arr(L,S)} + i\,T_L$ with $i = 0, \dots, m_L - 1$, there exists an *arrival node* $v[t]$ in the timetable graph. In addition, there is a *departure node* $v[t]$ for every time value $t = t_{dep(L,S)} + i\,T_L$ with $i = 0, \dots, m_L - 1$. Note that arrival and departure nodes are line, station, and time specific.

Our next goal is to model trip and dwell activities. Suppose S and S' are two consecutive stations on the route of line L. Then, a *trip arc* connects every departure node $v[t]$ of line L at station S with an arrival node $w[t']$ of the same line at station S'. With $t_{trip(L,S,S')}$ being the running time of line L between the stations S and S' according to the timetable, the trip arc $(v[t], w[t'])$ has to satisfy $(t' - t) \bmod T_{lcm} = t_{trip(L,S,S')}$ (see Figure 4.2). The weight of the arc is set to $t_{trip(L,S,S')}$.

Similarly, if S denotes a station on the route of line L, then a *dwell arc* connects every arrival node $v[t]$ of line L at station S with a departure node $w[t']$ of line L at the same station. The dwell arc $(v[t], w[t'])$ has to satisfy $(t' - t) \bmod T_{lcm} = t_{dwell(L,S)}$, where $t_{dwell(L,S)}$ stands for the dwell time of line L at station S according to the timetable. The arc weight is equal to $t_{dwell(L,S)}$.

Table 4.1 Partial timetable of the Berlin subway line U4 Nollendorf Platz → Innsbrucker Platz valid from 18 April to 1 November 2008.

Station	Arrival	Departure
Nollendorf Platz		06:02
Viktoria-Luise Platz	06:04	06:04
Bayrischer Platz	06:05	06:05
Rathaus Schöneberg	06:07	06:07
Innsbrucker Platz	06:08	

As in the timetable computation, we want to incorporate minimum transfer times in the passenger flow computation. Therefore, we cannot directly model a transfer connection by an arc between an arrival node of one line and a departure node of another line. We need to add further nodes and arcs to forbid transfers that fall below the station specific minimum transfer time. Figure 4.3 illustrates the construction.

We duplicate every departure node of a station to obtain a new *transfer node* with the same time value. An auxiliary arc of zero weight connects the transfer node with its related departure node. A *transfer arc* joins each arrival node $v[t]$ with the "earliest possible" transfer node $w[t']$ that satisfies $(t' - t) \bmod T_{lcm} \geq \ell_{trans(S)}$. By "earliest possible" we mean that $(t' - t) \bmod T_{lcm}$ is as small as possible. The weight of the transfer arc is set to $(t' - t) \bmod T_{lcm}$.

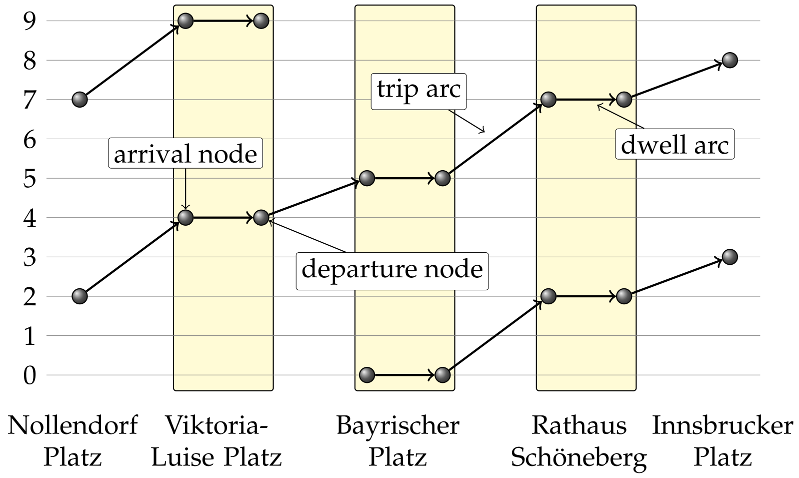


Figure 4.2 Basic elements of the timetable graph on example of the Berlin subway line U4.

Waiting at a station is modeled by *waiting arcs* between transfer nodes (see Figure 4.3). Suppose $v_1, \ldots, v_k$ are the transfer nodes of some station ordered according to their time values. Then, the timetable graph contains the waiting arcs (v_i, v_{i+1}) for $1 \leq i \leq k-1$ and (v_k, v_1).

What is left is to model the centroids of the traffic zones and their connection to the transportation network. For every traffic zone $Z \in \mathcal{Z}$ where traffic originates during the demand period and every time value t in the set $\{0, \ldots, T_{lcm} - 1\}$, there is an *origin node* $v[t]$ in the timetable graph. Moreover, if the traffic zone Z is also the destination of journeys during the demand period, we insert a *destination node*. Now, suppose $S \in \mathcal{S}(Z)$ is a neighboring station of traffic zone Z. Then, every origin node $v[t]$ of Z is connected to every departure node of station S. The weight of each arc $(v[t], w[t'])$ is comprised of the walking time $t_{walk(Z,S)}$ to station S and the difference t_{diff} between the arrival time t_{arr} at the station and the time value t' of the departure node $w[t']$. With

$$t_{arr} = (t + t_{walk(Z,S)}) \bmod T_{lcm} \qquad \text{and} \qquad t_{diff} = (t' - t_{arr}) \bmod T_{lcm},$$

the weight of the arc $(v[t'], w[t'])$ is given by $t_{walk(Z,S)} + t_{diff}$.
In addition, we connect the arrival nodes of every neighboring station $S \in \mathcal{S}(Z)$ with the destination node of the traffic zone Z (if any). The weight of the connecting arc is defined by the walking time from station S to the centroid of traffic zone Z.

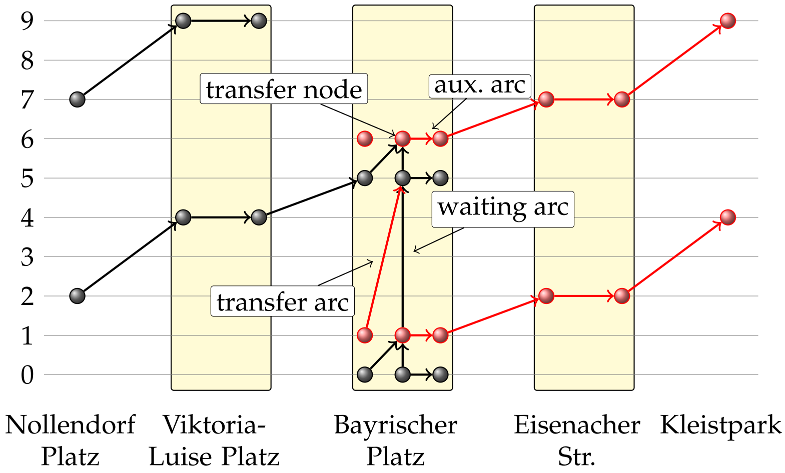


Figure 4.3 Transfer connection between the Berlin subway lines U4 (black) and U7 (red) at Bayrischer Platz. The minimum transfer time is 1 minute. Dwell arcs at Bayrischer Platz and arcs exceeding the interval $[0, 9]$ were omitted.

A path in the timetable graph connecting an origin node with a destination node is referred to as *passenger path*.

4.3 Passenger Flow Computation

The passenger flow computation as part of our re-timetabling procedure is performed in the timetable graph defined in the section above. The goal is to determine a passenger flow in the transportation network which is then used as input for the re-computation of the timetable. The main parts of the method may be summarized as follows:

1. We first compute for every OD-pair (Z, Z') with positive travel demand $M_{Z,Z'}$ a set of shortest passenger paths $\mathcal{P}(Z, Z')$.
2. Then, for every OD-pair (Z, Z'), we distribute the travel demand $M_{Z,Z'}$ over the passenger paths in the set $\mathcal{P}(Z, Z')$.

The passenger flow is then given by the set of passenger paths and the passenger distribution.

Computing the set of shortest passenger paths

Let us consider an OD-pair (Z, Z') with positive travel demand $M_{Z,Z'}$. Our purpose is to determine for every starting time in $\{0, \ldots, T_{lcm} - 1\}$ the shortest travel connections from traffic zone Z to traffic zone Z'. In the timetable graph every such travel connection corresponds to a certain passenger path. Hence, we have to compute for every origin node of traffic zone Z, the shortest passenger paths to the destination node of traffic zone Z'. Since the arc weights in the timetable graph are nonnegative, we may use Dijkstras algorithm to this end. This yields a set of passenger paths $\mathcal{P}(Z, Z')$ for the OD-pair (Z, Z').

Distributing the passengers

Having determined the shortest passenger paths for every OD-pair (Z, Z') with positive travel demand, we define the *passenger rate* of every passenger path $p \in \mathcal{P}(Z, Z')$. That is the fraction of passengers which we suppose to use path p. Here, the following problem arises. The OD-matrix only specifies the number of passengers traveling during the demand period. The data is imprecise in terms of exact starting times of the journeys. Therefore, we need to "guess" the passenger distribution. In practice, there are many factors that have an effect on the choice of the travel connection like

- the total travel time,
- the total transfer waiting time, or
- the total number of transfers required,

to name just a few examples. However, we do not have any information on how these factors are weighted against each other. For instance, how much additional travel time a passenger is willing to accept to save a single transfer. Furthermore, it is not known whether the passengers arrive at the station just in time. For these reasons, we equally distribute the passengers over all passengers paths in the set $\mathcal{P}(Z, Z')$. The passenger rate of every passenger path in $\mathcal{P}(Z, Z')$ is then given by $M_{Z,Z'}/|\mathcal{P}(Z, Z')|$. This value may be fractional.

The passenger paths in combination with the passenger distribution define the passenger flow.

Determining information for the timetable computation

In the following we describe how to obtain the information that is needed for the timetable computation. This includes the number of passengers on trips, stops, and transfers during the demand period (see Section 2.1).

We define a passenger rate for every arc in the timetable graph. That is the sum of the passenger rates over all passenger paths containing this arc.
Let $L \in \mathcal{L}$, and denote by S and S' two consecutive stations on the route of line L.

First, we need to know the number of passengers who are involved in trips of line L between S and S' during the demand period. This number is given as the sum of the passenger rates over all trip arcs of line L between S and S', and we denote it by $F(L, S, S')$.
Second, we require the number of passengers involved in stops of line L at any station S on its route during the demand period. This number is obtained by accumulating the passenger rates of all dwell arcs of line L at station S, and we use the notation $F(L, S)$.
Third, we have to determine the number of passengers for every transfer connection in the transportation network. Recall that a transfer connection is given by a 3-tuple (L, L', S), where $L, L' \in \mathcal{L}$ and $S \in \mathcal{S}$. The transfers on a passenger path are easily identified, if we only consider the arrival and departure nodes on the path. If an arrival node and its succeeding departure node are related to different lines, this corresponds to a transfer. Suppose the arrival node belongs to line L and station S, and the departure node is related to line L'. Then, we add the passenger rate of the passenger path to the number of passengers using the travel connection (L, L', S). We do this for every passenger path.

Evaluating quality of a timetable
The passenger flow has several properties by which we can evaluate the quality of a timetable of the same problem instance. For every passenger path, the timetable implies a certain travel time. Summing over the travel times of all passenger paths weighted with the passenger rate yields the *total travel time*. Furthermore, every transfer on a passenger path involves a certain waiting time which depends on the timetable. The *total transfer waiting time* is given by the sum of the waiting times over all transfers weighted with number of passengers involved. Finally, the timetable that underlies the passenger flow can be characterized by the *total number of transfers* which is obtained by accumulating the number of passengers of all transfer connections.

4.4 The Re-timetabling Procedure

Algorithm 1 illustrates the re-timetabling procedure. For convenience, we let I stand for the information that is not necessary to understand the functionality. A detailed exposition of required data can be found in Section 2.1 and Section 4.1.

In each iteration i of the re-timetabling procedure, a timetable B^i is computed on the basis of the previous passenger flow F^{i-1} (step 4). This timetable is then used

Algorithm 1 Re-timetabling

```
Input: Passenger flow $F_0$, other timetabling input $I$
Output: Timetable $B$
1: $i \leftarrow 0$
2: repeat
3:     $i \leftarrow i + 1$
4:     $B^i \leftarrow$ computeTimetable($F^{i-1}$, $I$)
5:     $F^i \leftarrow$ computePassengerFlow($B^i$, $I$)
6: until $F^i = F^{i-1}$
7: $B \leftarrow B^i$
```

to determine the next passenger flow F^i (step 5). The algorithm terminates if the computed passenger flow is the same as in the previous iteration (step 6).

The re-timetabling procedure produces an alternating sequence of timetables and passenger flows $F_0 \rightarrow B_1 \rightarrow F_1 \rightarrow B_2 \rightarrow \ldots$. This raises several questions, for instance, whether the sequence converges, and how the passenger flow changes in each iteration. In the following, we discuss the second question and investigate the development of the passenger flow by means of the total travel time. It will turn out that, under certain conditions, this value is monotone decreasing.

The total travel time of a passenger flow is comprised of two parts, one that is independent of the timetable and another that is not. The timetable independent part is given as sum of the total access time, the total minimum running time, the total minimum dwell time, and the total minimum transfer time. Given a passenger flow F, these times are defined as follows.

Total access time: Sum of the walking time to the initial station and the walking time from the final station over all journeys during the demand period. We denote it by $t_{access(F)}$.

Total minimum running time: Sum of the minimum running times over all trips during the demand period. With the notation $(S, S') \in L$ for two consecutive stations on the route of line L, the total minimum running time $\ell_{trip(F)}$ is defined by

$$\ell_{trip(F)} = \sum_{L \in \mathcal{L}} \sum_{(S,S') \in L} F(L, S, S') \cdot \ell_{trip(L,S,S')},$$

where $\ell_{trip(L,S,S')}$ denotes the minimum running time of line L between the stations S and S'.

Total minimum dwell time: Sum of the minimum dwell times over all stops during the demand period. If we denote by $S \in L$ a station on the route of line L, then the total minimum dwell time $\ell_{dwell(F)}$ is given by

$$\ell_{dwell(F)} = \sum_{L \in \mathcal{L}} \sum_{S \in L} F(L, S) \cdot \ell_{dwell(L,S)},$$

where $\ell_{dwell(L,S)}$ denotes the minimum dwell time of line L at station S.

Total minimum transfer time: Sum of the minimum transfer times over all transfers during the demand period. Let $\mathcal{C}$ stand for the set of transfer connections. Then, the total minimum transfer time $\ell_{trans(F)}$ is defined by

$$\ell_{trans(F)} = \sum_{(L,L',S) \in \mathcal{C}} F(L, L', S) \cdot \ell_{trans(S)}.$$

The remaining part of the total travel time of the passenger flow depends on the timetable. This applies to the total initial waiting time and the total surplus time. For a passenger flow F and a timetable B of the same problem instance, these times are defined as follows.

Total initial waiting time: Sum of the initial waiting times over all journeys during the demand period. By the *initial waiting time*, written $t_{init(F,B)}$, we mean the time between the arrival at the initial station and the departure time of the desired line. For a fixed passenger distribution the total initial waiting time depends on the timetable.

Total surplus time: Total additional time for trips, stops, and transfers during the demand period (with respect to the minimum time durations). Each individual time is weighted with the number of passengers involved (according to the passenger flow F). We denote the total surplus time by $t_{plus(B,F)}$.

With the above notation, we have

$$t_{travel(F,B)} = \underbrace{t_{access(F)} + \ell_{trip(F)} + \ell_{dwell(F)} + \ell_{trans(F)}}_{\text{timetable independent}} + t_{plus(F,B)} + t_{init(F,B)}, \qquad (4.1)$$

for the total travel time of the passenger flow F subject to the timetable B.

In general, there is no relation between the total travel time of two succeeding re-timetabling iterations. The timetable independent part as well as the remaining part may change in an unpredictable way. We can, however, present two special cases where the total travel time is monotone decreasing.

From now on we make the assumption that the timetable B^i, computed in step 4 of the re-timetabling procedure, minimizes the total surplus time $t_{plus(F^{i-1},B^i)}$ based on the previous passenger flow F^{i-1}.

Just-in-time distribution
For the first scenario, we assume that the passengers always choose their travel connections a priori, maybe supported by a timetable information system, and reach the initial station just in time. As a consequence, we have no initial waiting time.
In the passenger flow computation, this can be realized if we only consider those passenger paths for the distribution that do not involve any waiting at the initial station. We call this the *just-in-time distribution*.

Theorem 4.1 Let $i \in \mathbb{N}$ and suppose the total initial waiting time is zero, then

$$t_{travel(F^{i-1},B^{i-1})} \geq t_{travel(F^i,B^i)}.$$

Proof. Let $i \in \mathbb{N}$. First, consider the passenger flow F^{i-1} and the timetables B^{i-1} and B^i. By assumption, the timetable B^i yields the minimum total surplus time for the passenger flow F^{i-1} while the timetable B^{i-1} may not. Thus, we have $t_{plus(F^{i-1},B^{i-1})} \geq t_{plus(F^{i-1},B^i)}$. According to equation (4.1) and the hypothesis of the theorem, this gives

$$t_{travel(F^{i-1},B^{i-1})} \geq t_{travel(F^{i-1},B^i)}. \tag{4.2}$$

Second, consider the timetable B^i and the passenger flows F^{i-1} and F^i. We have

$$t_{travel(F^{i-1},B^i)} \geq t_{travel(F^i,B^i)}, \tag{4.3}$$

because the passenger flow F^i is optimal for timetable B^i according to the total travel time while the passenger flow F^{i-1} is only feasible. Combining the inequalities (4.2) and (4.3) we deduce $t_{travel(F^{i-1},B^{i-1})} \geq t_{travel(F^{i-1},B^i)} \geq t_{travel(F^i,B^i)}$. ■

Equal distribution in consideration of the initial waiting time
In real life, it is certainly not realistic that all passengers are perfectly informed on the timetable and avoid any waiting at the initial station. For this reason, let us again assume that the passengers start their journeys equally distributed over $\{0, \ldots, T_{lcm} - 1\}$. We now investigate another condition which implies monotonicity for the total travel time. The idea is to incorporate the average initial waiting

time at any station in the timetable computation. Based on the equal distribution of the passengers, the timetable B^i in iteration i then minimize the sum of $t_{plus(F^{i-1},B^i)}$ and $t_{init(F^{i-1},B^i)}$.

In order to minimize the average initial waiting time it is necessary to synchronize the departure times of certain groups of lines. For every OD-pair (Z, Z') we need to know at which stations the passengers start there journey to traffic zone Z' and by which line. All lines departing at the same station $S \in \mathcal{S}(Z)$ make up a group of lines that has to be synchronized. In case all lines of that group have the same period time, this can be easily achieved by requiring a minimum headway time of T/n and a maximum headway time of $T - T/n$, where n denotes the number of lines in the group. Each constraint is weighted with the fraction of the passengers that depart at station S (with respect to the travel demand $M_{Z,Z'}$). If the lines have different period times, synchronization is more complicated. The problem is discussed in Brucker *et al.* (1990).

Theorem 4.2 Suppose $i \in \mathbb{N}$. If the timetable B^i minimizes the sum of the total surplus time $t_{plus(F^{i-1},B^i)}$ and the total initial waiting time $t_{init(F^{i-1},B^i)}$, then

$$t_{travel(F^i,B^i)} \geq t_{travel(F^{i+1},B^{i+1})}.$$

Proof. By the hypothesis of the theorem, it follows that

$$t_{plus(F^{i-1},B^{i-1})} + t_{init(F^{i-1},B^{i-1})} \geq t_{plus(F^{i-1},B^i)} + t_{init(F^{i-1},B^i)}$$

for $i \in \mathbb{N}$. This establishes equation (4.2) and we can apply the same argumentation as in the proof Theorem 4.1. ■

5 Computational Results

In Section 2.2 we presented a rich set of existing integer programming formulations for the Periodic Event Scheduling Problem. We now provide a preliminary computational study in which we compare these formulations with the new PESP formulations based on time discretization from Chapter 3. For a better distinction, we refer to the new formulations as X-PESP formulations.

It will turn out that the PESP formulations are currently state of the art. For smaller problem instances, however, the X-PESP formulations are competitive. This is particularly due to the great effect of event fixation, valid equalities, and constraint branching.

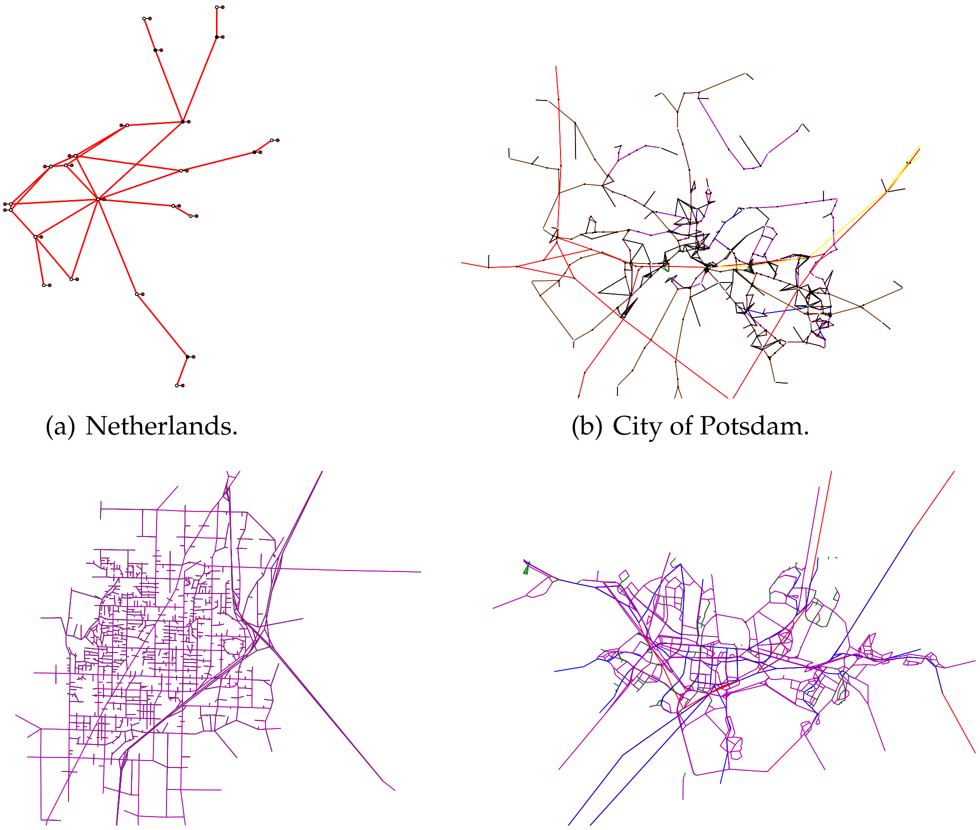
Finally, Section 5.4 is devoted to the re-timetabling procedure. On the basis of an own implementation, we confirm our theoretical results on the properties of the computed timetables (see Section 4.4).

5.1 Test Instances

Our computational experiments considered five real-life instances. Two of which are based on the Dutch InterCity network. The others are variants of the public transport networks of three major cities, namely, Potsdam, Lynnwood, and Karslruhe. Figure 5.1 shows the route diagrams for each instance. The data for the Netherlands and Potsdam network was provided by the Konrad-Zuse-Institut Berlin while the networks of Lynnwood and Karslruhe were obtained from the PTV AG. From now on we use the following symbols to denote the different instances.

`N`, `Ns`	InterCity network of the Netherlands
`P`	Public transport network of the city of Potsdam
`L`	Bus network of the city of Lynnwood
`K`	Public transport network of the city of Karlsruhe

In Table 5.1 we have compiled the key data of the problem instances. Of particular importance are the entries on the number of directed lines, transfer relations,



(a) Netherlands.

(b) City of Potsdam.

(c) City of Lynnwood.

(d) City of Karlsruhe.

Figure 5.1 Route diagrams of the test instances.

and turnaround relations. The smallest instance is the variant Ns of the Dutch InterCity network with only 18 directed lines. The most directed lines, transfer relations, and turnaround relations has the Karlsruhe instance K.

Table 5.1 Problem instances.

	N	Ns	P	L	K
Stations	23	23	262	157	463
Directed lines	30	18	38	34	96
Transfer relations	128	81	333	256	853
Turnaround relations	30	18	38	34	96

5.2 Test Environment

The computational experiments on the X-PESP formulations have been performed on a 3.2 GHz Pentium 4 with 2 GB main memory. The results were obtained with SCIP version 1.00.6, using CPLEX 11.0 as underlying LP solver. ZIMPL version 2.07 was used to generate the mixed-integer programming models. The PESP formulations were tested on a 2.13 GHz Core 2 6400 with 2 GB main memory, using CPLEX 11.0 as MIP solver. In both environments we defined a time limit of two hours and a memory limit of 512 MB.

5.3 Periodic Timetabling

The first part of our computational study concerns timetabling. The purpose was to compute period timetables for the various problem instances, where we considered identical frequencies for all lines. Furthermore, we assumed that the running times as well as the dwell times of the lines are fixed.

The timetables were supposed to respect minimum turnaround times and minimum transfer times. The objective was to minimize the sum of the transfer waiting time and the additional turnaround time over all transfer and turnaround relations. The priorization of the transfer relations was made according to a given passenger flow, while for the turnaround relations we used a fixed uniform weight.

We use the following definitions and notations for our computational results.

Gap — Primal-dual gap. With *pb* being the objective value of the incumbent solution and *db* being the best dual bound found, the Gap is defined by

$$\text{Gap} = \begin{cases} 0 & \text{if } |pb - db| \leq tol, \\ (pb - db) \;/\; |db| & \text{if } pb \cdot db > 0, \\ \infty & \text{otherwise,} \end{cases}$$

where *tol* denotes the default zero tolerance of the MIP solver. For convenience, we let ’–’ stand for infinity.

Root Gap — Primal-dual gap in the root node with respect to the objective value $pb^\star$ of the best solution that is known for the problem instance.

$$\text{Root Gap} = \begin{cases} 0 & \text{if } |pb^\star - db| \leq tol, \\ (pb^\star - db) \;/\; |db| & \text{if } pb^\star \cdot db > 0, \\ \infty & \text{otherwise.} \end{cases}$$

Again, we let ’–’ stand for infinity.

Time — Elapsed CPU time in seconds

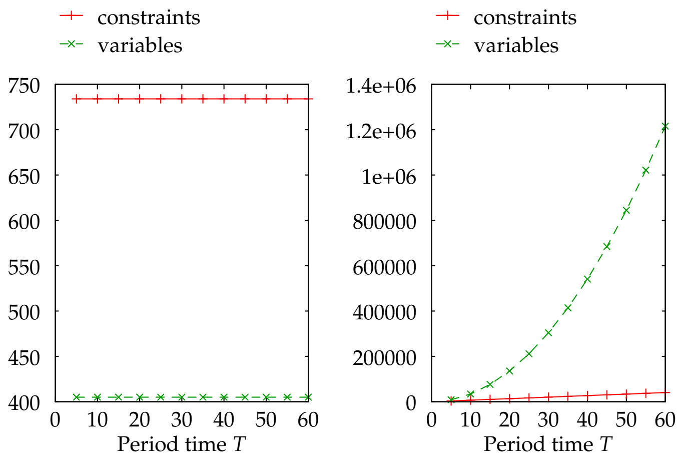
Nodes — Number of branching nodes

5.3.1 Formulation Sizes

The size of a formulation can be an important factor for its solvability. Therefore, Figure 5.2 compares the sizes of two representative PESP and X-PESP formulations for the instance P depending on the period time. Notice that SCIP and CPLEX contain a preprocessor for reducing the formulation size. The values were obtained after using the preprocessor.

In the comparison, PESP *box* (2.6) is the representative PESP formulation. For every period time considered the formulation is very small in size with respect to X-PESP *pro* (3.5). As typical for the PESP formulations, the period time has no effect on the number of variables and constraints, i. e., the size of the formulation is constant. By contrast, we can observe an explosive growth in the number of variables for the X-PESP *pro* (3.5) formulation. The reason lies in the time discretization of the event-activity graph that is based on the period time.

We conclude, that if we consider a larger set of timetable requirements, the size of the X-PESP formulations becomes critical for period times larger than 20.



(a) PESP *box* (2.6) (b) X-PESP *pro* (3.5)

Figure 5.2 Formulation sizes for the instance `P`.

5.3.2 Event Fixation and Valid Equalities for X-PESP

We now present the results on event fixation and the adding of valid equalities for the X-PESP formulations (cf. Section 3.3.1 and Section 3.2.1). Recall that we assumed fixed running and dwell times for the lines. Therefore, an event fixation is equal to a line fixation. For the test we consider X-PESP *pro* (3.5) as a representative X-PESP formulation. The tag 'L' means that we applied line fixation and by 'I' we indicate the use of valid equalities.

Figure 5.3 shows the root gaps resulting from both techniques for the instance `N` and different period times. The fixation of the line with the highest weight yields root gaps of under 185%. By adding valid equalities for every line cycle and crosswise transfer relation, the root gaps fall below 50%. To value the effect, note that the raw X-PESP *pro* (3.5) formulation on the instance `N` yields, except for the period time $T = 5$, root gaps of more than 1000%. It can be seen that the combination of line fixation and valid equalities is even more effective. We obtain root gaps of less than 25%.

In Figure 5.4 we illustrate, on example of the instance `P`, that the choice of the line fixation has a great influence on the resulting root gap. Every bar corresponds to a different line and the points indicate the weight of each line as defined in Section 3.3.1. We see that the line with the highest weight yields the smallest root gap.

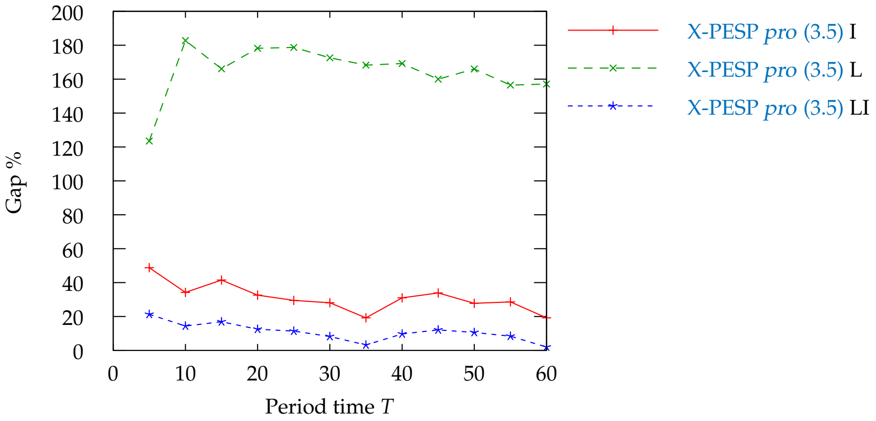


Figure 5.3 The effect of line fixation and valid inequalities on the root gap of X-PESP *pro* (3.5). The figure applies to the instance `N`.

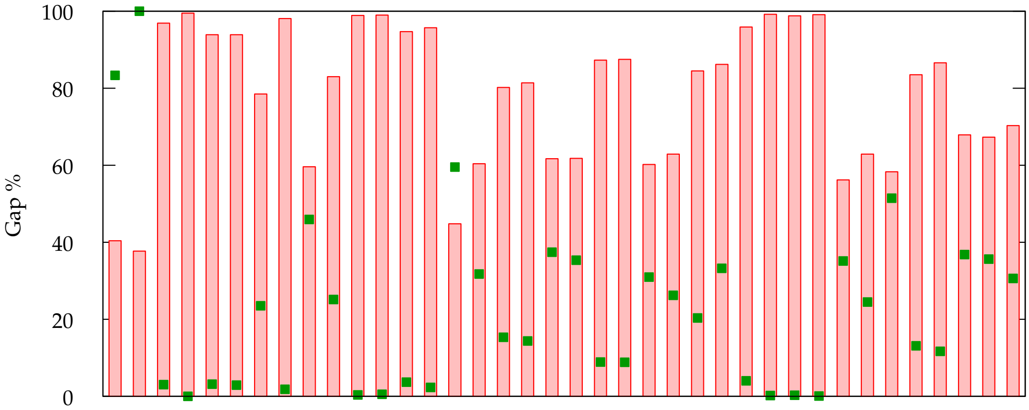


Figure 5.4 The effect of a line fixation depends on the weight of the line. The figure applies to X-PESP *pro* (3.5) LI on instance `P` with period time $T = 10$. The bars show the root gaps of different line fixations. The points indicate the weight of the lines.

Finally, we have compared the root gaps of X-PESP *pro* (3.5) LI and that of the PESP formulations from Section 2.2. Figure 5.5 shows the results for the instance N. The formulation PESP *y-box* (2.9) yields root gaps of under 30% for all period times. Only PESP x (2.11) is better, with root gaps below 20%. Our formulation X-PESP *pro* (3.5) with line fixation and valid inequalities behaves well and can outperform three of the five PESP formulations with root gaps of under 25%.

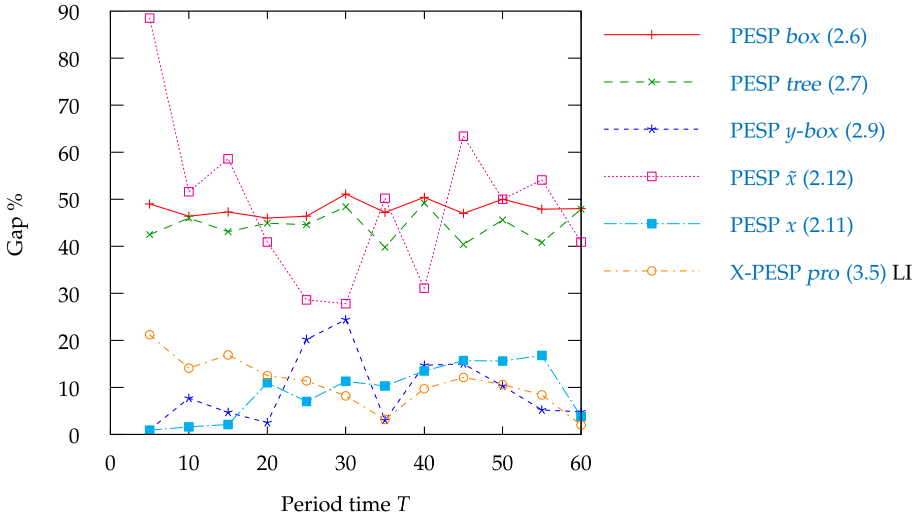


Figure 5.5 Root gaps of different PESP formulations. The figure applies to the instance N.

Constraint Branching for X-PESP

We have also investigated whether constraint branching for X-PESP can improve the solving performance. Our implementation corresponds to the description in Section 3.3.2. The branching candidates were obtained from a MAXIMUM SPANNING TREE of the line-activity graph. Furthermore, we predefined a branching order for the constraints based on the weights of the related activities. Our comparison considers the formulation X-PESP *pro* (3.5) with line fixation and valid equalities. The tag 'C' means that we have applied constraint branching.

Table 5.2 summarizes the results for all test instances and different period times. In order to clearly see the effect of the strategy, we deactivated all primal heuristics when solving the formulations with constraint branching. We can assert that by applying this technique more instances could be solved to optimality within the time limit. However, if the size of the formulation becomes too large, as is the case for the instances P, L, and K, without heuristics no feasible solutions were found. Figure 5.6 shows the branch-and-bound trees for the instance N with and without constraint branching. On closer inspection, we can see that the tree resulting from constraint branching is more balanced.

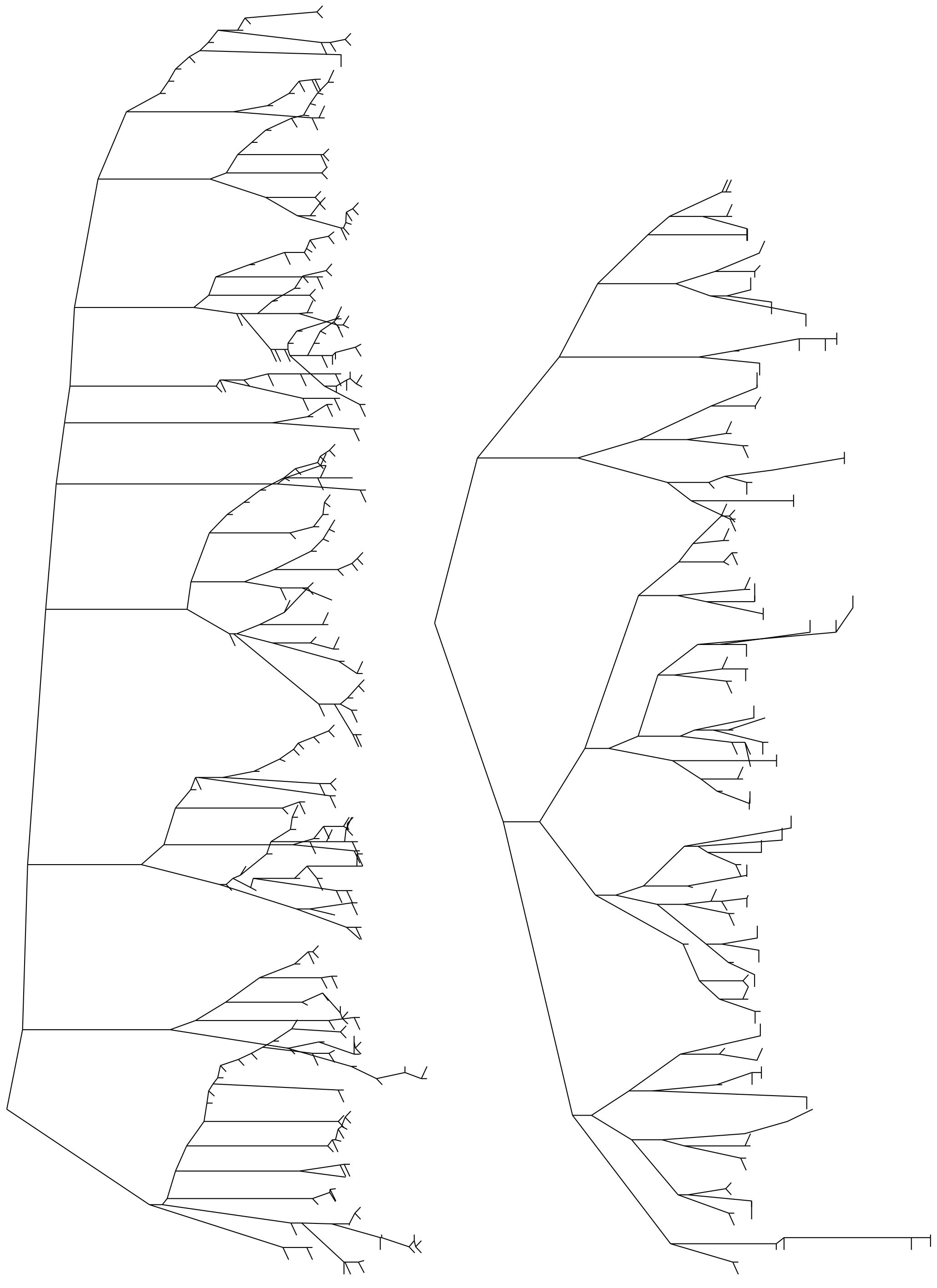
Table 5.2 Computational performance of X-PESP *pro* (3.5) LIC in comparison with X-PESP *pro* (3.5) LI.

	X-PESP *pro* (3.5) LI			X-PESP *pro* (3.5) LIC		
	Gap %	Nodes	Time	Gap %	Nodes	Time
N						
$T = 15$	0.0	874	951.8	0.0	230	154.0
$T = 20$	0.0	1394	2754.8	0.0	250	457.2
$T = 30$	2.5	407	7200.0	0.0	83	897.8
$T = 60$	1.8	12	7200.0	0.0	48	3001.5
Ns						
$T = 15$	0.0	21	59.1	0.0	29	15.6
$T = 20$	0.0	91	89.6	0.0	65	52.0
$T = 30$	0.0	1	60.5	0.0	12	94.9
$T = 60$	3.7	53	7200.0	0.0	106	5841.0
P						
$T = 15$	16.1	175	7200.0	0.0	429	3341.7
$T = 20$	34.9	31	7200.0	8.6	227	7200.0
$T = 30$	38.7	20	7200.0	–	37	7200.0
L						
$T = 15$	23.9	392	7200.0	7.6	1312	7200.0
$T = 20$	23.9	87	7200.0	9.2	299	7200.0
$T = 30$	34.6	17	7200.0	–	0	7.9
K						
$T = 15$	173.0	25	7200.0	–	110	7200.0
$T = 20$	–	18	7200.0	–	30	7200.0

5.3.3 Computational Performance of PESP and X-PESP Formulations

In Table 5.3 we give a comparison of the solving performance for the PESP formulations from Section 2.2. Except for the formulation PESP *y-box* (2.9), the instances N and Ns could be solved to optimality within the time limit. As already indicated by a computational study of Liebchen (2006), on rather small instances the formulation PESP $\tilde{x}$ (2.12) behaves best. We can also confirm that PESP *box* (2.6) and PESP *y-box* (2.9) have a poor performance. The formulations PESP *tree* (2.7) and PESP *x* (2.11) have the best overall performance.

It can be observed that the solution time does not depend on the period time. Furthermore, the root gap of a formulation gives no hint on the performance (cf. PESP *tree* (2.7) in Figure 5.5).



(a) X-PESP *pro* (3.5) LI (562 nodes).

(b) X-PESP *pro* (3.5) LIC (209 nodes).

Figure 5.6 Constraint branching and its effect on the branch-and-bound tree of X-PESP *pro* (3.5). The figure is subject to the instance N and period time $T = 10$. The depth of a node corresponds to its local dual bound or, in case of infeasibility, to that of its father.

Table 5.3 Computational performance of the PESP formulations.

	PESP *box* (2.6)		PESP *tree* (2.7)		PESP y-*box* (2.9)		PESP x (2.11)		PESP $\tilde{x}$ (2.12)	
	Gap %	Time	Gap %	Time	Gap %	Time	Gap %	Time	Gap %	Time
`N`										
$T = 15$	0.0	2014.0	0.0	90.0	6.9	7200.0	0.0	119.0	0.0	60.0
$T = 20$	0.0	1646.0	0.0	268.0	5.8	7200.0	0.0	126.0	0.0	74.0
$T = 30$	0.0	853.0	0.0	36.0	0.5	4008.0	0.0	21.0	0.0	12.0
$T = 60$	0.0	56.0	0.0	14.0	1.7	7200.0	0.0	6.0	0.0	9.0
`Ns`										
$T = 15$	0.0	33.0	0.0	9.0	0.4	7200.0	0.0	11.0	0.0	5.0
$T = 20$	0.0	34.0	0.0	12.0	0.0	258.0	0.0	9.0	0.0	8.0
$T = 30$	0.0	10.0	0.0	3.0	0.0	60.0	0.0	1.0	0.0	4.0
$T = 60$	0.0	45.0	0.0	30.0	0.0	916.0	0.0	7.0	0.0	7.0
`P`										
$T = 15$	6.0	7200.0	0.1	1900.0	16.2	7200.0	0.0	4678.0	0.0	5111.0
$T = 20$	5.2	7200.0	0.1	2189.0	13.9	7200.0	3.5	7200.0	2.4	2790.0
$T = 30$	5.4	7200.0	0.1	2743.0	13.6	7200.0	0.0	4820.0	3.4	7200.0
$T = 60$	7.2	7200.0	0.1	2669.0	13.0	6267.0	0.0	4852.0	0.0	4990.0

For a direct of comparison of our X-PESP formulation X-PESP *pro* (3.5) LIC with PESP *tree* (2.7) we restate the results for both formulations in Table 5.4. It is easily seen that for solving the PESP formulation much more branching nodes could be processed in less time with respect to the X-PESP formulation. This is due the smaller size of the PESP problems (see Figure 5.2). For the instances `N` and `Ns` the X-PESP formulation is on par with PESP *tree* (2.7) in terms of optimality within the time limit. Larger instances like `P`, however, could not be solved properly.

In the last part of our PESP formulation round-up let us consider the three variants of X-PESP, namely, X-PESP *basic* (3.1), X-PESP *plus* (3.2), and the formulation X-PESP *pro* (3.5). The reason why we have chosen the latter formulation for our study can be found in Table 5.5. We tested each formulation without applying further speedup techniques on all problem instances for different period times. It turned out that X-PESP *basic* (3.1) has the worst solving performance. In nearly all cases no feasible solution could be found within the time limit. For ten out of eleven problems the X-PESP *pro* (3.5) formulation yields the best gaps after two hours.

Table 5.4 Computational performance of X-PESP *pro* (3.5) LIC in comparison with PESP *tree* (2.7).

	PESP *tree* (2.7)			X-PESP *pro* (3.5) LIC		
	Gap %	Nodes	Time	Gap %	Nodes	Time
N						
$T = 15$	0.0	278162	90.0	0.0	230	154.0
$T = 20$	0.0	672164	268.0	0.0	250	457.2
$T = 30$	0.0	87793	36.0	0.0	83	897.8
$T = 60$	0.0	40483	14.0	0.0	48	3001.5
Ns						
$T = 15$	0.0	33490	9.0	0.0	29	15.6
$T = 20$	0.0	51225	12.0	0.0	65	52.0
$T = 30$	0.0	10127	3.0	0.0	12	94.9
$T = 60$	0.0	160665	30.0	0.0	106	5841.0
P						
$T = 15$	0.1	1922820	1900.0	0.0	429	3341.7
$T = 20$	0.1	2100666	2189.0	8.6	227	7200.0
$T = 30$	0.1	1929843	2743.0	–	37	7200.0
$T = 60$	0.1	1937384	2669.0	–	–	–

Table 5.5 Computational performance of the X-PESP formulations.

	X-PESP *basic* (3.1)		X-PESP *plus* (3.2)		X-PESP *pro* (3.5)	
	Gap %	Nodes	Gap %	Nodes	Gap %	Nodes
N						
$T = 15$	5232.9	64	513.7	470	222.7	1036
$T = 20$	–	29	2795.3	91	1169.0	228
$T = 30$	–	2	Large	53	3271.0	68
$T = 60$	–	1	–	22	–	16
Ns						
$T = 15$	1641.9	212	32.2	29703	0.0	28196
$T = 20$	6669.7	61	181.5	1198	73.6	5256
$T = 30$	–	15	1688.2	170	839.2	158
$T = 60$	–	1	–	19	–	35
P						
$T = 15$	–	18	2829.9	125	2021.7	118
$T = 20$	–	1	3808.6	64	4465.3	81
$T = 30$	–	1	–	44	–	44
L						
$T = 15$	Large	30	2162.4	134	1364.8	327
$T = 20$	–	4	Large	86	3668.5	90
$T = 30$	–	1	–	36	–	36
K						
$T = 15$	–	1	2687.7	44	2654.6	49
$T = 20$	–	1	–	24	–	17
$T = 30$	–	0	–	1	–	1

5.4 Re-timetabling

For our computational experiments on the re-timetabling procedure from Chapter 4, we implemented a passenger routing algorithm on the basis of the timetable graph. This algorithm supports two passenger distributions: the equal distribution (see Section 4.3) and the just-in-time distribution (see Section 4.4).

The timetabling part of our implementation is based on the X-PESP *pro* (3.5) formulation in combination with a standard MIP solver. For the computation we made the following assumptions. The lines were assumed to have identical frequencies. Running and dwell times of the lines were assumed to be fixed.

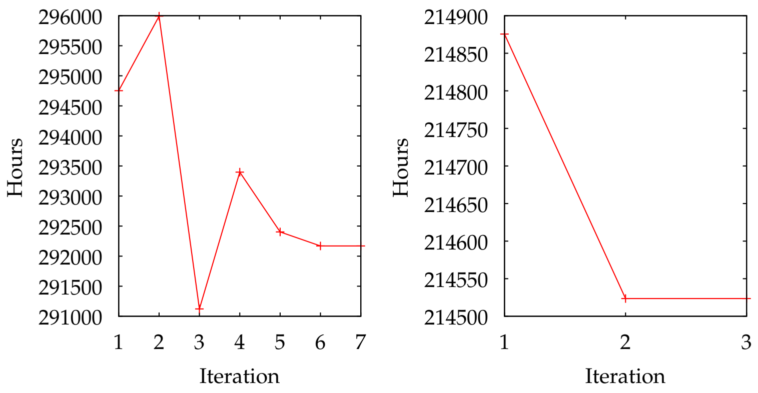
The goal was to verify the theoretical result stated in Theorem 4.1. Therefore, we optimized the timetable exclusively for the transfer waiting time. The result were obtained for the instance `N` and different period times. In particular, we investigated two different scenarios

1. passenger routing based on the equal distribution, and
2. passenger routing based on the just-in-time distribution.

Table 5.6 summarizes the results, where we use the following notation in the table header.

Iter	Iteration
Travel	Total travel time
Waiting	Total transfer waiting time
Transfers	Total number of transfers

Let us investigate the total travel time of the passenger flow in each iteration of the re-timetabling procedure. Figure 5.7 shows the progress for both scenarios and $T = 60$. As already pointed out in Section 4.4: in general, there is no monotonicity of the total travel time if we consider an equal distribution of the passsengers and the timetable does not incorporate the initial waiting time. For the just-in-time distribution the initial waiting time is zero and according to Theorem 4.1 the total travel time is monotone decreasing. An interesting point is that in both cases the procedure terminates after less than 7 iterations. This was also the case for the remaining period times we investigated, as to be seen in Table 5.6. The table also reports on the total transfer waiting time and the total number of transfers in each re-timetabling iteration.



(a) Equal distribution. (b) Just-in-time distribution.

Figure 5.7 Total travel time for the instance N and period time $T = 60$.

Table 5.6 Re-timetabling on instance N.

		Equal Distribution			Just-in-time Distribution		
	Iter	Travel	Waiting	Transfers	Travel	Waiting	Transfers
$T = 15$							
	1	830574000	3490920	33884	767315000	2295960	24064
	2	830613000	2890030	33795	766881000	1858020	24130
	3	830933000	2820150	38629	766881000	1858020	24130
	4	830933000	2820150	38629			
$T = 20$							
	1	856892000	4473620	34990	768755000	3047580	25143
	2	856241000	3803680	36490	768579000	2896260	24988
	3	856186000	3775960	37348	768579000	2896260	24988
	4	854977000	3598000	36773			
	5	855257000	3632530	36773			
	6	855257000	3632530	36773			
$T = 30$							
	1	907332000	6549220	34697	769878000	3796440	25942
	2	908477000	5922900	35548	769878000	3796440	25942
	3	909309000	6016150	37079			
	4	909309000	6016150	37079			
$T = 60$							
	1	1061110000	15641400	38423	773552000	6452340	26221
	2	1065570000	13244100	37528	772285000	5388540	26128
	3	1048040000	13481700	43077	772285000	5388540	26128
	4	1056230000	11067100	39764			
	5	1052650000	11569500	41708			
	6	1051810000	10817900	42160			
	7	1051810000	10817900	42160			

6 Conclusions

In this thesis we studied mathematical models for periodic timetabling and investigated a rich set of existing and new integer programming formulations.

We think there remain some interesting topics concerning the X-PESP formulations that we have developed for the PERIODIC EVENT SCHEDULING PROBLEM (see Chapter 3). Until now we have not investigated primal heuristics for determining feasible start solutions. In particular, the combination of such heuristics with constraint branching would be of interest. Another point concerns preprocessing of the expanded event-activity graph. Lindner (2000) presents several techniques by which we can tighten the time windows of the activities and reduce the number of nodes in the event-activity graph. Minor adjustments included, we are confident that these techniques can be used to reduce the size of the expanded event-activity graph, and thus the size of the X-PESP formulations. Another idea is to apply Lagrangian relaxation as done by Caprara *et al.* (2000) for a similar model.

In our computational study, we tested the event fixation technique, described in Section 3.3.1, only for a special case. We assumed that the running times as well as the dwell times of the lines are fixed. It would be desirable to see if it is similar effective in a more general setting. This, however, requires to find an alternative for the valid equalities that we developed for the X-PESP formulations (see Section 3.2.1). These proved extremely effective to strengthen the LP relaxation, but only work under the special assumptions.

Furthermore, we are interested in an improvement of the constraint branching scheme for solving the X-PESP formulations (see Section 3.3.2). We made the tests with an implementation of preliminary nature. A deeper investigation could confirm the positive results on a more balanced branch-and-bound tree.

In Chapter 4 we examined an iterative approach of timetabling and passenger routing. For the special case in which we do not optimize the timetable according to the number of required vehicles, the quality of the resulting timetables could be characterized. Our computational experiments confirmed these results for the just-in-time distribution and show a relation between the timetable and the travel behavior of the passengers. What is left is to investigate the equal distribution while considering the initial waiting time in the timetable computation.

A question still unanswered is whether the procedure converges if we also minimize the number of required vehicles. If this is the case, it is natural to question the quality of the computed timetable compared to a solution of an integrated approach.

Our computational study indicated that the mixed integer programs of the new X-PESP formulations are large in size even though we have only considered a small subset of timetable requirements. This is surely a limiting factor when solving the formulations by mixed integer programming.

The approach shows, however, that with the current PESP formulations, we have not reached the end of the modeling possibilities. Whether time discretization is an adequate method to formulate the PERIODIC EVENT SCHEDULING PROBLEM remains to be seen.

A

Tables

In Section 5.3.3 we compared PESP and X-PESP formulations on a set of test instances and for different period times. Here, we provide additional computational results for two designated formulations. As a representative PESP formulation, we consider PESP *tree* (2.7). This formulation is compared to X-PESP *pro* (3.5) with line fixation (L), valid equalities (I), and constraint branching (C). In contrast to the previous comparison, we now also give results for more theoretical period times, such as $T = 35$. Table A.1 gives an overview of the tests. It follows an explanation of the table headers.

T	Period time
Conss	Number of constraints
Vars	Number of variables
Gap	Primal-dual gap as defined in Section 5.3
Root Gap	Primal-dual gap in the root node as defined in Section 5.3
Time	Elapsed CPU time in seconds
Nodes	Number of branching nodes

Table A.1 Overview of the tables.

	PESP *tree* (2.7)	X-PESP *pro* (3.5) LIC
N	A.2	A.5
Ns	A.3	A.6
P	A.4	A.7

Table A.2 PESP *tree* (2.7) on instance N.

T	Conss	Vars	Root Gap%	Primal Bound	Gap%	Nodes	Time
5	308	155	42.5	30839.2	0.0	373442	112.0
10	316	159	46.0	73389.1	0.0	1187039	349.0
15	314	159	43.1	107783	0.0	278162	90.0
20	316	159	44.9	153184	0.0	672164	268.0
25	316	159	44.6	194000	0.0	996196	322.0
30	314	159	48.4	229440	0.0	87793	36.0
35	312	157	39.8	264859	0.0	9950	4.0
40	316	159	49.2	304892	0.0	742201	235.0
45	316	159	40.4	332293	0.0	978141	382.0
50	316	159	45.6	378745	0.0	110917	46.0
55	316	159	40.8	402467	0.0	571675	229.0
60	314	159	47.9	440948	0.0	40483	14.0

Table A.3 PESP *tree* (2.7) on instance Ns.

T	Conss	Vars	Root Gap%	Primal Bound	Gap%	Nodes	Time
5	194	98	44.8	35812	0.0	8327	3.0
10	198	100	54.7	76088.5	0.0	5567	1.0
15	198	100	47.1	119306	0.0	33490	9.0
20	198	100	49.2	171746	0.0	51225	12.0
25	198	100	45.7	197636	0.0	19708	4.0
30	198	100	51.3	233511	0.0	10127	3.0
35	198	100	52.3	298948	0.0	62389	14.0
40	198	100	52.7	347074	0.0	48683	14.0
45	198	100	47.0	378694	0.0	93886	20.0
50	198	100	51.8	426852	0.0	19000	5.0
55	198	100	49.5	488264	0.0	70953	17.0
60	198	100	44.5	534990	0.0	160665	30.0

Table A.4 PESP *tree* (2.7) on instance P

T	Conss	Vars	Root Gap%	Primal Bound	Gap%	Nodes	Time
5	712	357	50.1	19491.2	0.0	636861	629.0
10	732	367	49.5	43762.7	0.1	2197169	2135.0
15	730	366	41.5	66739.8	0.1	1922820	1900.0
20	732	367	49.8	93458	0.1	2100666	2189.0
25	738	370	45.6	99153.5	0.1	1906383	2738.0
30	740	371	45.1	125266	0.1	1929843	2743.0
35	736	369	51.1	167395	0.1	1988811	2490.0
40	736	369	52.3	197736	0.1	1868981	1388.0
45	736	369	51.1	202464	0.1	2046663	2217.0
50	740	371	53.6	210069	0.1	2174814	3013.0
55	740	371	54.9	229663	0.1	1981413	2143.0
60	740	371	56.0	258422	0.1	1937384	2669.0

Table A.5 X-PESP *pro* (3.5) LIC on instance N.

T	Conss	Vars	Root Gap%	Primal Bound	Gap%	Nodes	Time
5	1322	2870	20.6	30839.19	0.0	98	6.3
10	2592	11190	14.2	73389.09	0.0	209	37.5
15	3862	24960	16.9	107782.63	0.0	230	154.0
20	5132	44180	12.5	153183.89	0.0	250	457.2
25	6402	68850	11.4	194000.38	0.0	390	1584.8
30	7672	98970	8.2	229440.38	0.0	83	897.8
35	8942	134540	3.2	264859.08	0.0	20	444.9
40	10212	175560	9.7	304892.19	0.0	124	3226.4
45	11482	222030	12.1	339145.89	4.4	80	7200.0
50	12752	273950	10.6	383219.88	3.1	56	7200.0
55	14022	331320	8.4	1e+20	–	44	7200.0
60	15292	394140	2.0	440948.45	0.0	48	3001.5

Table A.6 X-PESP *pro* (3.5) LIC on instance Ns.

T	Conss	Vars	Root Gap%	Primal Bound	Gap%	Nodes	Time
5	926	1964	9.6	35812	0.0	28	3.1
10	1826	7679	3.7	76088.5	0.0	10	5.2
15	2726	17144	7.4	119306.5	0.0	29	15.6
20	3626	30359	7.1	171746.5	0.0	65	52.0
25	4526	47324	2.0	197636.5	0.0	11	20.2
30	5426	68039	0.5	233511	0.0	12	94.9
35	6326	92504	6.0	298948.5	0.0	51	268.5
40	7226	120719	10.5	347074.5	0.0	158	999.6
45	8126	152684	8.5	378693.5	0.0	129	1517.8
50	9026	188399	6.1	426852	0.0	37	793.6
55	9926	227864	10.1	488264	0.0	170	3852.6
60	10826	271079	8.6	534990	0.0	106	5841.0

Table A.7 X-PESP *pro* (3.5) LIC on instance P.

T	Conss	Vars	Root Gap%	Primal Bound	Gap%	Nodes	Time
5	3578	8614	–	19491.14	0.0	58	34.5
10	7098	34079	–	42426.27	0.0	656	926.0
15	10618	76394	–	65382.56	0.0	429	3341.7
20	14138	135559	–	93719.28	8.6	227	7200.0
25	17658	211574	–	1e+20	–	98	7200.0
30	21178	304439	–	1e+20	–	37	7200.0
35	24698	414154	–	1e+20	–	26	7200.0
40	28218	540719	–	1e+20	–	12	7200.0
45	31738	684134	–	1e+20	–	5	7200.0
50	35258	844399	–	1e+20	–	1	2851.8
55	0	0	–	1e+20	–	0	8.1
60	0	0	–	1e+20	–	0	10.0

Bibliography

Beale, E. and Tomlin, J. (1970). Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables. In *Proceedings of the Fifth IFORS Conference*. 447–454.

Bollobás, B. (1998). *Modern Graph Theory*, Springer, vol. 184 of *Graduate texts in mathematics*.

Brucker, P., Burkard, R. E., and Hurink, J. (1990). Cyclic schedules for r irregularly occurring events. *Journal of Computational and Applied Mathematics*, 173–189.

Caprara, A., Fischetti, M., and Toth, P. (2000). Modeling and solving the train timetabling problem.

Daduna, J. R. and Voß, S. (1993). Practical Experiences in Schedule Synchronization. In *Proceedings of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport*. 39–55.

Domschke, W. (1989). Schedule synchronization for public transit Networks.

Fortet, R. (1960). Applications de lálgebre de Boole en recherche operationelle. *Revue Francaise Recherche Opérationelle*, *4*, 17–26.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability*. Freeman.

Gertsbakh, I. and Serafini, P. (1991). Periodic transportation schedules with flexible departure times : An interactive approach based on the periodic event scheduling problem and the deficit function approach. *European Journal of Operational Research*, *50*(3), 298–309.

Klemt, W.-D. and Stemme, W. (1987). Schedule Synchronization for Public Transit Networks. In *Proceedings of the Fourth International Workshop on Computer-Aided Scheduling of Public Transport*. 327–335.

Kroon, L. G. and Peeters, L. W. P. (2003). A variable trip time model for cyclic railway timetabling. *Transportation Science*, *37*(2), 198–212.

Liberti, L. (2007). Compact linearization for binary quadratic problems. *4OR*, *5*(3), 231–245.

Liebchen, C. (2006). *Periodic Timetable Optimization in Public Transport*. Ph.D. thesis, Technische Universität Berlin.

Liebchen, C. and Peeters, L. (2001). Some Practical Aspects of Periodic Timetabling. In *Operations Research Proceedings*.

Lindner, T. (2000). *Train Schedule Optimization in Public Rail Transport*. Ph.D. thesis, Universität Braunschweig.

Müller-Hannemann, M., Schulz, F., Wagner, D., and Zaroliagis, C. (2007). *Algorithmic Methods for Railway Optimization*, Springer, chap. Timetable Information: Models and Algorithms. To appear.

Nachtigall, K. (1996a). Cutting Planes for a Polyhedron associated with a Periodic Network. DLR Interner Bericht 112-96/17.

Nachtigall, K. (1996b). Periodic network optimization with different arc frequencies. *Discrete Applied Mathematics*, *69*(1-2), 1–17.

Nachtigall, K. (1998). *Periodic Network Optimization and Fixed Interval Timetables*. Ph.D. thesis, Universität Hildesheim.

Nachtigall, K. and Opitz, J. (2007). A Modulo Network Simplex Method for Solving Periodic Timetable Optimisation Problems. In *Operations Research Proceedings*. Springer, 461–472.

Nemhauser, G. L. and Wolsey, L. A. (1999). *Integer and Combinatorial Optimization*. Wiley.

Odijk, M. A. (1994). Construction of periodic timetables; part I: A cutting plane algorithm. Tech. Rep. DUT-TWI-94-61, Delft, The Netherlands.

Peeters, L. W. (2003). *Cyclic Railway Timetable Optimization*. Ph.D. thesis, Erasmus Universiteit Rotterdam.

Sahni, S. and Gonzalez, T. (1976). P-complete approximation problems. *J. ACM*, *23*(3), 555–565.

Schrijver, A. (2003). *Combinatorial Optimization*. Springer.

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

Serafini, P. and Ukovich, W. (1989b). A mathematical model for the fixed-time traffic control problem. *European Journal of Operational Research*, *42*(2), 152–165.