Stellingen behorende bij het proefschrift

Railway Timetable Generation

door

Michiel Adriaan Odijk

Technische Universiteit Delft, 12 januari 1998

1. De resultaten in dit proefschrift dragen bij tot het ontwerpen van spoor-infrastructuur op een zodanige manier dat de balans tussen kosten en bruikbaarheid beter kan worden opgemaakt dan tot op heden het geval is.

2. Zij $C \subseteq \mathbb{R}^n$ vol-dimensionaal, convex, compact en puntsymmetrisch met centrum $c$. Door middel van simulatie van een Markov keten die $C$ als toestandsruimte heeft en die convergeert naar de uniforme verdeling op $C$ kan een punt $y \in C$ worden generereerd volgens een verdeling die dicht ligt bij de uniforme verdeling op $C$, zie bijvoorbeeld [1]. Om deze techniek praktisch toepasbaar te maken moet vaak vooraf het aantal te simuleren toestandsovergangen $n$ worden bepaald, waarbij $y = x_n$ wordt genomen. Ofschoon het aantrekkelijk is om zo'n simulatie in $c$ te beginnen, kan het op basis van Stelling 4.15 uit dit proefschrift beter zijn om te starten in een punt $s \in C$, $s \neq c$, en na afloop van de simulatie met kans $\frac{1}{2}$ het punt $y = x_n$ te nemen en met kans $\frac{1}{2}$ het punt $y = 2c - x_n$.

3. Zij $A \in \{0, 1\}^{m \times n}$ een matrix en $e_k = (1, \ldots, 1) \in \mathbb{R}^k$. Beschouw het Set Covering Probleem

$$\min \ c^T x$$

o.d.v. $Ax \geq e_m$

$x \in \{0, 1\}^n$.

Op basis van Steiner Triple Systems (zie [2]) wordt in [3] aangetoond hoe een klasse van probleeminstanties gegeneerd kan worden voor $n = 3, 9, 15, 27, 45, 81, 135, 243$. Voor het probleem uit deze klasse met $n = 135$ is er een cover met cardinaliteit ten hoogste 103 en voor $n = 243$ is deze waarde 198. Deze laatste oplossing is vermoedelijk optimaal, zie Hoofdstuk 6 van dit proefschrift.


4. Beschouw het systeem

$$l \leq B^T \tau + T p \leq u$$

$$\tau \in \mathbb{R}^N$$

$$p \in \mathbb{R}^M,$$

waarbij $T > 0$ een geheeltallige parameter is. Hierin zijn verder $l, u \in \mathbb{Z}^M$ met $0 < u - l < T$ en is $B$ de knoop-kant incidentiematrix van een gerichte graaf met $N$ knopen en $M$ kanten, zie Hoofdstuk 3 van dit proefschrift. Veronderstel dat het oplosbaar is. Zij $\kappa$ de kleinste fractie van intervallen uit $\{[l_m, u_m] \mid m = 1, \ldots, M\}$ die aangepast moeten worden om het systeem oplosbaar te maken. Dan geldt

(a) Als $T$ polynomaal mag afhangen van $N$, dan is er een klasse van probleeminstanties waarvoor $\kappa \geq (N - 2)/(2N - 1) \rightarrow \frac{1}{2} (N \to \infty)$ en

(b) als $T$ exponentieel mag afhangen van $M$ (of equivalent $N^2$), dan is er een klasse van probleeminstanties waarvoor $\kappa \geq (N - 2)/N \rightarrow 1 (N \to \infty)$.

5. Het volgende veralgemineert een resultaat uit Hoofdstuk 4 van dit proefschrift. Beschouw het systeem

$$Ax + By \leq h$$

$$x \in \mathbb{R}^m$$

$$y \in \mathbb{R}^k,$$

waarin $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{n \times k}$ en $h \in \mathbb{R}^n$. Zij $\Lambda \subseteq \mathbb{R}^m$ de verzameling punten $x$ waarvoor

$$K(x) = \{ y \in \mathbb{R}^k \mid By \leq h - Ax \}$$

niet leeg is en zij $V \subseteq \mathbb{R}^n$ de verzameling van extreme richtingen van de polyhedrale kegel $\{ u \in \mathbb{R}^n \mid B^T u = 0, \ u \geq 0 \}$. Dan geldt

$$\Lambda = \{ x \in \mathbb{R}^m \mid v^T Ax \leq v^T h \text{ voor alle } v \in V \}.$$
Voor $x \in \Lambda$ zij $\text{vol}(K(x))$ het $k$-dimensionale geometrische volume (Lebesgue maat) van $K(x)$. Veronderstel dat $\Lambda$ $m$-dimensionaal is en dat $\text{vol}(K(x)) < \infty$ voor alle $x \in \Lambda$. Dan geldt dat $\text{vol}^{1/k}(K(\cdot))$ concaaf is op $\Lambda$ en dat voor $x \in \Lambda$ $\text{vol}(K(x)) = 0$ dan en slechts dan als $x$ op de rand van $\Lambda$ ligt.

Definieer vervolgens voor $x \in \Lambda$ en $\epsilon > 0$

$$f_{\epsilon}(x) = \int_{\Lambda \cap B_\epsilon(x)} \text{vol}(K(y)) \, dy,$$  \hspace{1cm} (5)

waarbij $B_\epsilon(x)$ de bol om $x$ met straal $\epsilon$ is en zij $\text{dim}(K(x))$ de geometrische dimensie van $K(x)$. Zij verder $\Phi \subset \Lambda$ eindig en definieer

$$\eta = \max \{ \text{dim}(K(p)) \mid p \in \Phi \}.$$  \hspace{1cm} (6)

Tot slot definieer voor $p \in \Phi$

$$\mathcal{R}(p) = \lim_{\epsilon \to 0} \frac{f_{\epsilon}(p)}{\sum_{z \in \Phi} f_{\epsilon}(z)}. $$  \hspace{1cm} (7)

Dan geldt dat $\mathcal{R}(p) > 0$ dan en slechts dan als $\text{dim}(K(p)) = \eta$ en als $\eta = k$ dan geldt tevens

$$\mathcal{R}(p) = \frac{\text{vol}(K(p))}{\sum_{z \in \Phi} \text{vol}(K(z))}. $$  \hspace{1cm} (8)

6. Beschouw het Hopfield Netwerk zoals beschreven in [4], d.w.z. een netwerk van $N$ neuronen met geleidingswaarde $T_{ij} > 0$ tussen neuron $i$ en $j$ en externe invoor $I_i$ voor neuron $i$. De effectieve invoor $u_i$ voor neuron $i$ ontwikkelt zich volgens de bewegingsvergelijking

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} + \sum_{j=1}^{N} T_{ij} V_j + I_i.$$  \hspace{1cm} (9)

Hierin is $\tau > 0$ een constante en $V_i = g(u_i) \in (0,1)$ de toestand van neuron $i$ met

$$g(x) = \frac{1}{1 + e^{-x/\beta}},$$  \hspace{1cm} (10)

$\beta > 0$. De totale energie in het netwerk wordt gegeven door

$$E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} T_{ij} V_i V_j + \sum_{i=1}^{N} I_i V_i + f(V_i),$$  \hspace{1cm} (11)

welke een Liapunov functie voor (9) is en waarin $f(V_i)$ de integraal-term wordt genoemd die erg klein is als $\beta$ klein is. Dus voor kleine $\beta$ is $E$ ongeveer gelijk aan

$$E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} T_{ij} V_i V_j + \sum_{i=1}^{N} I_i V_i $$  \hspace{1cm} (12)
en op deze manier kan $E$ geminimaliseerd worden door (9) te simuleren, gegeven de $I_i$ en de $T_{ij}$. Derhalve heeft het model rekenkracht om hoger-dimensionale kwadratische functies over het inwendige van de eenheidskubus te minimaliseren. Echter het weglaten van de integraal-term verstoord de rekenkracht ($E$ daalt niet monotoon). In [5] worden twee oplossingen voor dit probleem vergeleken. In de eerste oplossing wordt in (9) de term $-u_i/\tau$ weggelaten waardoor (12) een Liapunov functie wordt voor de aangepaste bewegingsvergelijking en in de tweede oplossing wordt $g$ vervangen door

$$g_\eta(x) = \frac{1}{1 + e^{-x/q_\eta,\tau}},$$

met

$$q_{\eta,\tau} = \alpha e^{t/\tau},$$

waarin $t$ de tijd expliciet aangeeft. Bij deze laatste oplossing is (12) een Liapunov functie voor (9). Noemen we voor het eerste aangepaste model de bijbehorende oplossing $V_1 = g(u)$ met startwaarde $V_1(0) = \hat{V}$ en voor het tweede model $V_2 = g_\eta(u)$ voor startwaarde $V_2(0) = \hat{V}$, dan geldt

$$V_1(t) = V_2(\tau \ln(1 + \frac{\alpha t}{\beta \tau})).$$

Beide modellen hebben derhalve dezelfde rekenkracht.


7. Het volgende is een uitbreiding van de Perceptron convergentie stelling [6] naar vaste drempelwaarde. Zij $S$ een eindige deelverzameling van $\mathbb{R}^n$ en neem aan dat er een getal $\theta \geq 0$ en een vector $w \in \mathbb{R}^n$ bestaan zodanig dat $x^Tw > \theta \|w\|$ voor alle $x \in S$, waarbij $\|w\| = (w^Tw)^{1/2}$. Zij $\psi > 0$ en zet $w_0 \in \mathbb{R}^n$. Voer iteratief voor $k = 1, 2, \ldots$ de volgende stap uit. Indien mogelijk, kies $x_k \in S$ zodanig dat $x_k^Tw_{k-1} \leq \theta \|w_{k-1}\|$ en defineer $w_k = w_{k-1} + \psi x_k$, anders defineer $v = w_{k-1}$ en stop. Dit algoritme eindigt in een eindig aantal stappen met $v \in \mathbb{R}^n$ waarvoor $x^Tv > \theta \|v\|$ voor alle $x \in S$. De Perceptron convergentie stelling volgt voor $\theta = 0$.


8. Het verlenen van ere-doctoraten dient te worden afgeschaft.

9. Het plannen van vervoer per trein wordt in sterke mate bemoeilijkt door EG besluit 91/440.
Railway Timetable Generation
Railway Timetable Generation

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. J. Blaauwendraad,
in het openbaar te verdedigen ten overstaan van een commissie,
doors het College van Dekanen aangewezen,
op maandag 12 januari 1998 te 16.00 uur

doors

Michiel Adriaan ODJIK

wiskundig ingenieur

geboren te Papendrecht
Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. R.M. Cooke
Toegevoegd promotor: Dr. H. van Maaren

Samenstelling promotiecommissie:
Rector Magnificus
Prof. dr. R.M. Cooke
Dr. H. van Maaren
Prof. dr. F.M. Dekking
Prof. Dr.-Ing. habil G. Hertel
Prof. dr. A. Schrijver
Dr. H.E. Romeijn
Dr. E.A.G. Weits

voorzitter
Technische Universiteit Delft, promotor
Technische Universiteit Delft, toegevoegd promotor
Technische Universität Dresden,
Daimler-Benz AG, Frankfurt am Main
Universiteit van Amsterdam,
Centrum voor Wiskunde en Informatica, Amsterdam
Erasmus Universiteit Rotterdam
Railned b.v., Utrecht

Published and distributed by:
Delft University Press
Mekelweg 4
2628 CD Delft
The Netherlands
Telephone: +31 15 2783254
Telefax: +31 15 2781661
E-mail: dup@dup.tudelft.nl

CIP-GEGEvens KONINKLIJKE BIBLIOTHEEK DEN HAAG

Odijk, Michiel A.
Railway Timetable Generation / Michiel A. Odijk – III.
Ph.D. thesis Delft University of Technology. – With ref. – With summary in Dutch
ISBN 90-407-1611-0
NUGI 811
Subject headings: Railway infrastructure / Railway timetables / Randomized algorithms

Copyright © 1997 by Michiel A. Odijk
All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the publisher: Delft University Press.

Printed in The Netherlands
Contents

I Introduction, Background, and Preliminaries 1

1 Introduction 3
    1.1 Specifying railway infrastructure at stations 5
    1.2 Generating timetables: conceptually 9
    1.3 Generating timetables: technically 12
    1.4 Other applications and related models 17

2 Mathematical Background and Preliminaries 25
    2.1 Randomized algorithms 26
    2.2 Sampling from convex bodies 29
    2.3 Markov chain sampling 30
    2.4 Notations and conventions 34

II Railway Timetable Generation 35

3 Periodic Timetables: Model, Analysis, Algorithms, and Complexity 37
   3.1 The timetable structure model 38
   3.2 Computational complexity 41
   3.3 Generating a timetable 45
   3.4 An example 51

4 Random Generation of Classes of Railway Timetables 59
   4.1 The timetable structure model 60
       4.1.1 Partitioning of the set of timetables 61
   4.2 Generating phase shifts at random 64
       4.2.1 Properties of the phase shift polytope 64
       4.2.2 A distance measure on the phase shift polytope 68
       4.2.3 The volume function 69
       4.2.4 The volume distribution 70
       4.2.5 The robustness distribution 72
   4.3 The heuristic 85
   4.4 Examples 89
5 Random Generation and Counting of Railway Timetables
5.1 The model ........................................ 110
5.2 Uniform generation of timetables ......................... 112
5.3 A small example ........................................ 121
5.4 Computational complexity .................................. 126
5.5 Counting timetables .................................... 131

III Miscellaneous ........................................ 137

6 Improved Solutions to the Steiner Triple Set Covering Problem ........................................ 139
6.1 Known and new results .................................. 141
6.2 The algorithm ........................................... 142
6.3 Discussion ............................................. 145

Bibliography ............................................. 149

Railway Timetable Generation (summary)

Generatie van Treindienstregelingen (samenvatting)

Curriculum Vitae
Part I

Introduction, Background, and Preliminaries
Chapter 1

Introduction

Dutch Railways (Utrecht, the Netherlands, abbr. NS) exploit a nation-wide railway passenger and cargo transportation system consisting of a network of about 2,800 kilometres of railway tracks that connect about 400 train stations, thereby throughputing up to 5,000 trains per day [144]. NS offers its clients one of the most extensive sets of regular fixed-interval timetables in the world at a high service level. To maintain and even improve this performance, NS has initiated a major railway investment program in the late eighties, called Rail 21, thereby anticipating an expected increase in the number of people traveling by rail in the next century. Also, in the early nineties it was decided that NS will loose its monopoly on the railway network as agreed within the European Community. In this new situation several railway exploiters may claim the use of (parts of) the railway infrastructure, the building and upkeep of which remains the responsibility of the government. Hence, a flexible, long lasting, railway system must result and for the first time in post-war Dutch railway history, commercial marketing demands will determine the production means and push railway technology.

Among other things Rail 21 involves the doubling of tracks in the western part of the Netherlands (where traffic is congested the most) as well as the reconstruction of several railway stations [10]. To develop support systems for the planning of railway infrastructure a number of projects have been carried out by Railned (Utrecht, the Netherlands), an independent intermediate between NS and other railway exploiters on the one side and the government on the other side. Railned primarily advises the government on the allocation of track capacity and railway safety and, more or less as a consequence, also on investments in the infrastructure. Some of the projects are realized in conjunction with others, among which are academic/scientific institutes like Delft University of Technology (Delft, the Netherlands), Eindhoven University of Technology (Eindhoven, the Netherlands), Erasmus University (Rotterdam, the Netherlands), University of Amsterdam (Amsterdam, the Netherlands), and the Center for Mathematics and Informatics (Amsterdam, the Netherlands).

In this thesis we propose a method to support the specification of railway tracks at stations and work out the mathematical details for a specific part of the method. It is important to understand that with (railway) infrastructure we mean tracks (inside and
outside stations) plus station platforms and exclude power supply systems, station buildings, etc. Requirements for the latter set of real estate are assumed to be implied by the railway track specifications. Rolling stock and personnel is also excluded from our notion of railway infrastructure. Instead we assume that enough rolling stock and personnel can always be made available on relatively short notice. Also, we only deal with railway stations for transportation of passengers and not with railway yards dedicated to freight transportation.

Building or reconstructing railway infrastructure may take ten to fifteen years, counting from the moment a project starts. Moreover, usually large amounts of money are involved and infrastructure, once built, must be useful for a long time. Therefore, it is crucial to define building projects such that they are robust with respect to future timetables. However, these timetables are founded on many uncertain starting points. An important reason for this is an EC directive to allow for competition on the European railway networks. It is expected that the Dutch railway system will respond sensitively as it is easily accessible from Germany and France and, moreover, the political climate in The Netherlands is favourable to the directive. What’s more, a number of important and far reaching political decisions concerning strategic investments in the railway infrastructure are deferred to later years. As a contrast, in Switzerland the uncertainties regarding future timetables are much smaller because timetables are fixed for a long time ahead [41].

When Rail 21 kicked off in 1985 one of the first problems that infrastructure planners encountered was the handling of the aforementioned uncertainties. More precisely, the big question is how to specify railway infrastructure such that it is likely to be robust with respect to future timetables. Experience was hardly available because up to 1985 modifications of the infrastructure were mainly restricted to maintenance and small isolated projects. Large investments as in the framework of Rail 21 have never been in order in modern history. Moreover, over the past few decades timetables slowly evolved and never fluctuated much, so that specifications of small adaptations were always based on reliable and complete information.

In an attempt to deal with the uncertainties about future timetables, rough capacity checks were developed to determine, for instance, the number of platforms required at a railway station given some estimated characteristics of future timetables [146]. However, many characteristics are not normed and difficult to interpret [96], so that the use of them has always been controversial. Moreover, rough capacity checks do not yield a clear and constructive way to specify infrastructure.

It was felt that things would improve if specifications could be based on more down-to-earth information. In particular the use of timetables seems inevitable. However, many timetables would be needed to make the specifications robust with respect to future timetables and until recently constructing a single new timetable was extremely laborious because it had to be done manually (it took a few experienced timetable planners a couple of months). Hence, the need to automate the construction of sets of different future timetables, something that we address in this thesis.

In this thesis we propose a novel approach to specify railway infrastructure at railway stations. It is meant to deal with the uncertainties about future timetables and presupposes
knowledge about the railway infrastructure that connects railway stations. That is, in the
approach it is assumed that the infrastructure outside the station is fixed and known.
Decisions about infrastructure that connects stations are made on a higher planning level
and is not an issue in this thesis.

The approach is based on the generation of many timetables given some presuppositions
concerning the construction of future timetables. Specifying infrastructure using these
timetables, then, should make the final specification timetable-independent to a certain
extent and, hence, robust with respect to these starting points. Although the concept
is rather intuitive, it is rather complex too. Especially the generation of timetables is
far from trivial and mathematically challenging. The larger part of this thesis is devoted
to timetable generation, whereas the other elements of the specification concept are only
discussed in head lines in this chapter.

The remainder of this thesis is organized as follows. In this chapter we propose the
approach to specify infrastructure at railway stations. That is, in the following section
the approach is outlined and in the two sections thereafter the generation of timetables is
discussed on two levels, one being the conceptual level, the other being the technical level.
Moreover, in Section 1.1 the scope of this thesis is further narrowed down. The technical
details of timetable generation are deferred to the chapters of Part II of this thesis. In
Section 1.3 the coherence between these chapters will be made clear. Section 1.4 deals
with some related models.

Part III contains a chapter that has little to do with timetable generation. It states
new results for a hard set covering problem arising from Steiner triple systems.

In Chapter 2 some background material is provided for the chapters in Part II and III.
There, the level of treatment is kept rather general and serves no other purpose than to
clarify and motivate the approaches explained in subsequent chapters.

1.1 Specifying railway infrastructure at stations

Hereafter, we refer to the concept of specifying railway infrastructure at railway stations,
that we propose in this section, as the infrastructure specification method. It consists of
several stages, which are sequentially dealt with, give or take some backtracking. Figure 1.1
depicts these steps (backtracking is not indicated in the figure).

The idea behind the infrastructure specification method is to generate many reference
timetables, search for features they have in common, and use these features to draw up a
specification of the track layout of the station under study. This way such a specification
is based on clear and easily to interpret data. Moreover, since each reference timetable is
in fact just another interpretation of the starting points on which it is based, robustness
of the specification with respect to these starting points can be created. Of course the
specification does depend on the set of reference timetables, but if this set covers at least
some important future railway scenario's (i.e. timetables), then the specification can be
Figure 1.1: The infrastructure specification method.
hoped to be timetable-independent (and, hence, robust) to a reasonable extent. What’s more, full timetable-independency will certainly lead to expensive over-specifications. At all times it remains the responsibility of the infrastructure planner to determine at what point timetable-independency and cost break even.

The infrastructure specification method proceeds as follows. On basis of a set of reference timetables we determine the number of station platforms required by any of those timetables and the maximum number of simultaneous train movements on either side of the station at any time. Together with some other requirements, this information gives direction to the functional specification of the station track layout.

Given the specification, the planner makes a design of the track layout, which is then subjected to several consistency checks and performance evaluations. For example, the design is required to handle a relatively large portion of the reference timetables, i.e. for most of the reference timetables it must be possible to route the trains over the station without modifying the timetables too much. This is what is called ‘verification of design w.r.t. reference timetables’ in Figure 1.1.

To see if the design is robust with respect to planned and fixed changes of the reference timetables, we can repeat the routing step for timetables that deviate slightly from the reference timetables. That is to say, given a reference timetable, we shift arrival and/or departure times a little bit such that the resulting timetables still comply with the starting points that were used to the generate reference timetables. For these perturbed timetables, then, we try to find feasible routings through the station, like we did for the original reference timetables. If we succeed, we gain confidence that the infrastructure design is robust with respect to planned deviations, see Figure 1.1. This part of the anaylsis of the infrastructure design can be viewed as a sensitivity analysis.

We mention two other ways to analyze infrastructure, although we do not recommend to incorporate them in the infrastructure specification method. One way is to simulate a given timetable. In [96, 98, 141] Colored Petri Net models [95, 140] were proposed for the purpose of simulation. It turns out that Petri Net models are quite suitable. For example, the sectional-release route-locking principle [21, 22], which is used at Dutch railway stations to control the traffic, says that a route is claimed ahead as a whole and released bit by bit. This can be modeled in a natural manner using Petri Nets. A major drawback of simulation is that not only the railway network and the timetable must be modeled, but also operational aspects such as traffic control, safety systems, etc. Put differently, simulation requires that too many operational aspects need to be incorporated in the strategic planning level, which is the level at which infrastructure is specified. Simulation can be a useful means to study operational issues such as the robustness of a single timetable with respect to disruptions and to our believe it should stay at the operational level. Along these lines we also mention Max Plus algebras, notably [23] in which a railway application is worked out.

The second alternative way to analyze station track layout designs involves the use of Queueing Theory [40, 96, 123, 143] to calculate blocking probablililities of trains and
occupation rates of track sections. The main drawback of this approach is that it deals with regularity and periodicity very poorly. See [96] for an overview of specific problems with applying Queueing Theory to railway networks. Still, at present efforts are being made by Railned and the University of Amsterdam to develop a new queueing model that are specifically suited for railway networks.

One of the biggest advantages of the infrastructure specification method is that planners of infrastructure can better motivate the choices they must make when specifying: not only the starting points for the reference timetables are clear, but it is also clear where they come from and the way they are converted into specifications becomes logical and transparent. This, of course, helps when the planner must backtrack in the process. Another advantage is that the resulting infrastructure design will be robust or at least it can be better understood why and to what extent it is not robust. For example, a planner may choose to ignore some of the reference timetables because they would lead to the specification of an additional platform, thereby exceeding budget or space limits. With the infrastructure specification method, then, the planner can be fully aware of the risks or opportunities that are at stake.

The main focus of this thesis is on the generation of timetables for the purpose of specifying railway infrastructure at railway stations. Since the timetables form the basis of the infrastructure specification method, this part of the method is of eminent importance, see also [99, 101].

Routing trains over station track layouts in the design evaluation phase of the infrastructure specification method falls outside the scope of this thesis. For information about this consult e.g. [21, 22, 79, 149, 150, 151]. Also, how to draw up specifications for railway infrastructure given a set of timetables is not an issue in this thesis. Preliminary, unpublished, studies indicate that parts of this activity can be automated, although we do not believe that push-button solutions exist for the problem of drawing up a specification.

To understand the history of the proposed infrastructure specification method we refer to [113] for the recent history and to [96, 98] for the very first attempts to develop a specification method based on timetable generation. In these first attempts Colored Petri Nets were used to model railway stations, see also [141]. However, the fundamental difference with the method explained in the above is that the Petri Net models were only used to analyze infrastructure designs and that, hence, the Petri Net approach is not a constructive method to specify infrastructure.

Two final remarks are in order. First, the question of why and how a particular station is selected for reconstruction is addressed in [102, 113]. Such decisions are taken at a higher planning level and are not discussed in this thesis. Here we adopt the point of view to take for granted that a railway station needs to be (re)designed. Second, almost always only a specific part of an existing railway station needs to be redesigned or the station is just slightly extended (which still may have a large impact on other parts of the station). The infrastructure specification method accounts for this by generating the timetables such that they are feasible for the parts of the station that will not be changed.
1.2 Generating timetables: conceptually

An important part of the infrastructure specification method consists of generating future timetables. In this section we explain how this can be done in concept. To this end, we start with briefly outlining the process of constructing timetables. Figure 1.2 provides a visualization. See also [102] for more details.

The process starts with an Origin/Destination-matrix (abbr. OD-matrix), whose entries state the expected future number of travellers between each pair of railway stations. These numbers stem from political targets and estimations of the numbers of travellers, mainly on the basis of historical, economical, geografical, and demographical considerations. Actually, the OD-matrix itself is, in turn, derived from another matrix stating the mobility of people between different geographical zones. The OD-matrix is used to determine the number of travellers between any pair of adjacent railway stations and is input for the next step, being the linking of railway sections into railway lines.

A railway line is a geographically consecutive sequence of stations that can be served by a family of trains with regular time intervals. The way lines are designed may be governed by the wish to offer as many travellers as possible a direct route, the wish to minimize total
waiting time, by other criteria, or a suitable mixture [31]. The set of lines that emerge in
this way is called the line system. Lines that cross each other at a railway station can be
connected so as to enable passengers to transfer from one line to another. Furthermore, the
line system induces a number of safety requirements to prevent collisions. Together these
requirements make up what we call a timetable structure, which is a set of mathematical
constraints modeling all sorts of requirements a timetable must meet, such as headway
times, connections between trains, periodicity (i.e. timetables must repeat themselves at a
constant rate), and regularity (e.g. train families that serve a station twice an hour should
do so at an interval of about half an hour).

The last step of the timetabling process comprises the construction of a timetable from
a timetable structure. Here a timetable is a basic hourly pattern that needs further re-
vision by adding special trains and shunting movements, assigning personnel, and so on,
thereby creating a detailed and executable production plan. For the purpose of specifying
and designing station track layouts this final refinement is not done in order not to com-
promise the planning horizon. In fact, using timetables in the sense of hourly patterns may
already introduce a bit too many details, although this problem can partly be overcome
by considering many such timetables so that effectively the timetable structure is the most
detailed object we consider.

The choices that can be made in each step of the timetabling process lead in a natural
way to a scenario tree, whose root is the OD-matrix and whose leaves are the timetables,
see Figure 1.3. Thus, timetables are considered as railway scenario's in the infrastructure
specification method. By simulating the timetabling process a number of times from start
to finish a set of reference timetables emerges.

At the top of the scenario tree, only a few OD-matrices are considered, each representing
for example a particular government or company policy [10, 102]. On the bottom of the
tree there are many timetables, because, given a timetable structure, many choices for the
resulting timetable can be made. Since these choices will not be made until about ten to
fifteen years from now, analysis of a given timetable structure must necessarily take this
uncertainty into account. One way to do this (and in fact the only way we know of) is to
sample timetables from a given timetable structure at random and analyze them. At the
root and the line systems level of the scenario tree it seems wise to consider a couple of
alternatives, but the need to select them at random is less apparent because the information
is rather aggregated and will, therefore, be less sensitive to future developments.

In this thesis we deal with randomized generation of timetables from a given timetable
structure since this is the level of decision making that is pre-eminently suitable to be
treated formally. In fact, without such treatment this step would not even be possible in
practice. Branching in the scenario tree at the other levels will be based on deterministic
and human decision making because there are too many complicating factors that need to
be approached by common sense rather than by automated tools that are based on formal
reasoning.

The Dutch railway network consists of a large number of stations that all lie close to-
tgether. Local changes in the timetable inevitably bear consequences for railway operations
at neighboring stations and connecting track sections. Hence, ignoring the network func-
Figure 1.3: The scenario tree.
tion of a station may lead to infrastructure specifications that will be difficult to justify, and it is necessary to generate timetables that reflect the fact that a station is just a node in a network.

The ideal way to cope with the network structure is to generate timetables for the entire railway network such that for the target station, i.e. the station under study, unlimited infrastructure is assumed. This way, we obtain realistic timetables that can be used to specify the infrastructure for the target station by requiring that those timetables can be handled by the station layout. Since in a timetable structure so-called infrastructure constraints model the limitations of the infrastructure, assuming unlimited infrastructure can be done by omitting those infrastructure constraints that are relevant to the vicinity of the target station (recall that we generate timetables from timetable structures, see Figure 1.3).

Once timetables are generated, they are projected onto the target station, so that the specification of the required infrastructure can be made. If timetable structures for the entire railway network are too large to quickly generate timetables, we may consider just a large area surrounding the target station, sample timetables for that area, and project those onto the station.

Another thing is the extent to which the timetable structures should express details, see Figure 1.4. For example, apart from the target station, we can include all infrastructure constraints induced by all station track layouts and connecting tracks. However, this will lead to very large systems of constraints. On the other hand, we may confine ourselves to taking only infrastructure constraints for connecting tracks into account and ignore the infrastructure of all stations. This approach seems better if also some other stations may be subject to changes. Anyway, the latter approach suits the strategic decision making level better and taking into account the limitations of the connecting tracks seems to be the least one must do, but doing more is questionable. Note, however, that often only a specific part of the target station is actually considered for reconstruction and, since the rest of the station's infrastructure is closely connected to it, it is better to include all infrastructure constraints for the station that have nothing to do with the part that is under study.

### 1.3 Generating timetables: technically

From the foregoing it is clear that generating timetables at random from a timetable structure is an important first step in the infrastructure specification method. In this section we place the timetable generation problem in a mathematical context. That is, we outline and discuss the timetable structure model that we employ. Also, we present the highlights of our approach to generate timetables at random. Details of the timetable structure model and the generation method can be found in Part II of this thesis.

Constructing a timetable amounts to solving a system of constraints, which is the so-called timetable structure that we encountered before and which defines the structure of the desired timetable. In a timetable structure all constraints are of the form

\[ l_{ij} \leq \tau_i - \tau_j + T_{pij} \leq u_{ij}, \]

(1.1)
Figure 1.4: The highest level of this figure constitutes a railway network connecting stations a, b, c, d, and e. This network consists of connecting tracks, drawn as fat lines, and of stations, drawn as boxes. The box of station c is exploded.
where \( \tau_i \) and \( \tau_j \) are the unknown event times of event \( i \) and \( j \) respectively, \( i, j \in \{0, \ldots, N\} \). For instance, event \( i \) can be the departure (arrival) of a certain train from (at) a certain station. Integers \( l_{ij} \) and \( u_{ij} \) are given numbers such that \( 0 < u_{ij} - l_{ij} < T \) and integer \( T > 0 \) expresses the period of the timetable. The number \( p_{ij} \) is also a priori unknown and must be an integer. In words (1.1) expresses that, given \( \tau_j \), event \( i \) takes place in one of the time intervals \( \ldots, [\tau_j + l_{ij} - T, \tau_j + u_{ij} - T], [\tau_j + l_{ij}, \tau_j + u_{ij}], [\tau_j + l_{ij} + T, \tau_j + u_{ij} + T], \ldots \). This way a solution \( \tau = (\tau_0, \ldots, \tau_N) \) is a periodic timetable. Often \( T \) is set to 60 minutes to model that timetables must be hourly patterns. We may write

\[
l \leq B^T \tau + T p \leq u
\]  

(1.2)

to denote the model in compact vector notation. Vector \( p \) is called the phase shift.

The constraints in a timetable structure may stem from various requirements a timetable is supposed to meet. For one thing, some constraints express driving times between stations. Others denote that two trains must offer passengers a transfer at a certain station. A third type of constraints is induced by headway times that are necessary to account for infrastructural limitations. These are the infrastructure constraints that we already encountered in Section 1.2. Finally, frequency constraints ensure that a certain series of trains that visit a certain station does so at equidistant moments in time. See [121, 122, 139] and Chapter 3 for more details about the timetable structure model.

The problem of solving a timetable structure, i.e. finding a feasible timetable or establishing that this is not possible, is NP-complete in the strong sense [56], as can be verified by reductions from well-known graph problems like e.g. Hamiltonian Circuit [126] or Graph Coloring, see Chapter 3. For details of algorithms to solve timetable structures with, see [100, 121, 122, 126, 139, 142] ¹ and Chapter 3. The method in [142] has been successfully implemented by Schrijver and Steenbeek in 1995. It proved to be suitable to deal with timetable structures that cover large parts of the Netherlands.

The constraints in a timetable structure can be represented by a directed graph \( G \) of which the vertices corresponds to the event times and the arcs to the constraints. Thus, matrix \( B \) in (1.2) is the vertex-arc incidence matrix of \( G \). Now, if we consider an elementary circuit in \( G \), i.e. a circuit that can not be decomposed in subcircuits, and if we add the constraints that correspond to the arcs of that circuit, then a constraint is derived in the \( p_{ij} \) variables only because the event times \( \tau_i \) cancel each other out. We call such a derived constraint a circuit constraint and we can derive such a circuit constraint for all elementary circuits of \( G \). Thus, we can derive a system of linear constraints in the phase shift \( p \) only and it will be shown in Chapter 3, see also [100], that for any phase shift \( \hat{p} \) the system

\[
l - T \hat{p} \leq B^T \tau \leq u - T \hat{p}
\]  

(1.3)

is feasible if and only if \( \hat{p} \) is a solution to the system of circuit constraints. The cutting plane algorithm to find a timetable described in [100] and Chapter 3 exploits this feasibility.

¹In [139] the remark made in Section 3 that in each iteration it is sufficient to keep track of only \( M - N + 1 \) circuit constraints, is not correct. See Chapter 3 for the correct statements.
characterization by using the circuit constraints as cuts to search for a feasible phase shift. Moreover, if no feasible timetable exists, then this algorithm returns an infeasible subsystem, thereby providing a certificate of infeasibility of (1.2).

The feasibility characterization in terms of a system in $p$ only is of interest because it decomposes the process of generating timetables from a timetable structure in two stages. First, we solve a phase shift $\hat{p}$ from the system of circuit constraints, and then we take a timetable $\tau$ from the set $K(\hat{p})$, which is the set of feasible timetables consistent with $\hat{p}$, i.e.

$$K(\hat{p}) = \{ \tau \mid \tau_l - T\hat{p} \leq B^T \tau \leq u - T\hat{p} \}.$$  

(1.4)

Whereas the first task is computationally hard, the second task is easy [100].

Without loss of generality we may stipulate that $\tau_l = 0$ for all $\tau \in K(\hat{p})$. Then, $K(\hat{p})$ is a polytope. Also, without loss of generality we assume that the variables $p_i$ corresponding to a spanning tree in $G$ can be set to 0. Furthermore, the circuit constraints describe a polytope $\Lambda$, so that the feasible phase shifts can be characterized as the integral points in a polytope.

The feasibility characterization has an even more important implication. That is, the set of feasible phase shifts partitions the set of feasible timetables and this allows us to specify more precisely what we mean with a reference timetable and a planned deviation from it, two notions that were introduced in Section 1.1. Recall that reference timetables are used to specify and design infrastructure and that planned deviations from the reference timetables are used to analyze the robustness of the design. If $\hat{r}$ is a reference timetable, then with a planned deviation from $\hat{r}$ we mean any timetable in $K(\hat{p})$, where $\hat{p}$ is the unique phase shift corresponding to $\hat{r}$. Hence, it is natural to generate reference timetables and planned deviations in two steps. In the first step we generate feasible phase shifts from $\Lambda$ and select a timetable for every generated phase shift. These timetables, then, are the reference timetables. In the second step we generate for each reference timetable a couple of planned deviations.

In Chapter 4 we introduce the robustness distribution on $\Lambda$ and propose to use it to generate phase shifts at random. This distribution has the property that, if $\eta$ is the highest geometric dimension of the sets $K(p)$ among all phase shifts in $\Lambda$, then the robustness distribution assigns positive probability mass to a phase shift $p$ if and only if the dimension of $K(p)$ equals $\eta$. If, in addition, $\eta = N$, where $N$ is the number of event times in a timetable ($\tau_0$ excluded), then the probability mass is proportional to the $N$-dimensional geometric volume of the sets $K(p)$. These properties are nice because, if the dimension of $K(p)$ is low, a timetable in $K(p)$ is subjected to many binding dependencies between the arrival and departure times. Hence, such a timetable cannot be expected to be very robust. Since a timetable planner will always be in search for robust timetables, it is therefore natural to use the robustness distribution to generate feasible phase shifts from.

Given a phase shift $\hat{p}$, it is easy to find a timetable in $K(\hat{p})$, see Chapter 3 or [111]. In Chapter 5 we show how to generate planned deviations from a given reference timetable, or, put more precisely, how to generate timetables from $K(\hat{p})$ uniformly at random, given a reference timetable with corresponding phase shift $\hat{p}$. Next, we highlight the approaches used in Part II of this thesis.
As mentioned before, the set of feasible phase shifts are the integral vectors in the polytope $\Lambda$. The general problem of searching for an integral vector in a given polytope is very hard, let alone generating one in compliance with a given probability distribution. Our problem of randomly generating a phase shift from $\Lambda$ is no exception to this. In principle, to sample from the robustness distribution we have to enumerate all phase shifts, calculate their probability mass, and sample. Unfortunately, enumerating all phase shifts is often not tractable. Therefore, we turn to a heuristic approach to approximate the robustness distribution.

The heuristic for approximate sampling from the robustness distribution is described in Chapter 4 and can be summarized as follows. First, a vector $x$, not necessarily integral, in $\Lambda$ is generated according to the volume distribution $V$. This is a continuous probability density on $\Lambda$ that assigns probability mass to a vector $x$ in $\Lambda$ proportional to the geometric volume of $K(x)$, where the definition of $K(x)$ is the continuous analogue of (1.4). Then, $x$ is rounded to an integral vector in $\Lambda$, yielding a feasible phase shift $\hat{p}$.

On the basis of computational experience it is reasonable to expect that the approximation of the robustness distribution by the heuristic is good enough for practical purposes. That is, although the approximation can be very bad in theory, it seems that with the heuristic approach indeed samples are obtained in compliance with the desire to favor ‘robust timetable sets’ over others.

Whereas sampling from the volume distribution $V$ is easy, rounding a sample to a phase shift is a very hard problem because it requires the optimization of a quadratic objective over $\Lambda$. Recall that searching for a phase shift is already an NP-complete problem. Consequently, the applicability of the heuristic is limited by the possibility to deal with the rounding problem. To set the mind, for a timetable structure covering the province of North-Holland and involving about twelve trains we succeeded in obtaining a number of samples within an hour. However, for many bigger timetable structures the rounding problem is too big. In Chapters 3 and 4 an example is worked out for station Arnhem CS.

Another probability distribution of $\Lambda$ that would be appealing to use to generate phase shifts is the uniform distribution because it reflects the situation in which a timetable planner is completely indifferent between any of the phase shifts. However, in Chapter 4 it is shown that from computational point of view the uniform distribution seems even more difficult to handle. Therefore, given that we want to have some control over the generation of the phase shifts and given that we do not want this control to depend on algorithms that solve the timetabling problem, using the robustness distribution is a good basis for generating timetables.

Given $\hat{p}$, to generate timetables from $K(\hat{p})$ we generate integral vectors uniformly at random from this set. In Chapter 5 we describe a Markov chain with state space $K(\hat{p})$ that converges to the uniform distribution. Considering only integral timetables is not really a restriction because the polytope $K(\hat{p})$ is integral, i.e. its vertices are integral vectors.

An interesting by-product of uniform sampling from a set of timetables $K(p)$ is that we can estimate the number of (integral) timetables in it. However, randomized counting involves the sequential simulation of a number of Markov chains and may take too long to be practical, see Chapter 5.
Figure 1.5 shows an example of a part of a timetable that was generated using a software implementation of the theory developed in this thesis. The timetable involves a number of trains in the province of North-Holland in the Netherlands. Figure 1.5 shows the trains that run on the trajectory Amsterdam – Hoorn (Asd–Hnk) in a so-called time-position diagram. A sequence of line segments that forms a monotone line, apart from some interruptions at stations, denotes the running of a single train. If such a line ascends, then the train runs towards Hnk and it runs towards Asd if it is descending. At the top of Figure 1.5 the tracks are shown on the line from Asd to Hnk. For example, from Zd to Hnk there are two tracks. Note that both trains 033 and 045 serve the stations at intervals of about half an hour.

1.4 Other applications and related models

The timetable structure model we use was already introduced in [126] as a model for the Periodic Event Scheduling Problem (PESP). Since then the model has been applied in aviation [60], traffic light scheduling [127], and job shop scheduling [126].

A model very similar to the timetable structure model was introduced by Modrow et al. in [90] and further developed in [34]. The constraints in this model are of the form

\[ l_{ij} \leq \tau_i - \tau_j + T p_{ij} < T, \tag{1.5} \]

with \(0 < l_{ij} < T\) and \(p_{ij}\) integral. Since the inequality in (1.5) does not allow equation and, since the upper bound is fixed to \(T\), the model seems less general than our model at first sight. However, we will show below that for \(T \geq 3\) and at the expense of additional dummy event times each timetable structure can be converted into a model with constraints like (1.5). To this end, consider the constraint

\[ l_{ij} \leq \tau_i - \tau_j + T p_{ij} \leq u_{ij} \tag{1.6} \]

in some arbitrary timetable structure and assume, without loss of generality, that \(0 < u_{ij} - l_{ij} < T\). Note that this constraint is equivalent to the constraint

\[ l_{ij} + kT \leq \tau_i - \tau_j + T p_{ij} \leq u_{ij} + kT \tag{1.7} \]

for any fixed \(k \in \mathbb{Z}\). Now, if we can model the constraints

\[ 1 \leq \tau_i - \tau_j + T p_{ij} \leq u_{ij} \tag{1.8} \]

for all \(1 \leq u_{ij} \leq T\) in the model of Modrow et al., then by concatenating a number of these constraints we can model (1.6).

To show that (1.8) can be modeled in Modrow's model, consider the constraint

\[ T - u_{ij} \leq \tau_j - \tau_i + T p_{ji} < T \tag{1.9} \]
Figure 1.5: Timetable for trains on the trajectory Amsterdam – Hoorn (Asd–Hnk) in North-Holland depicted as a time-position diagram. Time is on the vertical axis and the position of the trains is on the horizontal axis. At the top the (number of) tracks are shown.
for integer \( u_{ij} \) with \( 1 \leq u_{ij} \leq T - 1 \). Clearly, this constraint fits into Modrow's model and it is equivalent to

\[
0 < \tau_i - \tau_j + T \hat{p}_{ij} \leq u_{ij}.
\]

(1.10)

Then, note that (1.10) and

\[
1 \leq \tau_i - \tau_j + T p_{ij} < T
\]

(1.11) imply (1.8) for all \( 1 \leq u_{ij} \leq T - 1 \). Then, by concatenating (1.8) once for \( u_{ij} = 2 \), once for \( u_{ij} = T - 1 \), and \( T - 1 \) times for \( u_{ij} = 1 \) we find

\[
T + 1 \leq \tau_i - \tau_j + T p_{ij} \leq 2T,
\]

(1.12)

which is equivalent to

\[
1 \leq \tau_i - \tau_j + T p_{ij} \leq T.
\]

(1.13)

Hence, in Modrow's model (1.8) can be modeled for all integers \( u_{ij} \) with \( 1 \leq u_{ij} \leq T \).

By discarding dummy event times that were added to the model owing to the concatenations, any solution in the so-constructed Modrow model yields a solution to (1.6). Note that (1.5) can not be modeled by any timetable structure of bounded size.

It is shown in [34] that feasibility can be characterized in terms of the \( p_{ij} \) by deriving a circuit constraint for each elementary circuit, that the problem is NP-complete by reducing the Graph Coloring Problem [56] to it, and that finding \( p_{ij} \) for which a timetable exists constitute the hard part of solving the model. In Chapter 3 we arrive at similar conclusions for the PESP.

In [63] a model for scheduling traffic lights is described that contains the timetable structure model as a special case. Solving that model amounts to solving the program

\[
\min \sum_{(i,j) \in \mathcal{M}} f_{ij}(\delta_{ij}) \\
\text{s.t. } \delta_{ij} = \tau_i - \tau_j,
\]

(1.14)

where the \( f_{ij} \) are periodic functions corresponding to the arcs \((i,j)\) of a directed graph with arc set \( \mathcal{M} \). By taking appropriate penalty functions for the \( f_{ij} \), constraints like (1.6) can be modeled. In the same paper a heuristic approach is presented to solve the model. However, a PESP instance reformulated this way is feasible if and only if (1.14) has value zero and the experimental results presented in [63] suggest that one must be very lucky to 'guess' a starting solution for the heuristic that results in the desired value (if it exists, of course).

Apart from the models mentioned above it is also worth mentioning some other models in relation to the timetable structure model. In [126] several extensions of the timetable structure model are discussed, such as allowing each constraint to have its own period or adding side-constraints on the phase shift in order to establish a sequential ordering of a subset of the event times, see also [121] where such side-constraints are used to model that trains cannot overtake each other on a single track. Furthermore, for a given phase shift the problem to find a timetable is known as the Feasible Differential Problem (FDP) [111]. What's more, in Chapter 3 we solve the PESP by solving a sequence of FDPs.
Characteristic for our timetable structure model is the absence of resource constraints dealing with the availability of rolling stock. Several authors (e.g. [12, 13, 58, 59, 104]), however, deal with the particular problem of minimizing the number of vehicles (i.e. trains in our case) to meet a given periodic timetable and they have developed models and solution procedures for this purpose. For instance, in [58, 59] it is shown that the minimum number of vehicles required to meet a timetable (not necessarily periodic) equals the sum of the maxima of so-called deficit functions for each of the terminals (i.e. stations). For a given timetable, a deficit function for a terminal expresses the number of vehicles present at that terminal as a function of the time. Thus, its maximum can be interpreted as the maximum number of vehicles present at the terminal at any time. In [60] the a solution procedure for the PESP and the deficit function approach are combined in an interactive scheme to find good timetables using as few vehicles as possible. We stress, however, that, since we assume that rolling stock can be abundantly available in ten to fifteen years from now and that, since we are primarily interested in designing a robust and high-quality railway system, this approach is not very appealing to our situation. Finally, in relation to the assignment of limited resources to periodic activities we mention the Extended PESP, that can be used to model resource constraints when scheduling jobs on machines [126], and furthermore [78], which contains a brief survey on this subject.

As to related models for railway timetable planning we mention two approaches, developed in 1983 in Germany [145] and in 1990 in Austria [81, 82], respectively. The German model aims at setting up a periodic timetable with fixed period $T$ such that network caused waiting times at stations are minimized, where passenger flows are used as weights in the objective function. For this purpose a graph $G$ is constructed whose vertices are the arrival and departure times of the trains for a single period and whose arcs denote traveling. There are two types of arcs. One type is associated with traveling between stations, i.e. from a departure from a station to the arrival at the next station, whereas the other type denote traveling inside stations, i.e. from an arrival at a station to the departure from that station. Arcs that denote traveling between stations are equiped with a fixed driving time and arcs denoting traveling inside a station are associated with a variable waiting time that is to be determined by a minimization procedure. A weighted sum of the waiting times at stations is to be minimized, where passenger flows constitute the weights. This leads to an integer linear programming formulation that is solved heuristically.

Mathematically, the German model can be described as follows. Let $w_{ij}$ be the variable waiting time at stations caused by a positive difference between arrival time $\tau_i$ and departure time $\tau_j$, and let $d_{ij}$ be the fixed difference between departure time $\tau_i$ from a station and arrival time $\tau_j$ at the next station. Indices $i$ and $j$ run over the set of arrival and departure times. Also, let $c$ be an elementary circuit of $G$ with associated set of driving times $D_c$ between stations and set of waiting times $W_c$ in stations. For a timetable to be periodic it is required that the sum of the waiting times and the driving times associated with $c$ must be a multiple of period $T$, i.e.

$$
\sum_{k \in D_c} c_k d_k + \sum_{k \in W_c} c_k w_k + T \rho_c = 0 ,
$$

(1.15)
where the $w_k \in \mathbb{R}$ and $p_c \in \mathbb{Z}$ are the variables. The $c_k \in \{-1, 1\}$ are constants denoting arc direction. In addition it is required that

$$0 \leq w_k < T$$  \hspace{1cm} (1.16)

for all waiting times $k \in W_c$, because passengers will never wait more than a period $T$ at the same station in a periodic timetable. Since all waiting times in the network are fixed by a solution to the system of equalities like (1.15) and inequalities like (1.16) that correspond to the fundamental circuits of $G$ relative to some spanning tree, we arrive at the programming problem

$$\min \sum_{k \in W} q_k w_k$$

$$\text{s.t.} \quad \sum_{k \in D_k} c_k d_k + \sum_{k \in W_c} c_k w_k + T p_c = 0 \quad c \in F$$

$$0 \leq w_k < T \quad k \in W$$

$$w_k \in \mathbb{R}_+ \quad k \in W$$

$$p_c \in \mathbb{Z} \quad c \in F,$$  \hspace{1cm} (1.17)

where $F$ is the set of fundamental circuits of $G$, $W$ is the set of waiting times, and $q_k$ is the number of passengers that travel over the $k$-th arc.

Timetabling problem (1.17) is heuristically solved in [145] by choosing an initial spanning tree and assigning zero waiting time to all arcs on that tree. From there the waiting times at non-tree arcs follow from (1.15) and (1.16). Then a search starts for the spanning tree that yields the lowest objective value by iteratively exchanging a tree arc for a non-tree arc if this yields an improvement.

The fundamental difference with the timetable structure model that we use is that for the German approach timetable feasibility is not an issue. That is, the initial solution is feasible by construction and successive improvements are made such that feasibility is maintained. On the other hand, even the best solution to (1.17) may include large waiting times at some stations for some trains, whereas in our timetable structure model waiting times can be guaranteed to be small if one wishes to specify so. As to the complexity of the German model, it induces a mixed integer linear programming problem with a number of integer variables that equals the number of non-tree arcs in the constraint graph. In this respect the German model resembles our model.

The second foreign model for periodic railway timetabling that we pay attention to is the so-called Integrierte Taktfahrplan (In English: integrated fixed-interval timetable), developed in Austria in 1990 [81, 82]. This model is also used in e.g. Switzerland [41]. According to such integrated fixed-interval timetable all trains are at a station at fixed and equidistant moments on the clock, e.g. at minutes 0/30 or 0/15/30/45. Then, travellers can always transfer from any train to any other train at those moments. The underlying idea of such system is to avoid waiting times at stations. However, as we will see below, this imposes severe restrictions on driving times between stations, thereby possibly implying huge investments in railway infrastructure.

The variables in the Austrian model are the driving time reductions on the connecting tracks between stations. With $A$ we denote the set of those tracks and, for track $k \in A,$
variable \( \delta_k \) is the required driving time reduction for trains that run on track \( k \) (it is assumed that the driving times only depend on the tracks and not on the trains). Furthermore, for \( k \in A \), \( d_k \) is the current driving time, so that \( d_k - \delta_k \) is the actual driving time that must be realized according to the resulting timetable. A circuit in the railway network is a closed sequence of tracks that does not intersect itself. Modeling the tracks of the railway network as a graph, a circuit in the railway network is an elementary circuit. It is important to understand that we mean the physical railway network and not an abstraction of it like \( G \) in the German model and in our model. With \( F \) we denote the set of fundamental circuits. In order to enforce that the timetable has period \( T \), which is fixed, and that all trains calling at a certain station do so at the same time it is required that

\[
\sum_{k \in c} c_k (d_k - \delta_k) + T p_c = 0
\]

(1.18)

for all circuits \( c \in F \) and

\[
d_k - \delta_k - \frac{T}{2} n_k = 0
\]

(1.19)

for all tracks \( k \in A \), where the \( p_c \) and \( n_k \) are integer variables. Again, the \( c_k \in \{-1, 1\} \) are constants and denote arc direction.

In the Austrian model an integrated fixed-interval timetable is sought that imposes minimal investments in railway infrastructure to execute it. Suppose that to reduce driving times on track \( k \in A \), there is a set \( M_k \) of measures that can be taken. Furthermore, implementing measure \( m_{kh} \in M_k \), \( h = 1, \ldots, \# M_k \), costs \( r_{kh} \) and measure \( m_{kh} \) by itself reduces the driving time with \( q_{kh} \), independent of other measures. This brings the total costs to

\[
\sum_{k \in A} \sum_{h \in M_k} r_{kh} z_{kh} .
\]

(1.20)

To express that measure \( m_{kh} \) is taken or not a boolean variable \( z_{kh} \) is introduced. Hence, in order to realize driving time reduction \( \delta_k \) on track \( k \), it is necessary that

\[
\sum_{h \in M_k} q_{kh} z_{kh} \geq \delta_k .
\]

(1.21)

Then, the overall timetabling problem reads

\[
\begin{align*}
\text{min} & \quad \sum_{k \in A} \sum_{h \in M_k} r_{kh} z_{kh} \\
\text{s.t.} & \quad \sum_{k \in c} c_k (d_k - \delta_k) + T p_c = 0 & & \quad c \in F \\
& \quad d_k - \delta_k - \frac{T}{2} n_k = 0 & & \quad k \in A \\
& \quad d_k \geq \delta_k \geq 0 & & \quad k \in A \\
& \quad \sum_{h \in M_k} q_{kh} z_{kh} \geq \delta_k & & \quad k \in A \\
& \quad z_{kh} \in \{0, 1\} & & \quad k \in A, \ h \in M_k \\
& \quad \delta_k \in \mathbb{R} & & \quad k \in A \\
& \quad n_k \in \mathbb{Z} & & \quad k \in A \\
& \quad p_c \in \mathbb{Z} & & \quad c \in F .
\end{align*}
\]

(1.22)
The driving times between stations are the unknowns to be solved from this system, in compliance with the Swiss viewpoint that the timetable comes first and pushes technology and resource availability. Thus, if it turns out that driving times must be shorter than the current state of technology admits, track modifications are necessary. In our timetable structure model driving times are fixed within given (and often tight) ranges that are defined on the basis of what the current state of the tracks allow. In [81, 82] a branch-and-bound technique is used to solve (1.22).

Since an integrated fixed-interval timetable imposes severe restrictions on the infrastructure and the rolling stock, in the Rail 2000 program [41] 130 kilometers of new track connections, faster rolling stock, and up-grading of existing tracks were included in order to establish the necessary driving time reductions. Obviously, this means that expensive infrastructure is built on the basis of a specifically chosen timetable concept, thus creating a rigid system. Moreover, railway stations must be equipped with a lot of tracks and platforms in order to handle all the trains that come and go at the same time. It should be kept in mind, however, that in Switzerland many choices are already made, thereby reducing the uncertainty about the future timetable considerably. As a contrast, in the Dutch Rail 21 program the infrastructure is designed to be robust with respect to future scenario's and for this purpose the timetable structure model is suited much better.
Chapter 2

Mathematical Background and Preliminaries

For a large number of problems it is recognized that placing parts of algorithms that solve them under the control of a randomized mechanism may increase the efficiency and effectiveness of computations a great deal. Moreover, randomization can also save a lot of computer memory as the amount of information that must be kept track of is often small compared to deterministic algorithms. Algorithms that are (partly) driven by stochastic forces are called randomized algorithms.

In this thesis we will develop and apply randomized algorithms in several ways, so that it is appropriate to mention some of the background of such algorithms. In [75, 91, 147] randomized algorithms are surveyed to much greater extent, [91] being one of the first textbooks in this area.

As to the relevance of this chapter to other parts of this thesis, in Chapters 4 and 5 we will discuss sampling schemes for generating railway timetables at random from specific probability distributions. In Chapter 6 we describe a powerful random walk technique for finding low cardinality covers of Steiner triple systems.

The outline of this chapter is as follows. In the next section we discuss some common features of randomized algorithms and refer to some landmarks in this field. We stress that the discussion is not meant to be tutorial nor to give a complete review of the literature. In Section 2.2 we address the problem of sampling at random from a convex body, a problem that, in a special form, returns in Chapter 4. Thereafter, in Section 2.3 the use of Markov chains to sample randomly from convex or discrete sets is discussed. This technique will be exploited in Chapter 5. Finally, Section 2.4 contains some mathematical preliminaries that we use throughout, although most of them are already commonly understood or speak for themselves.
2.1 Randomized algorithms

Although the interest in randomized algorithms goes back to the beginning of computer science, one of the first genuine applications of randomization in algorithms was proposed by Rabin [107] in 1976 and by Solovay and Strassen [133] in 1977. Both applications were concerned with primality testing, i.e. the problem of deciding about the primality of natural numbers. Rabin [107] gave some more examples involving number theory and computational geometry.

Most randomized algorithms are surprisingly simple, both conceptually and as far as implementation is concerned. Take for example primality testing. Whereas this problem is hard to solve deterministically, it can be shown that for numbers that are not prime it is very likely to find a ‘witness’ for compositeness by repeatedly sampling numbers from a suitably chosen probability space. Thus, such an algorithm always gives the right answer YES whenever the given number is prime and returns YES whenever it should say NO to primality with low probability.

Algorithms like this one for primality testing are referred to as Monte Carlo algorithms. Typically, when answering decision problems, they are allowed to err with some probability in case the answer should either be YES or NO (one sided error) or in both situations (two sided error). Running a Monte Carlo algorithm often consists of a number of identical random experiments and returning a correct answer if in any of the repetitions a witness for some alleged property of the problem instance under study is found. Usually, the number of runs and sometimes the duration of a run are parameters of such algorithms and they determine the efficiency, the effectiveness, and the failure probability to large extent. Analysis of Monte Carlo algorithms, therefore, tends to focus on quantifying the relations between these measures.

In Chapter 6 we present a Monte Carlo algorithm to find set covers of low cardinality for the Steiner Triple Set Covering Problem. A Steiner triple system is a set of distinct objects $S$ and a family $F$ of subsets of cardinality three such that every pair of objects is contained in precisely one such subset. The problem is to find a subset $\tilde{S}$ of objects such that every member of $F$ has at least one element in $\tilde{S}$. The Monte Carlo algorithm we use for this problem is surprisingly effective considering its simple appearance. It consists of a specified number of random walks of specified length through the solution space, keeping the best result in memory. Note that this problem can be converted into the decision problem of deciding about the existence of a cover of a certain cardinality and that in this application the failure probability, i.e. the probability to give a false answer, is not a control parameter, whereas with primality testing the failure probability can be bounded from above beforehand and determines the number of experiments.

Another type of randomized algorithms are the so-called Las Vegas algorithms. They always pass the right answer or result, however at the expense of a stochastic running time. Consequently, analysis of such algorithms often involves establishing a bound on the expected running time. Consider, for example, an algorithm to sort objects (given a consistent ordering) that proceeds by selecting an object at random, separating the objects that are smaller from the ones that are larger, and repeating this procedures recursively.
on those two subsets. Clearly, the objects will eventually be sorted correctly, but the running time required to do so depends on the choices that are made during the course of execution. It can be shown [75] that when these ‘splitters’ are chosen uniformly at random the expected running time (i.e. number of comparisons) is $2n \ln(n) + \mathcal{O}(n)$, which is quite good if we realize that $n \log_2(n)$ is a theoretical lower bound.

Note that the definitions of Monte Carlo and Las Vegas algorithms are taken from [91] and that in [75], for instance, a Las Vegas algorithm is allowed to fail to give an answer with probability at most $\frac{1}{2}$.

In [75] further distinction is made by mentioning so-called bounded-error randomized algorithms. Such algorithms give a correct answer with probability at least $\frac{1}{2} + \epsilon$ for some fixed $0 < \epsilon < \frac{1}{2}$ but there is no indication whatsoever as to whether or not an answer is correct. By running the algorithm a couple of times and taking the most frequent answer the probability that this answer is incorrect can be driven to zero, thereby amplifying the probability of ending up with a correct result.

Whereas Monte Carlo and Las Vegas algorithms most of the time give verifiably correct answers to decision problems, another type of randomized algorithms produce at best approximate answers to numerical problems. In this respect, the notion of correctness of an answer looses its meaning, because answers will always be accepted as correct, give or take some error, the analysis of which is important to evaluate the performance of the algorithm. Examples include randomized approximation schemes, which are algorithms for estimating a well defined quantity. For example, in Chapter 5 we address the problem of counting the number of railway timetables, which comes down to estimating the number of integral points in a polytope. Also, in the remainder of this chapter we will see another example, namely estimating the geometric volume of convex bodies. An algorithm $M$ of this kind that produces a numerical output $x$ approximating some quantity $y > 0$ often does so in compliance with

$$
Pr\left(\frac{|x - y|}{y} > \epsilon\right) \leq \delta ,
$$

(2.1)

or some other appropriate measure of effectiveness, where $\epsilon \geq 0$ and $0 < \delta < 1$. If the running time of $M$ is bounded by a polynomial in the size of the input, $M$ is called a polynomial randomized approximation scheme and, if, moreover, its running time is even bounded by a polynomial in the input size, $\epsilon^{-1}$ and $\ln(\delta^{-1})$, then it is called a fully polynomial randomized approximation scheme. Establishing a bound on the running time given the problem size, $\epsilon$, and $\delta$ is normally what analysis of such algorithms is about.

In Chapters 4 and 5 we will encounter algorithms to sample elements from sets according to a given probability distribution. Such algorithms are called sampling schemes and it is needless to say that they are inherently stochastic. In some cases it is possible to sample from the target distribution exactly, of course give or take an error on account of pseudo randomness of the random generator used. In Chapter 4 we will see an example of exact random sampling and we will discuss this further in the next section. On the other hand, it is not always possible to sample precisely according to the target distribution. This phenomenon will be encountered in Chapter 5 and will be further discussed in Section 2.3.
Typically, for a discrete probability distribution $f$ such algorithm $M$ produces a sample $x$ for which
\[
\frac{\left| \Pr(M \text{ outputs } x) - f(x) \right|}{f(x)} \leq \epsilon,
\]
for given parameter $\epsilon$. If $f$ is the uniform distribution, this kind of sampling is referred to as almost uniform sampling (or almost uniform generation) and can be generalized to continuous probability densities. Once again, analysis tends to relate the problem size and $\epsilon$ to the running time.

An intriguing relation between random sampling and counting was established by Jerrum et al. [68] who showed that in many cases almost uniform generation and approximate randomized counting are inter-reducible, thus making them equally difficult from computational point of view. This result applies to combinatorial structures that are self-reducible in the sense of [119]. In Chapter 5 this result is used to state an algorithm for counting the number of feasible timetables given a phase shift, see Chapter 1. See e.g. [51] for another example of approximate counting using Markov chain sampling. Another noteworthy application of sampling from stochastic distributions on discrete sets is the use of samples to experimentally analyze the average case performance of algorithms, see e.g. [69].

A common argument for applying a randomized algorithm while in principle some deterministic algorithm could also perform the computation correctly is what is referred to as foiling an adversary. This argument stems from the game-theoretic viewpoint that, if an adversary may choose a problem instance and receives a pay off equal to the running time of an algorithm that simultaneously is our pick, our adversary will have a hard time to come up with a problem instance that is likely to perform poorly if we choose our algorithm at random. As mentioned above, another important argument in favor of randomized algorithms is that they are often simple and easier to implement than their deterministic counterparts.

Of course, randomized algorithms also have drawbacks. a big one being that at best probabilistic statements about the outcome of randomized algorithms can be made. For example, Monte Carlo algorithms have the property that the running time increases when tolerance with respect to failure decreases. In practice, by randomizing parts of algorithms, control is given away to a random generator, so that the state and course of the algorithm are sometimes hard to understand or predict. Notably, random running time or not knowing the quality of an answer can be very frustrating.

To date, a general theory about the merits of randomization in algorithms does not exist. Survey papers like [75, 91, 147] tend to discuss results in the field of randomized algorithms by referring to a relatively small number of applications such as fingerprinting (comparing objects by randomly selected fingerprints), checking identities (plugging in values into an algebraic expression such that it is very likely that a witness of falsehood is found if the identity does not hold), sampling from stochastic distributions, sorting objects, computational geometry, number theory (e.g. primality testing), random graph theory (proving properties of a certain class of graphs), or symmetry breaking in distributed computations.
As to negative results, the theory of NP-completeness has been extended by introducing the Probabilistic Turing Machine as a formal computational model, see e.g. [56, 70, 147, 148]. In 1977, a paper by Gill [61] laid the foundations for this new branch in complexity theory. From complexity theory it becomes apparent how much randomization of computations may help to tackle some specific problems that are very hard to solve deterministically, albeit at the price of exactness and/or certainty about obtaining an answer at all. The pool of famous examples include e.g. computing the permanent of 0-1 matrices (see Section 2.3), computing the volume of convex bodies (see Section 2.2), or counting various combinatorial objects from implicit descriptions. See also Chapter 6 for another successful application.

### 2.2 Sampling from convex bodies

Given a full-dimensional, convex, bounded, and closed subset $C$ of $\mathbb{R}^n$, which we call a convex body, many questions can be asked about it, the classical ones having to do with

**Existence.** Is $C$ non-empty?

**Construction.** Return an element of $C$, or establish that $C$ is empty.

**Optimization.** Given a mapping $c \in C \rightarrow \mathbb{R}$, return an element $y \in C$ such that $c(y) \leq c(z)$ for all $z \in C$, or establish that $C$ is empty.

**Volume calculation.** Return the geometric volume of $C$ (Lebesgue measure).

Literature about these topics grows faster than anyone can read. When linear programming became popular right after the Second World War, the number of papers on it became overwhelming. The interest in volume calculation goes back to ancient history. As to the complexity of the problems, there exists a large gap between the existence, construction, and optimization problems on the one side and volume calculation on the other side. For example, if $C$ is a polytope and $c$ is linear, then the existence, construction, and optimization problems are linear programming problems, whereas for computing the volume of polytopes negative complexity results exist, see e.g. [11, 24, 42].

Another question that has received much less attention so far and that is addressed in this section and in parts of this thesis is

**Random generation.** Return an element of $C$ at random according to a given probability distribution on $C$ (with density $f$), or establish that $C$ is empty.

As to its complexity in relation to the other questions, the random generation problem is at least as difficult as the existence and construction problems and in [68] it is even made plausible that random generation is indeed computationally harder than construction. In a series of recent papers randomized approximate volume calculation is reduced to almost uniform generation, see e.g. [71].

As mentioned in [131], if $f$ is a probability density on $C$, sampling from $f$ can be reduced to uniform generation from the region under the graph of $f$, i.e. $\{(z, x) \in \mathbb{R}^n \times \mathbb{R} \mid x \leq f(z)\}$, and projecting the sample onto $C$. Since this principle applies quite general, there
is not much generality lost if we confine ourselves to uniform generation from $C$.

Three well-known and appealing techniques exist for the purpose of uniform generation from $C$, see e.g. [114, 131]. These are the transformation technique, the composition technique, and rejection sampling. The transformation technique is known to be only applicable to bodies that can be transformed to geometrically simple bodies such as cubes, rectangles, spheres, parallelepipeds, and simplices, by a one-to-one mapping $T$. It consists of two stages, namely generating a point from such a simple body and mapping it onto $C$ using $T$. In Chapter 4 this technique is applied to randomly generate railway timetables.

Simply put, with the composition technique one partitions $C$ into simple bodies like mentioned in the above, selects at random one of them, and then samples from the selected body. See e.g. [64] where the technique is applied to uniformly generate points from a polygon. The number of elements of the partitioning or the shape of $C$ limits the applicability of this technique.

Finally, rejection sampling [114, 115] employs the idea that, if a point $y$ is generated from a simple body that encloses $C$ happens to be in $C$ as well, it can be accepted as a uniform sample from $C$. If it is not in $C$, we reject $y$ and repeat the procedure. Although conceptually sound, it can be very inefficient as can be seen from the classical example of the hypercube enclosed by the hypersphere where the ratio of the respective volumes goes to zero at a rate that is exponential in the geometric dimension.

There is a growing list of applications where randomly generated objects are required as part of the input or as driving mechanism of randomized algorithms. To mention a few: estimation of integrals [8, 20, 114, 136], approximate volume calculation of convex bodies [8, 43, 44, 45, 50, 71, 72, 84, 85, 86], global optimization (random search methods) [27, 109, 110, 112, 132], nonredundant constraint identification [16, 29], or sampling from specific distributions [57, 114, 131].

In Chapters 4 and 5 we are interested in random generation for the purpose of uncertainty and sensitivity analysis, see e.g. Meeuwissen [88]. In an abstract sense, many systems take exogeneous parameters as input that are uncertain, e.g. because they represent future quantities. This problem can be handled by sampling at random from a specific distribution a number of values for the uncertain input parameters and study the systems behavior. Sensitivity analysis means that, given settings for all parameters, one perturbs the exogeneous parameters a little (because they are uncertain) and studies the changes in the systems behavior. As to specifying railway infrastructure, we generate several reference timetables for the purpose of uncertainty analysis, see Chapter 4, and, given a station track layout design and the set of reference timetables the design is based on, we sample timetables that are slightly different from those reference timetables for the purpose of sensitivity analysis, see Chapter 5 and [99].

### 2.3 Markov chain sampling

For obvious reasons the methods described in the previous section are often not applicable to general convex bodies. At present the best general purpose algorithms known are
approximative and based on Markov theory. The idea is to approximate the uniform distribution on $C$ by devising a Markov chain $\mathcal{M}$ whose state space $S$ is $C$ (or a lattice enclosed by $C$) and whose equilibrium and limiting distribution is the uniform distribution. In a series of papers, starting in 1988 with Dyer et al. [43, 41] and followed by Applegate and Kannan [8], Lovasz and Simonovits [84, 85, 86], the running time complexity has been brought down from $O(n^{19})$ to $O(n^6)$ membership tests, where $n$ is the dimension of $C$, and a claim on a better performance was made by Kannan et al. [71] in 1994 and later in 1996 [72]. The approach is known under the name of Markov chain sampling or random walk sampling.

Roughly, we distinguish between two types of Markov chains. The first one is a nearest neighbor random walk on the lattice $S = C \cap \mathbb{Z}^n$, where $\delta > 0$ is a small number and $\mathbb{Z}^n$ is the grid on $\mathbb{R}^n$ with mesh size $\delta$, i.e. $\delta \mathbb{Z}^n = \{ \delta \mathbb{Z} \mid z \in \mathbb{Z}^n \}$, see e.g. [8, 43, 44, 85]. The second type takes $S = C$ and proceeds by selecting in the current state $x$ the following state from a ball with center $x$ and with small radius, see e.g. [81, 86]. This type is therefore referred to as the ball walk. Another form of Markov chain sampling with continuous state space is the so-called Hit-and-Run approach. In this method a continuous state space Markov chain is constructed by iteratively selecting at random a line through the current state and then selecting at random the new state from the line segment that intersects $C$ [14, 16, 29, 76, 131].

Markov chain sampling also works for sampling from discrete sets. For example sampling from group structures [2, 3, 36, 37, 39, 105], sampling from general combinatorial structures [4, 89, 128, 130], sampling substructures from graphs [5, 26, 51], discrete optimization [1], linear programming [46], physics [65], or statistics [38].

A notable example for which Markov chain sampling adds to reduce complexity and that received much attention in literature is the computation of the permanent $\text{per}(A)$ of an $n \times n$ matrix $A$, which is defined as

$$\text{per}(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} a_{i,\pi(i)},$$

(2.3)

where $S_n$ is the set of permutations of $n$ elements. Whereas this definition resembles the definition of the determinant, there is a large complexity gap when it comes down to computing them [137, 138]. In fact, the best known algorithm for computing the permanent dates from 1963 and runs in $O(n2^n)$ time [116]. Even restricted to 0-1 matrices computing the permanent is $\#P$-complete, thus motivating the search for approximate methods. In this direction we mention a Monte Carlo method yielding an approximation scheme that requires $O(2^{n/2}\epsilon^{-1}\ln(\epsilon^{-1}))$ time, however at the expense of losing exactness [73] and still being bound to exponential running time. A breakthrough was established in 1986 when Broder [25], followed by Jerrum and Sinclair [66], also in 1986, used the equivalence of computing the permanent of a 0-1 matrix and counting the number of perfect matchings in bipartite graphs to devise a Markov chain sampling approach to derive a fully polynomial randomized approximation scheme for special cases. However, establishing such a result for general cases remains an open problem. See also [65, 67].
In [71] an overview of Markov chain sampling, including more details and many references, is presented. Reference [148] also contains a chapter devoted to Markov chain sampling. Below we will highlight some of the details. For sake of comprehensiveness we will only discuss the discrete state space Markov chain sampling technique in more detail.

Convergence of the Markov chain to the uniform distribution is implied by symmetry and ergodicity [47]. Symmetry is often established by construction. Ergodicity means a-periodicity and irreducibility. The former property can be enforced by equipping each state with a self-loop probability, usually of 1/2, although this is a technicality that often can be left out in practice. Obtaining irreducibility, or equivalently connectedness of state transition graph \( G \), often forms the hard part. For example, with the discrete state space approach, when walking through a convex body, irreducibility can only be realized by making the step size \( \delta \) small enough, whereas in [25, 66, 128] it is shown how an irreducible random walk through the space of perfect and near perfect matchings of a given graph can be constructed without much effort.

To assess the number of steps required to approach the uniform distribution close enough, many papers apply in some sense the Perron-Frobenius theory stating that the distance to uniformity \( \Delta(t) \) approaches zero (as the number of steps \( t \) goes to infinity) at an exponential rate with base equal to the second largest eigenvalue of the transition matrix [125]. In [128, 130] and many other references \( \Delta(t) \) is measured by the relative pointwise distance defined as

\[
\Delta_{rp}(t) = \max_{i,j \in S} \left| \frac{p_{ij}^{(t)} - 1/N}{1/N} \right| ,
\]

where \( p_{ij}^{(t)} \) is the \( t \)-step probability of going from state \( i \) to state \( j \), and \( N \) is the cardinality of \( S \). Other measures are also used. For example, in [105] the total variation distance is used, which is defined as

\[
\Delta_{tv}(i) = \frac{1}{2} \sum_{i \in S} \left| p_i^{(t)} - \frac{1}{N} \right| ,
\]

where \( p_i^{(t)} \) is the \( t \)-step probability distribution, given a starting distribution. Note that \( \Delta_{rp}(t) \) does not depend on the starting distribution, but only on the transition probabilities.

The efficiency of Markov chain sampling algorithms, or equivalent the rate of convergence to the uniform distribution (or mixing rate as it is often called, a term introduced by Aldous [2] in 1983), depends on the structure of \( G \) [128, 130]. Many proofs of the rapid mixing property rely on the concept of conductance of \( G \), following a result from Sinclair and Jerrum (1989) [128, 130] in which the second largest eigenvalue of the state transition matrix is bounded by using the conductance of \( G \). In words, \( G \) is said to have high conductance (from which rapid mixing follows) if the probability that the Markov chain gets stuck in a subset of state space \( S \) for a long time is very small. In a more formal sense, rapid mixing means that the time needed to approach the uniform distribution is bounded by a polynomial in the parameters that generate \( S \) and the logarithm of the reciprocal of the (relative) error allowed. For example, when generating spanning trees in a given simple connected graph with \( N \) vertices, rapid mixing means that the Markov chain settles
down in a number of steps that is bounded by a polynomial in $N$ and the logarithm of the reciprocal of the allowed relative deviation $\epsilon$ from the uniform distribution.

Showing high conductance is usually the obstacle when proving that the random walk is efficient. In case of uniform sampling from convex bodies by means of the discrete state space approach, high conductance follows from a proper isoperimetric inequality. For a body $C$ such an inequality puts a universal bound on the ratio of volume and surface of any measurable body contained in $C$. Put differently, an isoperimetric inequality guarantees that for any measurable body $S \subseteq C$ the size of its surface cannot be too small relative to its volume thereby implying that a random walk through $C$ has enough possibilities to ‘escape’ from $S$ once it gets there. The difficulty lies in finding a proper isoperimetric inequality. Another useful inequality to bound the second largest eigenvalue is the Poincaré inequality [39].

Aldous and Diaconis [2, 3] noticed for some examples the so-called cut-off phenomenon, meaning that $\Delta(t)$ adopts the form of a step function in the sense that at a certain time $\tau$ it suddenly drops from almost one to almost zero. Then, they propose to stop the Markov chain just after time $\tau$. In this respect a well-known example is card shuffling, see also [105].

We mention that, instead of conductance, closely related graph theoretical concepts such as expansion or resistance [6, 7, 89, 129] are also sometimes used. Like with conductance, these concepts are a sort of graph theoretical analogue of the isoperimetric inequality. For example, a graph is said to be an expander if any subset of vertices is connected to its complement in the graph through many edges [6]. Thus a random walk on the vertices of an expander will mix rapidly.

As to Markov chain sampling from a continuous state space, establishing convergence to the uniform distribution involves more technical details. See e.g. [86], where Markov theory is extended to abstract continuous state spaces with atoms.

Although the rapid mixing property has been established for convex bodies, Markov chain sampling is rather inefficient, witness the leading term constant of about $10^{10}$ in the running time polynomial of [86]. However, at present there are no alternatives that are provably more efficient for general convex bodies. It is conjectured by Stougie [134] that Hit-and-Run sampling also has polynomial runtime and is at least as fast as the ball-walk.

In Chapter 5 we apply the Markov chain sampling method to randomly generate timetables, although establishing irreducibility of the state transition graph appears to be non-trivial. Unfortunately, it is not difficult to show by example that this method does not yield a polynomial time algorithm.

The Markov chain sampling method for uniformly generating points is certainly not limited to convex bodies alone. In fact, convexity is not essential for convergence to the uniform distribution. However, for proving the rapid mixing property something like convexity is necessary. To get a feeling for this, think of a region consisting of two convex bodies, a small one and a large one, connected by a small full dimensional tunnel. A particle, representing the state of the Markov chain and starting in the larger convex part, will have great difficulty to ever reach its smaller brother, thus getting stuck in a subset of the entire region. Furthermore, Markov chain sampling can also be used to randomly
generate points on the boundary of $C$ [17, 20].

To make Markov chain sampling of practical use one has to be able to determine how many steps a simulation should run in order to be sure that the final state has a distribution close enough to the equilibrium distribution. Basically, we can do this in two different ways, one of which involves the determination of an upper bound on the number of steps beforehand, using distance measures and bounding techniques mentioned in the above. However, such bounds tend to be too large to be practical, even if they are polynomial in the input size of the problem at hand.

The other way to determine when to stop a simulation of a Markov chain is to let it run and decide on the basis of intermediate observations of the state whether to stop or not. In this respect, recently Propp and Wilson [106] made significant progress with a method to sample exactly (or to simulate perfectly, as it is also referred to). They propose to simultaneously start up a simulation for each state in the state space and identify two simulations whenever they reach the same state at some point. The procedure stops the moment all simulations have merged into a single simulation. Then, the resulting coalescent state is then output as the sample, for which it can be shown that it is unbiased. Moreover, the algorithm terminates in finite (but random) time. The pitfall, however, is that in most practical cases too many computer memory is required to keep track of all the simulations. In [106] it is shown that, if the state space admits a partial ordering $\preceq$ of its elements and, if for any pair of simulations $(\{X_t\}_{t \geq 0}, \{Y_t\}_{t \geq 0})$ it holds true that $X_t \preceq Y_t$ implies $X_{t+1} \preceq Y_{t+1}$ for all $t \geq 0$, then in finite random time the state flows coalesce and the coalescent state is exactly the equilibrium state. This is called monotone Monte Carlo sampling. In [106] perfect simulation is applied to statistical mechanics and the sampling of e.g. permutations of a set of items and independent sets in graphs. See also [77] for another application.

2.4 Notations and conventions

The notations used in this thesis are commonly encountered in literature so that it suffices to give only some brief comments.

With $\mathbb{R}$ ($\mathbb{Z}$) we denote the real numbers (integers) and with $\mathbb{R}_+$ ($\mathbb{Z}_+$) the non-negative numbers (integers) among them. For $n \in \mathbb{Z}$, the sets $\mathbb{R}^n$, $\mathbb{R}_+^n$, $\mathbb{Z}^n$, and $\mathbb{Z}_+^n$ denote Cartesian products in the usual sense.

Vectors $e$ and $e_i$ are the all-one vector and the $i$-th unit vector of appropriate sizes, respectively. By $I_n$ we denote the $n \times n$ unit matrix.

Unary or binary operations on scalars are component-wise applicable to vectors. For vectors $x$ and $y$, the same applies to $\max(x,y)$ and $\min(x,y)$. The expression $x \geq y$ ($x > y$) means that every entry of $x - y$ is (strictly) positive. Also, $x^+ = \max(0,x)$ and likewise $x^- = -\min(0,x)$.

$\#A$ is the cardinality of a finite discrete set $A$. For sets $A$ and $B$ and numbers $\alpha$ and $\beta$ we have $\alpha A + \beta B = \{\alpha a + \beta b \mid a \in A, b \in B\}$. If $A \in \mathbb{R}^{n \times m}$ is a matrix and $X \subseteq \mathbb{R}^m$ is a set, then $AX = \{Ax \mid x \in X\}$.

On some occasions we write $(a,b)$ for $(a^T,b^T)^T$ for sake of brevity.
Part II

Railway Timetable Generation
Chapter 3

Periodic Timetables: Model, Analysis, Algorithms, and Complexity

In this chapter\(^1\) we introduce and discuss the timetable structure to model and construct periodic railway timetables and show how this construction can actually be done. In the model departure and arrival times of trains are related pairwise on a clock by means of periodic time window constraints. A solution to a set of such constraints, then, constitutes a periodic timetable.

The timetable structure is discussed before in [126] where solving it is referred to as the Periodic Event Scheduling Problem (PESP). Several applications are known, e.g. airline scheduling [60], traffic light scheduling [127], and job shop scheduling [126]. Here we exploit the timetable structure and the PESP further by using it to model and construct periodic railway timetables. Also, we present a cut generation algorithm for solving the PESP and discuss some other methods known from literature.

The immediate motivation for studying the problem of constructing railway timetables stems from the need to have a method to generate timetables, which, once generated, can be used to specify railway station infrastructure (i.e. its track layout) that is needed to handle future traffic, see Chapter 1. However, apart from using timetables this way, we may want to construct timetables for other reasons as well, notably when timetables must be revised due to long lasting railway track maintenance projects.

The remainder of this chapter is organized as follows. In the next section we explain about timetable structures and discuss modelling issues. Also, we introduce the PESP formally. In the following two sections we discuss its complexity, and outline a new algorithm to solve PESP, i.e. to extract a timetable from a timetable structure, if possible. Section 3.4 is devoted to a non-trivial example to demonstrate the applicability of the theory. This example will be further used in Chapter 4.

\(^1\)This chapter is a revised version of 'M.A. Odijk: A Constraint Generation Algorithm for the Construction of Periodic Timetables' in *Transportation Research* 30B (6), pp. 455-464, 1996. In particular, Section 3.2 is added.
The timetable structure model that we discuss in this chapter is precisely the model that we introduced in Chapter 1 under the same name. As to the relation of this chapter with the other chapters in this part of the thesis, here we discuss modeling aspects and methods to construct periodic timetables deterministically, given a timetable structure. In Chapters 4 and 5 we address the problem of constructing such timetables in compliance with a given probability distribution on the set of all feasible timetables, again, given a timetable structure.

3.1 The timetable structure model

To formalize the notion of a timetable structure we adopt the mathematical model presented in [126]. Here we discuss the model and some of its modeling power, all with respect to the railway application. We refer to the model as the timetable structure model. A particular instance of the timetable structure model is called a timetable structure.

The variables in the timetable structure model are the event times that correspond with the arrival and departure times of trains at stations. For example, if train $t$ passes station $s$, the timetable structure model contains an arrival time $\tau^a_{is}$ and a departure time $\tau^d_{is}$ for the timing of this event. Moreover, with the timetable structure model we model periodic timetables. That is, if train $t$ arrives at station $s$ at time $\tau^a_{is}$, it also arrives at times $\tau^a_{is} - T, \tau^a_{is}, \tau^a_{is} + T, \ldots$, where $T$ is the period of the timetable. Thus, $t$ represents in fact a series of trains that arrive at station $s$ at all times $\tau_{is}^a + T p_i^s$ for $p_i^s \in \mathbb{Z}$. In general, there are $N + 1$ event times in the timetable structure model and they are identified by the symbols $\tau_0, \ldots, \tau_N$. In vector notation we write $\tau = (\tau_0, \ldots, \tau_N) \in \mathbb{R}^{N+1}$.

In the timetable structure model several operating conditions and policies can be expressed by means of mathematical constraints on the event times. In abstract sense, these constraints place upper and lower bounds on the difference between pairs of event times. For instance, we might want to stipulate that the departure of train $t_1$ (event $i$) takes place between 3 and 5 minutes after the arrival of train $t_2$ (event $j$) takes place. More precise, we want to stipulate that the departure of some representative of the train series $t_1$ takes place between 3 and 5 minutes after the arrival of some representative of train series $t_2$. Thus, the timetable structure contains the constraint that

$$l_{ij} \leq (\tau_i + T p_i) - (\tau_j + T p_j) \leq u_{ij} \quad (3.1)$$

must hold for some $p_i, p_j \in \mathbb{Z}$. Clearly, this constraint is equivalent to stipulating that

$$l_{ij} \leq \tau_i - \tau_j + T p_{ij} \leq u_{ij} \quad , \quad (3.2)$$

for some $p_{ij} \in \mathbb{Z}$.

In general, the timetable structure model contains $M$ constraints like (3.2). Constraint (3.2) specifies that the event times of event $i$ and $j$ are related in a sense that the difference between $\tau_i$ and $\tau_j$ is in one of the intervals $t_{ij} - T, u_{ij} - T, [l_{ij}, u_{ij}], [l_{ij} + T, u_{ij} + T], \ldots$. Vector $p = (p_{ij}) \in \mathbb{Z}^M$ is called the phase shift because it states which of these intervals
embrace $\tau_i - \tau_j$. In the model, $T$ is a given and fixed integer denoting the period of any timetable satisfying the constraints. It is often set to 60 to express the desire to create hourly patterns. The numbers $l_{ij}$ and $u_{ij}$ are also given and fixed integers and the are such that $0 < u_{ij} - l_{ij} < T$.

The constraints in a timetable structure may stem from various practical requirements, see e.g. [139]. For one thing, a constraint might express that two trains must connect at a railway station. Think, for instance, of passengers transferring from one train to another, or two trains being physically coupled. Such constraints are called operating policy constraints. Another type of constraint, called infrastructure constraint, arises because the limited infrastructure does not allow for some simultaneous train movement or it enforces headway times between trains. For example, if train $j$ can leave the railway station no sooner than 2 minutes after train $i$ does and vice versa, a constraint like

$$2 \leq \tau_i - \tau_j + 60p_{ij} \leq 58$$

(3.3)

is part of the timetable structure ($T = 60$, $l_{ij} = 2$, $u_{ij} = 58$). Finally, there are frequency constraints that specify that trains belonging to the same family, i.e. trains having the same origin and destination, visit the railway station every half an hour in a semi-regular fashion. This can be modeled by defining two independent train series, the one with model train $i$ and the other with model train $j$, and coupling them, for example, by

$$28 \leq \tau_i - \tau_j + 60p_{ij} \leq 32$$

(3.4)

Note that, if a strict half an hour pattern is specified, i.e. $l_{ij} = u_{ij} = 30$, one of the events can be substituted out, because in that case event $j$ strictly follows the schedule of event $i$.

If a train family to be scheduled once an hour must connect to a train family with a frequency of twice an hour (in a strict 30/30 pattern), it matters to which of the two occurrences of the latter train family the train belonging to the former one is connected in the timetable structure. This problem can be overcome by adding a constraint like

$$3 \leq \tau_i - \tau_j + 60p^1_{ij} \leq 5$$  \quad \text{or}  \quad (3.5)$$

$$33 \leq \tau_i - \tau_j + 60p^2_{ij} \leq 35$$  \quad \text{or}  \quad (3.6)$$

where $\tau_i$ corresponds to an event associated with the train family with frequency one and $\tau_j$ to an event associated with a specific train of the train family with frequency two. More generally, we would like to be able to handle a constraint like

$$l^1_{ij} \leq \tau_i - \tau_j + Tp^1_{ij} \leq u^1_{ij}$$  \quad \text{or}  \quad (3.7)$$

$$l^2_{ij} \leq \tau_i - \tau_j + Tp^2_{ij} \leq u^2_{ij}$$  \quad \text{or}  \quad (3.8)$$

$$\vdots$$

$$l^K_{ij} \leq \tau_i - \tau_j + Tp^K_{ij} \leq u^K_{ij}$$  \quad \text{or}  \quad (3.9)$$
where $K > 1$ is an integer, $l_{ij}^1 < u_{ij}^1 < \ldots < l_{ij}^K < u_{ij}^K$, and $u_{ij}^K - l_{ij}^1 < T$. To this end, note that (3.7)-(3.9) is equivalent to

\begin{align}
    l_{ij}^1 &\leq \tau_i - \tau_j + Tp_{ij}^1 \leq u_{ij}^K \quad \text{and} \quad (3.10) \\
    l_{ij}^2 &\leq \tau_i - \tau_j + Tp_{ij}^2 \leq u_{ij}^K + T \quad \text{and} \quad (3.11) \\
    \vdots &\quad \vdots \\
    l_{ij}^K &\leq \tau_i - \tau_j + Tp_{ij}^K \leq u_{ij}^{K-1} + T \quad . \quad (3.12)
\end{align}

This equivalence is visualized in Figure 3.1 (taken from [126]). The bottom line is that

\[ \bigcirc \cup \bigcirc = \bigcirc \cap \bigcirc \]

\textit{Figure 3.1 : Using the model to model the union of disjoint intervals.}

with expressions like (3.2) we can model many temporal relations between pairs of event times that arise in practice.

The constraints in a timetable structure can be visualized by drawing up a directed graph $G$ in which each vertex represents an event and each arc a constraint. We call such graph a constraint graph and Figure 3.2 shows an example. In this example there are four

\[ \tau_1 \quad \text{[2,4]} \quad \tau_2 \quad \text{[51,54]} \quad \tau_3 \quad \text{[2,6]} \quad \tau_4 \quad \text{[2,58]} \quad \tau_4 \quad \text{[4,1]} \]

\textit{Figure 3.2 : Example of a constraint graph.}

events and five constraints. Note that, somewhat sloppy, for event $i$, we denote by $\tau_i$ both
its event time \( \tau_1 \) and its node identification. Also, note the direction of the arcs, e.g. the arc between \( \tau_1 \) and \( \tau_2 \) should be understood as

\[
2 \leq \tau_2 - \tau_1 + 60p_{21} \leq 4 .
\]  

(3.13)

Furthermore, the two relations between \( \tau_1 \) and \( \tau_4 \) express that event 1 happens between 2 and 4 minutes after event 4 or vice versa. The constraint graph will play an important role when searching for a timetable.

To comply with [126] we call the problem of searching for a timetable in a timetable structure the Periodic Event Scheduling Problem (PESP). In concise vector notation, PESP amounts to solving \( \tau \in \mathbb{R}^{N+1} \) and \( p \in \mathbb{Z}^M \) from the linear system

\[
l \leq B^T \tau + Tp \leq u ,
\]

(3.14)

where \( l \) and \( u \) represent the intervals on the arcs of \( \mathcal{G} \) and \( B \) is the vertex-arc incidence matrix of \( \mathcal{G} \), i.e. \( B \) is of size \( (N + 1) \times M \) and all its entries are zero except that for all \( m = ij \in \mathcal{A} \) we have \( B_{im} = 1 \) and \( B_{jm} = -1 \).

In the remainder we denote a timetable structure by \( (\mathcal{G}, l, u, T) \) and its set of arcs by \( \mathcal{A} \). Unless stated otherwise, we assume it is fixed throughout.

Given \( p \in \mathbb{Z}^M \) we denote with \( K(p) \) the set of timetables corresponding to \( p \), i.e.

\[
K(p) = \{ \tau \mid l - Tp \leq B^T \tau \leq u - Tp \} .
\]

(3.15)

If \( K(p) \neq \emptyset \), then we call \( p \) a feasible phase shift, otherwise it is called an infeasible phase shift.

As to the graph structure of \( \mathcal{G} \), the blocks of \( \mathcal{G} \) form timetable structures by themselves. Solving them separately and compiling the results (if any), then, solves the original one. Since, by definition, the blocks of \( \mathcal{G} \) are two-connected [18], all vertex degrees are at least two. Moreover, if a vertex has degree two, it can be deleted from the timetable structure by concatenating the two arcs incident to it. Hence, we may assume that \( \mathcal{G} \) is two-connected and that the minimum vertex degree outnumbers two (however, in Chapter 5 we will drop this assumption). Note that a simplification may initiate a cascade of reductions of the original problem.

### 3.2 Computational complexity

In this section we deal with some complexity issues concerning PESP. After the following lemma we state the main result of this section.

**Lemma 3.1** Let \( (\mathcal{G}, l, u, T) \) be a timetable structure and let \( \tau \) be a feasible timetable. Then, there is a unique feasible phase shift \( p \) for which \( \tau \in K(p) \).

**Proof** Suppose that for some \( ij \in \mathcal{A} \) we have

\[
l_{ij} \leq \tau_i - \tau_j + p_{ij}T \leq u_{ij}
\]

(3.16)
and
\[ l_{ij} \leq \tau_i - \tau_j + \tilde{p}_{ij} T \leq u_{ij} . \] (3.17)
Subtracting them yields
\[ l_{ij} - u_{ij} \leq (p_{ij} - \tilde{p}_{ij} T \leq u_{ij} - l_{ij} . \] (3.18)
Then, use that \( 0 < u_{ij} - l_{ij} < T \) and the integrality of \( p - \tilde{p} \) to conclude that
\[ p_{ij} - \tilde{p}_{ij} = 0 . \] (3.19)

\[ \square \]

**Proposition 3.1** PESP is NP-complete.

**Proof** With PESP-Int we denote PESP with the additional requirement that feasible timetables must be integral, i.e. all event times are required to be integer valued. Given timetable structure \((\mathcal{G}, l, u, T)\), the vertex-arc incidence matrix \(B\) of \(\mathcal{G}\) is totally unimodular and, therefore, any feasible timetable \(\tau\) can be converted into a feasible integral timetable in \(K(p)\) in polynomial time, where \(p\) is the unique phase shift (see Lemma 3.1) for which \(\tau \in K(p)\). Also, it is not difficult to see that PESP and PESP-Int are both in NP, so that PESP is NP-complete if PESP-Int is NP-complete.

We will show that the Graph Coloring Problem, which is known to be NP-complete [56], can be reduced to PESP-Int. The Graph Coloring Problem is the problem of, given a graph \((V, E)\) with vertex set \(V\) and edge set \(E\), and a number \(K \leq \#V\), to decide if there is a mapping \(f \in V \rightarrow \{1, \ldots, K\}\) such that \(f(i) \neq f(j)\) whenever \((i, j) \in E\). From \((V, E)\) and \(K\) we construct a timetable structure \((\mathcal{G}, l, u, T)\) such that \(\mathcal{G}\) equals \((V, E)\) extended with an arbitrary arc orientation, \(T = K\), and all arc-intervals read \([1, \ldots, K - 1]\), see

![Figure 3.3: Reduction from Graph Coloring to PESP-Int.](image)

Figure 3.3. Clearly, any feasible timetable assigns a different event time to two events that are adjacent in \(\mathcal{G}\) so that a timetable yields a proper coloring of \((V, E)\). Conversely, if \(C\)
is a coloring of $V$, then $\tau = C$ yields a feasible timetable. Also, the reduction is done in polynomial time, which implies the result. \hfill \square

The idea of reducing Graph Coloring to PESP is based on [34]. In [63, 126] a reduction from the Hamiltonian Circuit Problem [56] is suggested. PESP is even NP-complete in the strong sense [126].

There are, however, classes of timetable structures that are solvable in polynomial time. For one thing, PESP-Int in the proof of Proposition 3.1 is trivial for $T = 2$ because in this case all constraints become trivially satisfied for any timetable $\tau \in \mathbb{Z}^{N+1}$. More significantly, we have the following sequence of results. To start with, it can be shown that a part of the phase shift that corresponds to an arbitrary, though fixed, spanning tree $T \subset \mathcal{A}$ of $\mathcal{G}$ can be set to 0 beforehand. A cut of $\mathcal{G}$ is a vector $s \in \mathbb{R}^M$ such that $s = B^T q$ for some $q \in \mathbb{R}^{N+1}$, i.e. $s$ is in the linear space, called the cut space of $\mathcal{G}$, generated by the rows of $B$. We have,

**Lemma 3.2** Let $p$ and $\hat{p}$ be two phase shifts. Then, $p - \hat{p}$ is in the cut space of $\mathcal{G}$ if and only if

$$K(\hat{p}) = K(p) + Tq$$

(3.20) for some $q \in \mathbb{Z}^{N+1}$. Furthermore, if $\tau$ and $\hat{\tau}$ are two feasible timetables such that $\tau - \hat{\tau} = Tz$ for some $z \in \mathbb{Z}^{N+1}$, then their respective phase shifts differ a cut of $\mathcal{G}$.

**Proof** Suppose that $p - \hat{p}$ is in the cut space of $\mathcal{G}$. We can choose $q \in \mathbb{Z}^{N+1}$ such that $B^T q = p - \hat{p}$, because the rows of $B$ span the cut space of $\mathcal{G}$ and $B$ is totally unimodular. Then,

$$K(\hat{p}) = \{ \tau \in \mathbb{R}^{N+1} | l \leq B^T \tau + T\hat{p} \leq u \}$$

(3.21)

$$= \{ \tau \in \mathbb{R}^{N+1} | l \leq B^T \tau + T(p - \hat{p}) \leq u \}$$

(3.22)

$$= \{ \tau \in \mathbb{R}^{N+1} | l \leq B^T (\tau - Tq) + Tp \leq u \}$$

(3.23)

$$= K(p) + Tq.$$  

(3.24)

The reverse follows easily. As to the second part of the lemma, use, again, that the rows of $B$ span the cut space of $\mathcal{G}$, to obtain that

$$l \leq H^T \tau + Tp \leq u$$

(3.25)

and

$$l \leq B^T \hat{\tau} + T\hat{p} \leq u$$

(3.26)

imply that

$$- T < l - u \leq B^T (\tau - \hat{\tau}) + T(p - \hat{p}) \leq u - l < T,$$

(3.27)

so that

$$p - \hat{p} = B^T z$$

(3.28)
follows all numbers are integral.

Note that timetables that differ multiples of the period $T$ in their co-ordinates are the same from operational point of view. In the remainder of this chapter $T$ is a fixed spanning tree of $G$. Such tree generates $N$ so-called fundamental cuts of $G$. These are cuts with entries in $\{0, \pm 1\}$ that span the cut space of $G$, which has dimension $N$. Moreover, if $s$ is a fundamental cut, then $s_m = 1$ for some arc $m$ of $T$ and $c_k = 0$ for all other arcs of $T$. We write

$$Z_T^M = \{ z \in \mathbb{Z}^M \mid k \in T \Rightarrow z_k = 0 \} \quad (3.29)$$

and conclude, like in [126], that

**Proposition 3.2** It is sufficient to consider only phase shifts in the set $Z_T^M$.

**Proof** Let $m$ be an arc of $T$ and let $s^m$ be the $m$-th fundamental cut of $G$, i.e. $(s^m)_m = 1$ and $(c^m)_k = 0$ for all arcs in $T \setminus \{m\}$. Then, if $p$ is a feasible phase shift, then, following Lemma 3.2, $p - p_m s^m$ is also a feasible phase shift that is equivalent with $p$ in the sense that they generate equivalent sets of timetables. Furthermore, by subtracting $s^m$ from $p$ no entries corresponding to an arc of $T$ other than $m$ are affected and $(p - p_m s^m)_m = 0$. Then, apply the above to the other $N - 1$ fundamental cuts of $G$. \qed

Since $T$ has $N$ arcs, only $M - N$ integral variables remain in the problem formulation. Note that in stead of 0 we may assign any other integer to each of the entries of the phase shift corresponding to $T$.

**Proposition 3.3** If $M$ is fixed or if $M - N$ is fixed, then PESP is solvable in polynomial time.

**Proof** PESP can be stated as the mixed integer linear program

$$\begin{align*}
\max & \quad 0 \\
\text{s.t.} & \quad I \leq B^T \tau + TP \leq u \\
& \quad p \in \mathbb{Z}^M,
\end{align*} \quad (3.30)$$

where $I$ is the unit matrix of size $M$. Clearly, the rank of $I$ is $M$, so that for fixed $M$ PESP is solvable in polynomial time [80, 120]. Furthermore, by virtue of Proposition 3.2, $N$ entries of $p$ can be set to 0 without loss or creation of solutions. Thus the mixed integer linear program formulation can be reduced to

$$\begin{align*}
\max & \quad 0 \\
\text{s.t.} & \quad I \leq B^T \tau + T \tilde{p} \leq u \\
& \quad p \in \mathbb{Z}^M,
\end{align*} \quad (3.31)$$

where $\tilde{I}$ is the identity matrix with $N$ columns (corresponding to $T$) deleted. Since $\tilde{I}$ has rank $M - N$, it follows that also for fixed $M - N$ PESP is solvable in polynomial time. \qed

In the remainder of this chapter we assume that all entries of feasible phase shifts are zero.
3.3 Generating a timetable

Several methods to solve the PESP exist. In 1989 [126], for instance, a timetable is constructed by means of a backtrack procedure. Each level of the search tree is associated with a constraint of the timetable structure so that the constraints are satisfied sequentially. Satisfying an additional constraint is done by solving a Max Tension Problem [111]. This, in turn, is done by solving a Shortest Path Problem, since this is the dual problem of the Max Tension Problem. The number of nodes visited is tried to keep as low as possible by processing the constraints with small interval width first, thereby applying the principle that the probability of being forced to backtrack is kept low if the accumulation of information when going deeper into the tree grows as slow as possible.

It is acknowledged in [126] that the algorithm behaves not satisfactory whenever a lot of arcs have large interval width, something which can be troublesome in relation to railway timetable structures as usually a large proportion of constraints have an interval width of almost the period, see the example in Section 3.4. Infeasibility diagnosis of a PESP instance at hand is done by returning a circuit of $\mathcal{G}$ that is infeasible with respect to the choices made in the search tree at higher levels. Thus, such a circuit does not need to represent the real problem but only a local one. What's more, even if such a circuit is returned, there might still be a feasible solution at some other leaf of the tree.

In 1994 two other approaches were communicated [108], although numerical experiments were not presented. The first approach resembles the cutting plane approach that we present below. The main difference between these two approaches is that they derive cuts from group techniques, whereas we derive cuts from the structure of $\mathcal{G}$. In the second approach the structure of $\mathcal{G}$ is exploited to grow a search tree, each level of which corresponds to an event time. At a certain level, say corresponding to event time $\tau$, the entries of the phase shift corresponding to arcs of $\mathcal{G}$ that are adjacent to vertex $\tau$ are set by considering the same cuts that we will use below.

Another algorithm based on backtracking is presented in 1993 in [142]. This method searches for integer valued timetables and also produces a search tree where each level is associated with an event time. In each node of the tree the event time to be scheduled is assigned a value from the set of possible values given assignments on higher levels of the tree. Thereafter, still at the same level, the choice made for the event time is propagated through the constraints of the timetable structure, thereby reducing possible assignment values at deeper levels.

In [121, 122] timetable structures are solved directly through the application of a branch-and-bound algorithm. It is suggested to do some feasibility checking beforehand by checking the consistency of the constraints in small circuits in $\mathcal{G}$, see also Proposition 3.4.

Like mentioned in [126] PESP instances with a relatively large number of constraints with wide interval seem to be very hard. Indeed, computational experience with railway timetable structures confirm this observation. However, at present timetable structures of considerable size can be solved within reasonable time using the method presented in [142].

In the remainder of this section we state a full characterization of feasibility of phase shifts, thereby extending results in [34, 126], and use it to formulate a cut generation
approach to solve PESP. We stress that we aim at giving more insight in the structure of the problem rather than developing a faster algorithm. This insight is needed in Chapters 4 and 5.

For a fixed phase shift $p$ we write (3.14) as

$$l_p \leq B^T \tau \leq u_p ,$$

(3.32)

where

$$l_p = l - Tp$$

(3.33)

and

$$u_p = u - Tp .$$

(3.34)

Solving (3.32) for fixed $p$ is known as the Feasible Differential Problem (FDP) [111] and can be solved in $O(N^3)$ time with the FDP algorithm that can also be found in [111]. Furthermore, we call a vector in $c \in \{0, \pm 1\}^M$ an elementary circuit of $G$ if $Bc = 0$ and $c$ contains no subcircuits. By $C$ we denote the set of all such circuits. $C$ contains for any elementary circuit $c$ its negation $-c$ as well. Also, for an elementary circuit $c$ we write $c^+ = \max(c, 0)$ and $c^- = -\min(c, 0)$ component-wise. The set of fundamental circuits $F$ is a subset of $C$ and consists of the elementary circuits of $G$ that can be constructed by taking an arc that is not in spanning tree $T$ and extending it to an elementary circuit by selecting a number of arcs in $T$. This can only be done in one way, which fixes $F$. Like with $C$, if $f \in F$, then also $-f \in F$. The fundamental circuits can be found algebraically by taking the $M - N$ columns of matrix

$$\begin{pmatrix} -B_T^{-1}B_A \setminus T \\ I_{M-N} \end{pmatrix} ,$$

(3.35)

where $B_T$ and $B_A \setminus T$ partition $B$ such that the columns of $B_T$ are the columns of $B$ corresponding to $T$ and the columns of $B_A \setminus T$ correspond to the remaining columns. Hence, there are $2(M - N)$ fundamental circuits in $F$ that are pairwise related.

Our main result yields

**Proposition 3.4** Let $p$ be a phase shift. Then $p$ is feasible if and only if

$$c^T p \leq \left[ \frac{1}{T} u^T c^+ - \frac{1}{T} l^T c^- \right]$$

(3.36)

holds for all $c \in C$.

**Proof** If $p$ is feasible, (3.36) follows trivially for each elementary circuit $c$ by summing over the constraints that form $c$, taking the arc orientations into account. The converse is less obvious, but follows nevertheless easily from a theorem [111] about the feasibility of FDP instances, which says that (3.32) is feasible if and only if for all $c \in C$ we have

$$u_p^T c^+ - l_p^T c^- \geq 0 .$$

(3.37)
Then, if \( p \) is infeasible, there is an elementary circuit \( c \in C \) for which
\[
u_s^T c^+ - l_p^T c^- < 0,
\]
which implies that
\[
c^T p > \frac{1}{T} u^T c^+ - \frac{1}{T} l^T c^-,
\]
thereby yielding
\[
c^T p > \left[ \frac{1}{T} u^T c^+ - \frac{1}{T} l^T c^- \right].
\]
\( \square \)

Recall that all entries of feasible phase shifts are set to zero without loss of generality. Then, Proposition 3.4 implies

**Corollary 3.1** For the \( m \)-th fundamental circuit \( f \in \mathcal{F} \), corresponding to arc \( m \in \mathcal{A} \setminus \mathcal{T} \), we have
\[
p_m \leq \left[ \frac{1}{T} u^T f_m^+ - \frac{1}{T} l^T f_m^- \right].
\]

**Proof** From the definition of the \( m \)-th fundamental circuit, it follows that \( m \) is the only arc in \( \mathcal{A} \setminus \mathcal{T} \) for which the corresponding entry of the phase shift is nonzero. Then, apply Proposition 3.2 and Proposition 3.1.
\( \square \)

We define the set
\[
\mathcal{Q}_0 = \{ p \in \mathbb{Z}_T^M \mid f^T p \leq \left[ \frac{1}{T} u^T f^+ - \frac{1}{T} l^T f^- \right] \text{ for all } f \in \mathcal{F} \},
\]
which is a box in \( \{ x \in \mathbb{R}^M \mid k \in \mathcal{T} \Rightarrow x_k = 0 \} \) by virtue of Corollary 3.1. Clearly, if \( p \) is a feasible phase shift, then \( p \in \mathcal{Q}_0 \). This fact will be exploited below when developing our cut generation algorithm.

In [96] a more direct proof of Proposition 3.4 is given based on Farkas' lemma [92] rather than on the feasibility characterization of the FDP as in the above. In [34] a proposition is stated similar to Proposition 3.4 for a model that is equivalent to the timetable structure model, see Chapter 1.

It is interesting to note that Proposition 3.4 can also be derived by using Benders' reformulation of mixed integer linear programming [15], see also [92]. To this end, write a PESP instance down as the programming problem
\[
\begin{array}{ll}
\max & 0 \\
\text{s.t.} & B^T \tau + Tp \leq u \\
& -B^T \tau - Tp \leq -l \\
& p \in D, \tau \geq 0,
\end{array}
\]
where
\[
D = \{ z \in \mathbb{Z}_T^M \mid z \geq 0 \}.
\]
Note that \( p \geq 0 \) can always be enforced by a linear translation over the lower bounds of the box constraints on \( p \) defined by \( Q_0 \). Benders' reformulation, then, says that, if we abusively write \((a, b)\) to denote \((a^T, b^T)^T\), we can rewrite (3.43) as

\[
\begin{align*}
\max \quad & 0 \\
\text{s.t.} \quad & (u - Tp)^T r_1 - (l - Tp)^T r_2 \geq 0 \text{ for all extremal rays } (r_1, r_2) \text{ of } L, \\
& p \in D,
\end{align*}
\]

(3.45)

with

\[
L = \{(y_1, y_2) \in \mathbb{R}^{2M}_+ \mid B(y_1 - y_2) \geq 0\},
\]

(3.46)

which is equivalent to

\[
L = \{(y_1, y_2) \in \mathbb{R}^{2M}_+ \mid B(y_1 - y_2) = 0\}.
\]

(3.47)

In Chapter 5 it is shown that the set of extremal rays of \( L \) is the union of

\[
\{(e_i, e_i) \in \{0, \pm 1\}^{2M} \mid i = 1, \ldots, M\}
\]

(3.48)

and

\[
\{(c^+, c^-) \in \{0, \pm 1\}^{2M} \mid c \in C\}
\]

(3.49)

so that (3.45) is equivalent to

\[
\begin{align*}
\max \quad & 0 \\
\text{s.t.} \quad & (u - Tp)^T c^+ - (l - Tp)^T c^- \geq 0 \text{ for all } c \in C, \\
& p \in D,
\end{align*}
\]

(3.50)

because \( u - l > 0 \). Proposition 3.4 now follows from (3.50).

Proposition 3.4 provides us with a means to search for timetables using a cutting plane method, see the box in Figure 3.4. The basic idea is to look for a feasible phase shift by starting with some initial phase shift \( p \in Q_0 \) and iteratively cutting off parts of \( Q_0 \) so as to either converge to a feasible phase shift or to a set of hyperplanes that does not contain any phase shift (thereby providing a certificate of infeasibility of the problem instance).

Mathematically, we construct a finite row of sets \( \{Q_i\}_{i=0,\ldots,K} \) with \( \mathbb{Z}^M_+ \supset Q_0 \supset Q_1 \supset Q_2 \supset \ldots \supset Q_K \), where the inclusions are strict, and a finite row \( \{p^i\}_{i=0,\ldots,K} \) of phase shifts with \( p^i \in Q_i \) (if \( Q_i \neq \emptyset \)) for all \( i = 0, \ldots, K \), such that either \( Q_K = \emptyset \) or \( p^K \) is feasible. The sets \( Q_i \) are constructed iteratively as follows. In iteration \( i \) we start with \( Q_i \), which are the points of \( \mathbb{Z}^M_+ \) enclosed by an explicitly given set of linear hyperplanes in \( \mathbb{R}^M \), and search for a phase shift \( p^i \) in \( Q_i \). If \( Q_i \) turns out to be empty, the set of hyperplanes defining \( Q_i \) is a certificate of infeasibility of the timetable structure. If, on the other hand, a phase shift \( p^i \) is found, then we test \( p^i \) on feasibility by applying the FDP algorithm [111]. If it is feasible, the FDP algorithm returns a timetable and we are done. If \( p^i \) is not feasible, the FDP algorithm returns an elementary circuit of \( G \) for which (3.36) is violated. Then, use this inequality to cut off \( p^i \) from \( Q_i \) so as to construct \( Q_{i+1} \) and go over to the next iteration.
Cutting plane algorithm for PESP

**Input.** Timetable structure \( \mathcal{G} \).

**Initialization.** Set \( i = 0 \) and \( \mathcal{Q}_0 = \{ p \in \mathbb{Z}_+^M \mid f^T p \leq [\frac{1}{2} u^T f^+ - \frac{1}{2} l^T f^-] \text{ for all } f \in \mathcal{F} \} \).

**First part of iteration** \( i \). If \( \mathcal{Q}_i = \emptyset \), then \( \mathcal{G} \) is infeasible, output \( \mathcal{Q}_i \) and stop, otherwise take a \( p^i \in \mathcal{Q}_i \) and go to the second part of this iteration.

**Second part of iteration** \( i \). Decide if \( K(p^i) = \emptyset \) by applying the FDP algorithm. If not, then \( \mathcal{G} \) is solved, output the timetable returned by the FDP algorithm and stop, otherwise construct \( \mathcal{Q}_{i+1} \) by adding the cut that is generated by the FDP algorithm. Set \( i \leftarrow i + 1 \), and go to the next iteration.

*Figure 3.4 : Outline of the cutting plane algorithm to search for a timetable.*
Given $Q_i$ in the $i$-th iteration searching for a phase shift $p^i \in Q_i$, boils down to searching for an integral point in a polytope with an explicit description. This problem can be solved by choosing some linear objective $d \in \mathbb{R}^{M-N}$ on $p^i$ and solve the integer linear program

$$\begin{align*}
\max & \quad d^T p \\
\text{s.t.} & \quad p \in Q_i.
\end{align*}$$

This way different timetables can be found by choosing different objectives $d$. Note that searching for a phase shift in $Q_0$ is trivial and that selecting different phase shifts in $Q_0$ can also be a way to influence the course of the algorithm.

We have

**Proposition 3.5** The cutting plane algorithm terminates within $(N+1)^{M-N}$ iterations.

**Proof** Let $m \in A \setminus T$ and let $f^m$ be the fundamental circuit corresponding to arc $m$. Then,

$$p_m \leq \left| \frac{1}{T} u^T (f^m)^+ - \frac{1}{T} l^T (f^m)^- \right|$$

and

$$-p_m \leq \left| \frac{1}{T} u^T (-f^m)^+ - \frac{1}{T} l^T (-f^m)^- \right|.$$  

Furthermore,

$$\left| \frac{1}{T} u^T (f^m)^+ - \frac{1}{T} l^T (f^m)^- \right| + \left| \frac{1}{T} u^T (-f^m)^+ - \frac{1}{T} l^T (-f^m)^- \right|$$

$$\leq \frac{1}{T} u^T (f^m)^+ - \frac{1}{T} l^T (f^m)^- + \frac{1}{T} u^T (-f^m)^+ - \frac{1}{T} l^T (-f^m)^-$$

$$= \frac{1}{T} (u^T (f^m)^+ + u^T (f^m)^-) - \frac{1}{T} (l^T (f^m)^- + l^T (f^m)^+)$$

$$= \frac{1}{T} (u - l)^T |f^m|$$

$$\leq \frac{1}{T} (u - l)^T e$$

$$< N + 1,$$

because $0 < u - l < T$. With $|f^m|$ we mean taking absolute values component wise and not length. Hence, $Q_0$ has no more than $(N+1)^{M-N}$ feasible phase shifts. Also, in each iteration of the cutting plane algorithm at least one infeasible phase shift is cut off. □

It would be fair to say that the algorithm only works well for instances with $M - N$ up to about 30 [97]. For practical purposes this is not always enough which limits the practical use of the algorithm. More importantly, Proposition 3.4 shows that a PESP instance can be decomposed in a natural way in two separate problems. First search for a feasible phase shift and, thereafter, search for a timetable consistent with it. In Chapters 4 and 5 this decomposition is further exploited and interpreted in the context of randomly generating phase shifts and timetables.
3.4 An example

In this section we describe an example of a timetable structure that arose in a study of a major railway station in the Netherlands, called Arnhem CS, and demonstrate the use of the cutting plane algorithm. The proposed layout is shown in Figure 3.5. It has six platforms and three ingoing or outgoing directions. These are Nijmegen (Nm), Utrecht (Ut), and Zevenaar (Zv), respectively. There are 12 different trains visiting it, resulting in 24 identifiable events, namely the arrival and departure of each such train. The trains are named after their train type, origin and destination. For instance, AS-Ut/Nm is an all-stations train coming from Utrecht and going to Nijmegen. The other two train types are intercity trains (IC) that stop only at major stations and interregional trains (IR), which are in between AS and IC. The classification AS, IR and IC is also a sort of ranking of the trains with respect to the average speed. We say, for instance, that an IC train is of higher rank than an IR train. Superscripts attached to the event names distinguish between arrival (a) and departure (d) events. All trains visit the station twice an hour in a strict half an hour pattern so that the period \( T \) can be set to 30 minutes.

The timetable structure consists of 54 constraints, each of which is stated on a separate line in Tables 3.1 through 3.4. Such line should be read as ‘train 2 arrives at, or departs from, Arnhem CS between \( l \) and \( u \) minutes after train 1 arrives at, or departs from, Arnhem CS’. The first 17 constraints are operating policy constraints (Table 3.1 and 3.2) of which the five in Table 3.1 state connections between trains to allow for passenger transfers. The graphical representation of these five constraints is shown in Figure 3.6. The operating policy constraints in Table 3.2 denote the halting times of the trains. Normally, the halting time is between 2 and 5 minutes but trains that turn at Arnhem CS, i.e. trains having the same origin and destination, need 10 to 15 minutes to start a new journey because they need to be cleaned and inspected first.

Constraints 18 through 54 constitute infrastructure constraints (Tables 3.3 and 3.4). The constraints in Table 3.3 relate the arrival times of trains that have a common origin.
<table>
<thead>
<tr>
<th>Nr</th>
<th>Train 1</th>
<th>Precedes</th>
<th>Train 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IC-Ut/Zv\textsuperscript{d}</td>
<td>[3, 4]</td>
<td>IR-Zv/Nm\textsuperscript{d}</td>
</tr>
<tr>
<td>2</td>
<td>IC-Ut/Zv\textsuperscript{d}</td>
<td>[5, 6]</td>
<td>AS-Ut/Zv\textsuperscript{d}</td>
</tr>
<tr>
<td>3</td>
<td>IC-Ut/Zv\textsuperscript{d}</td>
<td>[2, 3]</td>
<td>AS-Zv/Zv\textsuperscript{d}</td>
</tr>
<tr>
<td>4</td>
<td>IR-Nm/Zv\textsuperscript{d}</td>
<td>[2, 4]</td>
<td>IC-Zv/Ut\textsuperscript{d}</td>
</tr>
<tr>
<td>5</td>
<td>AS-Zv/Ut\textsuperscript{d}</td>
<td>[3, 4]</td>
<td>IR-Nm/Ut\textsuperscript{d}</td>
</tr>
</tbody>
</table>

Table 3.1: The operating policy constraints for train connections.

<table>
<thead>
<tr>
<th>Nr</th>
<th>Train 1</th>
<th>Precedes</th>
<th>Train 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>AS-Nm/Nm\textsuperscript{a}</td>
<td>[10, 15]</td>
<td>AS-Nm/Nm\textsuperscript{d}</td>
</tr>
<tr>
<td>7</td>
<td>AS-Nm/Zv\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>AS-Nm/Zv\textsuperscript{d}</td>
</tr>
<tr>
<td>8</td>
<td>AS-Ut/Zv\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>AS-Ut/Zv\textsuperscript{d}</td>
</tr>
<tr>
<td>9</td>
<td>AS-Zv/Ut\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>AS-Zv/Ut\textsuperscript{d}</td>
</tr>
<tr>
<td>10</td>
<td>AS-Zv/Zv\textsuperscript{a}</td>
<td>[10, 15]</td>
<td>AS-Zv/Zv\textsuperscript{d}</td>
</tr>
<tr>
<td>11</td>
<td>IC-Ut/Zv\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>IC-Ut/Zv\textsuperscript{d}</td>
</tr>
<tr>
<td>12</td>
<td>IC-Zv/Ut\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>IC-Zv/Ut\textsuperscript{d}</td>
</tr>
<tr>
<td>13</td>
<td>IR-Nm/Ut\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>IR-Nm/Ut\textsuperscript{d}</td>
</tr>
<tr>
<td>14</td>
<td>IR-Nm/Zv\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>IR-Nm/Zv\textsuperscript{d}</td>
</tr>
<tr>
<td>15</td>
<td>IR-Ut/Nm\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>IR-Ut/Nm\textsuperscript{d}</td>
</tr>
<tr>
<td>16</td>
<td>IR-Ut/Ut\textsuperscript{a}</td>
<td>[10, 15]</td>
<td>IR-Ut/Ut\textsuperscript{d}</td>
</tr>
<tr>
<td>17</td>
<td>IR-Zv/Nm\textsuperscript{a}</td>
<td>[2, 5]</td>
<td>IR-Zv/Nm\textsuperscript{d}</td>
</tr>
</tbody>
</table>

Table 3.2: The operating policy constraints for halting times.
<table>
<thead>
<tr>
<th>Nr</th>
<th>Train 1</th>
<th>Precedes</th>
<th>Train 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>AS-Ut/Zv²</td>
<td>[2, 6]</td>
<td>IC-Ut/Zv²</td>
</tr>
<tr>
<td>19</td>
<td>AS-Ut/Zv²</td>
<td>[2, 28]</td>
<td>IR-Ut/Nm²</td>
</tr>
<tr>
<td>20</td>
<td>AS-Ut/Zv²</td>
<td>[2, 28]</td>
<td>IR-Ut/Ut²</td>
</tr>
<tr>
<td>21</td>
<td>IC-Ut/Zv²</td>
<td>[2, 28]</td>
<td>IR-Ut/Nm²</td>
</tr>
<tr>
<td>22</td>
<td>IC-Ut/Zv²</td>
<td>[2, 28]</td>
<td>IR-Ut/Ut²</td>
</tr>
<tr>
<td>23</td>
<td>IR-Ut/Nm²</td>
<td>[2, 28]</td>
<td>IR-Ut/Ut²</td>
</tr>
<tr>
<td>24</td>
<td>IR-Nm/Ut²</td>
<td>[2, 28]</td>
<td>AS-Nm/Nm²</td>
</tr>
<tr>
<td>25</td>
<td>IR-Nm/Ut²</td>
<td>[2, 28]</td>
<td>IR-Nm/Zv²</td>
</tr>
<tr>
<td>26</td>
<td>IR-Nm/Ut²</td>
<td>[2, 28]</td>
<td>AS-Nm/Zv²</td>
</tr>
<tr>
<td>27</td>
<td>AS-Nm/Nm²</td>
<td>[2, 28]</td>
<td>IR-Nm/Zv²</td>
</tr>
<tr>
<td>28</td>
<td>AS-Nm/Nm²</td>
<td>[2, 28]</td>
<td>AS-Nm/Zv²</td>
</tr>
<tr>
<td>29</td>
<td>IR-Nm/Zv²</td>
<td>[24, 28]</td>
<td>AS-Nm/Zv²</td>
</tr>
<tr>
<td>30</td>
<td>AS-Zv/Ut²</td>
<td>[2, 28]</td>
<td>AS-Zv/Zv²</td>
</tr>
<tr>
<td>31</td>
<td>AS-Zv/Ut²</td>
<td>[2, 6]</td>
<td>IC-Zv/Ut²</td>
</tr>
<tr>
<td>32</td>
<td>AS-Zv/Ut²</td>
<td>[2, 28]</td>
<td>IR-Zv/Nm²</td>
</tr>
<tr>
<td>33</td>
<td>AS-Zv/Zv²</td>
<td>[2, 28]</td>
<td>IC-Zv/Ut²</td>
</tr>
<tr>
<td>34</td>
<td>AS-Zv/Zv²</td>
<td>[2, 28]</td>
<td>IR-Zv/Nm²</td>
</tr>
<tr>
<td>35</td>
<td>IC-Zv/Ut²</td>
<td>[2, 28]</td>
<td>IR-Zv/Nm²</td>
</tr>
</tbody>
</table>

*Table 3.3: The infrastructure constraints for train arrivals.*
<table>
<thead>
<tr>
<th>Nr</th>
<th>Train 1</th>
<th>Precedes</th>
<th>Train 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>AS-Ut/Zvd</td>
<td>[3, 29]</td>
<td>IC-Ut/Zvd</td>
</tr>
<tr>
<td>37</td>
<td>AS-Ut/Zvd</td>
<td>[3, 29]</td>
<td>IR-Nm/Zvd</td>
</tr>
<tr>
<td>38</td>
<td>AS-Ut/Zvd</td>
<td>[2, 28]</td>
<td>AS-Nm/Zvd</td>
</tr>
<tr>
<td>40</td>
<td>IC-Ut/Zvd</td>
<td>[1, 27]</td>
<td>IR-Nm/Zvd</td>
</tr>
<tr>
<td>41</td>
<td>IC-Ut/Zvd</td>
<td>[1, 27]</td>
<td>AS-Nm/Zvd</td>
</tr>
<tr>
<td>42</td>
<td>IC-Ut/Zvd</td>
<td>[1, 27]</td>
<td>AS-Zv/Zvd</td>
</tr>
<tr>
<td>43</td>
<td>IR-Nm/Zvd</td>
<td>[1, 27]</td>
<td>AS-Nm/Zvd</td>
</tr>
<tr>
<td>44</td>
<td>IR-Nm/Zvd</td>
<td>[1, 27]</td>
<td>AS-Zv/Zvd</td>
</tr>
<tr>
<td>45</td>
<td>AS-Nm/Zvd</td>
<td>[2, 28]</td>
<td>AS-Zv/Zvd</td>
</tr>
<tr>
<td>46</td>
<td>IR-Ut/Nmd</td>
<td>[1, 27]</td>
<td>AS-Nm/Nmd</td>
</tr>
<tr>
<td>47</td>
<td>IR-Ut/Nmd</td>
<td>[2, 28]</td>
<td>IR-Zv/Nmd</td>
</tr>
<tr>
<td>48</td>
<td>AS-Nm/Nmd</td>
<td>[3, 29]</td>
<td>IR-Zv/Nmd</td>
</tr>
<tr>
<td>49</td>
<td>IR-Ut/Utd</td>
<td>[2, 28]</td>
<td>IR-Nm/Utd</td>
</tr>
<tr>
<td>50</td>
<td>IR-Ut/Utd</td>
<td>[1, 27]</td>
<td>AS-Zv/Utd</td>
</tr>
<tr>
<td>51</td>
<td>IR-Ut/Utd</td>
<td>[3, 29]</td>
<td>IC-Zv/Utd</td>
</tr>
<tr>
<td>52</td>
<td>IR-Nm/Utd</td>
<td>[1, 27]</td>
<td>AS-Zv/Utd</td>
</tr>
<tr>
<td>53</td>
<td>IR-Nm/Utd</td>
<td>[3, 29]</td>
<td>IC-Zv/Utd</td>
</tr>
<tr>
<td>54</td>
<td>AS-Zv/Utd</td>
<td>[3, 29]</td>
<td>IC-Zv/Utd</td>
</tr>
</tbody>
</table>

*Table 3.4: The infrastructure constraints for train departures.*
Most of these constraints state that trains coming from the same direction must arrive some time apart from each other to respect headway times on the supply tracks. Additionally, some of these constraints state that a lower rank train with origin $x$ and destination $y$ arrives just before a higher rank train that has also origin $x$ and destination $y$. This way we model that during normal railway operation higher (faster) rank trains tend to catch up on lower (slower) rank trains if they use the same route.

The infrastructure constraints in Table 3.4 state that two trains going out in the same direction must exit with a certain time space between them. For instance, constraint 36 states that the IC-Ut/Zv must either depart at least one minute earlier than the AS-Ut/Zv or depart at least three minutes later than the AS-Ut/Zv. This deviation from the symmetrical situation, as in constraint 38 (two minutes separation on either side), stems from the difference in ranking of the trains.

Note that the timetable structure as specified contains some redundancy. For example, constraints 2 and 3 make constraint 39 superfluous. Also, constraint 2 implies constraint 36, etc.

We applied the cutting plane algorithm to the given timetable structure. Within a few milliseconds it was found that the timetable structure is infeasible and it was reported that one reason (there may be others as well) for this is the subsystem of constraints \{2, 8, 11, 18\}. These four constraints form a circuit $c$ and are shown in Figure 3.7. If we
walk over this graph clock-wise we calculate

$$c^T p \leq \left[ \frac{1}{T} u^T c^+ + \frac{1}{T} l^T c^- \right] = \left[ \frac{6 + 5 + 6 - 2}{30} \right] = 0$$

(3.60)

and if we do the same anti-clock-wise we obtain

$$(-c)^T p \leq \left[ \frac{1}{T} u^T (-c)^+ + \frac{1}{T} l^T (-c)^- \right] = \left[ -\frac{2 + 2 + 5 - 5}{30} \right] = -1 .$$

(3.61)

Combining (3.60) and (3.61) yields

$$1 \leq c^T p \leq 0 .$$

(3.62)

Clearly, no phase shift can ever satisfy (3.62), which proves that the timetable structure is indeed infeasible. Actually, what happens is that the AS-Ut/Zv comes in at least 2 minutes earlier than the IC-Ut/Zv (constraint 18) and leaves the station no sooner than 5 minutes after the IC-Ut/Zv leaves (constraint 2). Therefore, the AS-Ut/Zv must wait at the station at least 7 minutes, which is more than its largest possible halting time (constraint 8). So in order to make the timetable structure feasible we could specify a longer halting time for the AS-Ut/Zv, which is what we did. In fact, we changed the interval of constraint 8 into [6, 9]. This makes the timetable structure feasible and we generated four timetables. Different timetables can be obtained by searching for $p_i \in Q_i$ in different ways in the first part of an iteration, see Section 3.3. In Table 3.5 the timetables are shown. The runtimes as shown in this table are obtained using a Pascal implementation on a PC-486. Furthermore, the number of cutting planes that were generated during the course of execution are reported.
<table>
<thead>
<tr>
<th>Train</th>
<th>Timetable 1</th>
<th>Timetable 2</th>
<th>Timetable 3</th>
<th>Timetable 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS-Nm/Nm</td>
<td>28</td>
<td>43</td>
<td>29</td>
<td>39</td>
</tr>
<tr>
<td>AS-Nm/Zv</td>
<td>2</td>
<td>4</td>
<td>27</td>
<td>32</td>
</tr>
<tr>
<td>AS-Zv/Zv</td>
<td>3</td>
<td>12</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>AS-Zv/Ut</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>AS-Zv/Zv</td>
<td>28</td>
<td>40</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>IC-Ut/Zv</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>IC-Zv/Ut</td>
<td>6</td>
<td>11</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>IR-Nm/Ut</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>IR-Nm/Zv</td>
<td>4</td>
<td>8</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>IR-Ut/Nm</td>
<td>10</td>
<td>12</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>IR-Ut/Ut</td>
<td>8</td>
<td>23</td>
<td>21</td>
<td>31</td>
</tr>
<tr>
<td>IR-Zv/Nm</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

| Runtime | 225 | 88  | 82  | 110 |
| # Iterations | 49  | 27  | 28  | 32  |

Table 3.5: Timetables generated with the cutting plane algorithm. Runtimes are in milliseconds.
Chapter 4

Random Generation of Classes of Railway Timetables

Since railway infrastructure is expensive and building it can take up to 15 years, infrastructure must be designed in such a way that it is likely that the resulting railway network can cope with future traffic scenarios. However, there is a lot of uncertainty about what these scenarios look like. To handle the uncertainties, scenarios are generated and studied in order to derive specifications of railway infrastructure and/or to analyze designs, see Chapter 1.

Roughly speaking, the last step of the timetabling process comprises the construction of a timetable from a so-called timetable structure. Mathematically, a timetable structure is a set of constraints modeling various requirements a timetable must meet, e.g. headway times and connections between trains, see Chapter 3.

Since the uncertainty about the future timetables grows with every step in the process of preparing timetables, the need to account for this grows with it. Hence, the problem of randomly generating timetables from a given timetable structure. In this chapter\(^1\) we focus on the random generation of classes of timetables from a given timetable structure in compliance with the so-called robustness distribution. This is a probability distribution that only assigns probability mass to those classes of timetables that contain timetables that are likely to be robust compared to timetables in other classes. From these classes, then, timetables can be selected for the purpose of uncertainty and sensitivity analysis. How to perform this selection is addressed in Chapter 5.

Given a timetable structure, the set of feasible timetables can be represented as a finite set of disjoint polytopes, each of which is parameterized by a so-called phase shift. In turn, a phase shift is an integral point in the so-called phase shift polytope, which has an implicit description by way of an explicit separation algorithm, so that effectively in this chapter we address the problem of generating integral points in the phase polytope according to the robustness distribution.

The general problem of randomly generating integral points in a polytope is very hard and, unfortunately, this is no different for our specific problem. In particular, deciding about the existence of a feasible phase shift is already NP-complete. Therefore, we turn to a heuristic approach.

The heuristic proceeds in two steps. First a continuous point \( x \) in the phase shift polytope is sampled from the so-called volume distribution on the phase shift polytope. This probability distribution is related to the robustness distribution and sampling from it is easily done using a transformation technique. In the second step \( x \) is rounded to the nearest feasible integral phase shift in compliance with some specific norm.

The remainder of this chapter is organized as follows. We start in the next section with a brief discussion of timetable structures, make some definitions, and state a few preliminary results. In Section 4.2 we introduce the robustness distribution. Thereafter, in Section 4.3, we explain the heuristic. The approach is evaluated in Section 4.4 on the basis of two artificial examples and a real-life example. The theoretical aspects of the computational complexity of the problem of generating phase shifts is dealt with in Section 4.5. Finally, in Section 4.6, we make some remarks.

### 4.1 The timetable structure model

In this chapter we adopt the timetable structure model described in Chapter 3. For sake of convenience we briefly recapitulate this model below. In Section 4.1.1 we show how the solutions, i.e. timetables, to a timetable structure can be classified. In particular, in that section we introduce a special polytope that will play a central role in this chapter.

A timetable is a mapping of a set of arrival and/or departure events, indexed consecutively from 0 to \( N \), to \( \mathbb{R} \). It can be calculated from a so-called timetable structure, which is a set of \( M \) constraints, called interval constraints, that are of the form

\[
l_{ij} \leq \tau_i - \tau_j + T p_{ij} \leq u_{ij},
\]

where \( T \), \( l_{ij} \) and \( u_{ij} \) are integer numbers such that \( 0 < u_{ij} - l_{ij} < T \). The variables \( \tau_i, \tau_j \in \mathbb{R} \) are the event times and the variable \( p_{ij} \in \mathbb{Z} \) states that \( \tau_i - \tau_j \) must be in one of the intervals \( \ldots, [l_{ij} - T, u_{ij} - T], [l_{ij}, u_{ij}], [l_{ij} + T, u_{ij} + T], \ldots \). Parameter \( T \) denotes the period of any timetable and is often set to 60 (minutes) to express the desire to construct hourly patterns. The interval constraints are used to model driving times, headway times, connections between trains, and so on.

More abstractly, a timetable structure is a mixed integer system of linear inequalities

\[
l \leq B^T \tau + T p \leq u,
\]

where \( B \in \{0, \pm 1\}^{(N+1) \times M} \) is the vertex arc incidence matrix of a directed graph \( \mathcal{G} \), and \( l \) and \( u \) are integral vectors satisfying \( 0 < u - l < T \). Here \( M \) is the number of arcs of \( \mathcal{G} \) or, equivalently, the number of interval constraints. Graph \( \mathcal{G} \) is such that its vertices correspond to the events (and hence the event times) and such that its arcs correspond to the constraints. Each row of (4.2) corresponds to a constraint like (4.1). Solving system (4.2) is NP-complete in the strong sense, see Chapter 3.
4.1.1 Partitioning of the set of timetables

With \( \Lambda(\mathbb{Z}) \) we denote the set of vectors \( p \in \mathbb{Z}^M \) for which (4.2) has a solution \( \tau \). Throughout it is assumed that \( \Lambda(\mathbb{Z}) \) is not empty. Also, we define

\[
K(p) = \{ \tau \in \mathbb{R}^{N+1} | l \leq B^T \tau + Tp \leq u \},
\]

(4.3)

being the set of timetables \( \tau \) consistent with \( p \). We call \( p \in \Lambda(\mathbb{Z}) \) a phase shift because of the role of \( p \) in (4.2) and we refer to \( K(p) \) as the timetable class corresponding to \( p \).

The representation of (4.1) by (4.2) allows for reasoning about the system using concepts from graph theory. We encode circuits of \( G \) with their incidence vectors denoting arc orientations with 1 if the orientation is consistent with some canonical direction and \(-1\) otherwise. Elementary circuits are circuits that are not decomposable into circuits with fewer arcs.

Let \( C = [C_i] \in \{0, \pm 1\}^{\kappa \times M} \) be a matrix, the rows of which are the elementary circuits of \( G \). We distinguish between the circuits \( c \) and \(-c\) so that \( \kappa \) is twice the number of elementary circuits of the underlying undirected graph of \( G \). Also, define the operator \( U \) on a circuit \( c \) as

\[
U(c) = \frac{1}{T}(u^Tc^+ - l^Tc^-),
\]

(4.4)

where \( c^+ = \max(c, 0) \) and \( c^- = -\min(c, 0) \). Note that applying the operator \( U \) component-wise to \( C \) makes sense if we view \( C \) as a column vector of circuits. Then, the fundamental result for the timetable structure reads

\[
\Lambda(\mathbb{Z}) = \{ p \in \mathbb{Z}^M | C^Tp \leq U(C) \},
\]

(4.5)

the proof of which is given in Chapter 3 under Proposition 3.4. It reveals that feasibility of (4.2) is completely characterized by a linear system of inequalities in \( p \) alone and that a timetable can be constructed by first finding a phase shift \( p \) and then solving (4.2) for \( \tau \), given \( p \). Solving (4.2) for fixed \( p \) is easy (see [111], where an \( O(N^3) \) algorithm is described). Hence, finding a phase shift or establishing that \( \Lambda(\mathbb{Z}) \) is empty constitutes the hard part of solving (4.2).

Since \( 0 < u - l < T \), the timetable classes are disjoint, see Chapter 3. Therefore, it is natural to use the two phase approach to search for a timetable also for the random generation of timetables. That is, we first randomly generate a feasible phase shift \( p \) in \( \Lambda(\mathbb{Z}) \) and then generate a timetable from \( K(p) \) uniformly at random for the purpose of sensitivity analysis as explained in Chapter 1. Sampling from \( K(p) \) is addressed in Chapter 5. In this chapter we address the problem of generating feasible phase shifts at random.

Adding a multiple of all-one vector \( e \) to a timetable does not change its characteristics and if \( \tau \) satisfies (4.2), then so does \( \tau + \alpha e \) for all \( \alpha \in \mathbb{R} \). Thus, we may restrict ourselves to timetables with \( \tau_0 = 0 \) and re-define the timetable class corresponding to \( p \) by

\[
K(p) = \{ \tau \in \mathbb{R}^N | l \leq \overline{B}^T \tau + Tp \leq u \},
\]

(4.6)
where $\tilde{B}$ is the matrix $B$ from which the first row (corresponding to vertex 0) is deleted.

There is also redundancy in $\Lambda(\mathbb{Z})$. That is, if $p \in \Lambda(\mathbb{Z})$, then so are all vectors $p + s$, where $s \in \{0, \pm 1\}^M$ is an integral vector in the cut space of $G$. Moreover, $K(p + s)$ is a linear translation of $K(p)$ over $Tq$ for some $q \in \mathbb{Z}^N$. Hence, it suffices to consider only phase shifts $p$ that have some fixed value for the entries corresponding to some spanning tree $T$ of $G$, see Chapter 3. In this chapter $T$ is arbitrary, yet fixed, and without loss of generality we assume that $T$ consists of the last $N$ arcs of $G$ (numbered $M - N + 1, \ldots, M$). Moreover, we choose the corresponding entries of $p$ to be 0, i.e. without loss of generality we can use

$$\tilde{\Lambda}(\mathbb{Z}) = \{ p \in \mathbb{Z}^{M-N} \mid \tilde{C}p \leq U(C) \} \quad (4.7)$$

instead of $\Lambda(\mathbb{Z})$, where $\tilde{C}$ is the matrix $C$ whose last $N$ columns have been deleted. If we denote the unit matrix of size $n$ by $I_n$ and the $n \times m$ matrix with only zero entries by $O_{n \times m}$, then, by defining

$$E = \begin{pmatrix} I_{M-N} & 0_{N \times (M-N)} \end{pmatrix}, \quad (4.8)$$

defining $E$ appends $N$ zeroes to the end of a vector of length $(M - N)$, we can write

$$\Lambda(\mathbb{Z}) = \bigcup_{p \in \tilde{\Lambda}(\mathbb{Z})} \{ Ep + \tilde{B}^Tq \mid q \in \mathbb{Z}^N \} \quad (4.9)$$

to indicate the partitioning of $\Lambda(\mathbb{Z})$ in equivalence classes with set of representatives $\tilde{\Lambda}(\mathbb{Z})$.

Note that the rows of $\tilde{B}$ span the cut space of $G$ and that, using $E$, we can also write

$$\tilde{\Lambda}(\mathbb{Z}) = \{ p \in \mathbb{Z}^{M-N} \mid C Ep \leq U(C) \}. \quad (4.10)$$

Besides $\Lambda(\mathbb{Z})$ and $\tilde{\Lambda}(\mathbb{Z})$ we also define two other closely related sets, namely

$$\tilde{\Lambda}(\mathbb{R}) = \{ x \in \mathbb{R}^{M-N} \mid \tilde{C}x \leq U(C) \} \quad (4.11)$$

and

$$\Lambda(\mathbb{R}) = \{ x \in \mathbb{R}^{M} \mid Cx \leq U(C) \}. \quad (4.12)$$

Clearly,

$$\tilde{\Lambda}(\mathbb{Z}) \subseteq \tilde{\Lambda}(\mathbb{R}) \quad (4.13)$$

and

$$\Lambda(\mathbb{Z}) \subseteq \Lambda(\mathbb{R}). \quad (4.14)$$

Like with $\tilde{\Lambda}(\mathbb{Z})$ we can write

$$\tilde{\Lambda}(\mathbb{R}) = \{ x \in \mathbb{R}^{M-N} \mid CEx \leq U(C) \} \quad (4.15)$$

and also

$$\Lambda(\mathbb{R}) = \bigcup_{x \in \tilde{\Lambda}(\mathbb{R})} \{ Ex + \tilde{B}^Ty \mid y \in \mathbb{R}^N \}. \quad (4.16)$$
to denote the relation between \( \Lambda(\mathbb{R}) \) and \( \hat{\Lambda}(\mathbb{R}) \).

It will be convenient to extend the domain of \( K \) to \( \hat{\Lambda}(\mathbb{Z}) \), so that

\[
K(p) = \{ \tau \in \mathbb{R}^N \mid l \leq \hat{B}^T \tau + TE p \leq u \} \tag{4.17}
\]

for all \( p \in \hat{\Lambda}(\mathbb{Z}) \). Likewise, we extend its definition to \( \Lambda(\mathbb{R}) \) and \( \hat{\Lambda}(\mathbb{R}) \). That is,

\[
K(x) = \{ \tau \in \mathbb{R}^N \mid l \leq \hat{B}^T \tau + Tx \leq u \} \tag{4.18}
\]

for all \( x \in \Lambda(\mathbb{R}) \) and

\[
K(x) = \{ \tau \in \mathbb{R}^N \mid l \leq \hat{B}^T \tau + TEx \leq u \} \tag{4.19}
\]

for all \( x \in \hat{\Lambda}(\mathbb{R}) \). In the remainder of this chapter we will sometimes abuse the operator \( E \) to allow ourselves to identify \( \Lambda(\mathbb{R}) \) and \( \hat{\Lambda}(\mathbb{R}) \) and also \( \Lambda(\mathbb{Z}) \) and \( \hat{\Lambda}(\mathbb{Z}) \).

As to the spanning tree \( T \) we have chosen, let \( S \in \{0, \pm 1\}^{N \times M} \) be the matrix whose rows encode the \( N \) corresponding fundamental cuts. Thus, the rows of \( S \) span the cut space of \( G \) (as do the rows of \( \hat{B} \)). Recalling that the last \( N \) arcs constitute the spanning tree \( T \), we see that \( S \) can be written as

\[
S = (\hat{S} \quad I_N) \tag{4.20}
\]

for some unique matrix \( \hat{S} \in \{0, \pm 1\}^{N \times (M-N)} \). Furthermore, if \( p \in \Lambda(\mathbb{Z}) \), then its unique representative in \( \hat{\Lambda}(\mathbb{Z}) \) can concisely be written as \( Zp \), where

\[
Z = (I_{M-N} \quad -\hat{S}^T) \tag{4.21}
\]

Finally, let \( F \in \{0, \pm 1\}^{(M-N) \times M} \) denote the matrix whose rows are the fundamental circuits of \( G \). In fact, in a similar way as with \( S \), \( F \) may be written as

\[
F = (I_{M-N} \quad \hat{F}) \tag{4.22}
\]

for some unique matrix \( \hat{F} \in \{0, \pm 1\}^{(M-N) \times N} \) and, since \( SF^T = 0 \), we have \( \hat{F} = -\hat{S}^T \), so that \( F = Z \). Therefore, we immediately revoke the introduction of the symbol \( Z \) and use \( F \) exclusively to denote the projection onto \( \hat{\Lambda}(\mathbb{Z}) \) or \( \hat{\Lambda}(\mathbb{R}) \).

From the definitions it follows that

\[
E\hat{\Lambda}(\mathbb{Z}) \subseteq \Lambda(\mathbb{Z}) \tag{4.23}
\]

and

\[
\hat{\Lambda}(\mathbb{Z}) = F\Lambda(\mathbb{Z}) \tag{4.24}
\]

where \( AX = \{ Ax \mid x \in X \} \) for matrix \( A \in \mathbb{R}^{m \times n} \) and set \( X \subseteq \mathbb{R}^n \). Similarly, note that

\[
E\hat{\Lambda}(\mathbb{R}) \subseteq \Lambda(\mathbb{R}) \tag{4.25}
\]

and

\[
\hat{\Lambda}(\mathbb{R}) = F\Lambda(\mathbb{R}) \tag{4.26}
\]
Also, 

\[ C = \tilde{C} F. \] (4.27)

By considering the \( M - N \) fundamental circuits of \( \mathcal{G} \) it is not difficult to see that \( \tilde{\Lambda}(\mathbb{Z}) \) is finite and that \( \tilde{\Lambda}(\mathbb{R}) \) is bounded. More precise, we have

\[
\tilde{\Lambda}(\mathbb{Z}) \subseteq \{ p \in \mathbb{Z}^{M-N} \mid FEp \leq U(F') \text{ and } (-F)Ep \leq U(-F) \} 
= \{ p \in \mathbb{Z}^{M-N} \mid -U(-F) \leq FEp \leq U(F) \} 
= \{ p \in \mathbb{Z}^{M-N} \mid -U(-F) \leq p \leq U(F) \},
\] (4.28) (4.29) (4.30)

because \( FE = I_{M-N} \). Similarly, we derive

\[
\tilde{\Lambda}(\mathbb{R}) \subseteq \{ x \in \mathbb{R}^{M-N} \mid -U(-F) \leq x \leq U(F) \}.
\] (4.31)

See also Chapter 3. As a result \( \tilde{\Lambda}(\mathbb{R}) \) is a polytope in \( \mathbb{R}^{M-N} \) and we refer to it as the phase shift polytope.

Recall that for vectors \( x \) and \( y \) we sometimes write \((x, y)\) to denote \((x^T, y^T)^T\). Furthermore, for sake of demonstration we sometimes give more than one proof for the same lemma, proposition, or corollary.

### 4.2 Generating phase shifts at random

If for a phase shift \( p \) its timetable class \( K(p) \) is of full dimension, then a timetable somewhere in the 'center' of \( K(p) \) lies away from the boundary of \( K(p) \) and is therefore robust compared to timetables close to the boundary. As timetable planners are always in search for robust timetables, it is thus natural to generate phase shifts corresponding to timetable classes with the highest appearing dimension.

In Section 4.2.5 we define the robustness distribution on \( \tilde{\Lambda}(\mathbb{Z}) \). It assigns only probability mass to the phase shifts of which the underlying sets of timetables have the highest geometric dimension among all such sets. Moreover, if the maximal dimension equals \( N \), then the robustness distribution even assigns probability mass proportional to the \( N \)-dimensional geometric volumes of the sets of timetables.

Before we define the robustness distribution we first derive some properties of the phase shift polytope in the next section and introduce a distance measure on it in Section 4.2.2. Thereafter, we define the volume function on \( \tilde{\Lambda}(\mathbb{R}) \) in Section 4.2.3, and use this function to introduce a probability density on \( \tilde{\Lambda}(\mathbb{R}) \) in Section 4.2.4. This density, then, is used to define the robustness distribution on \( \tilde{\Lambda}(\mathbb{Z}) \) in Section 4.2.5. Throughout, it will be shown that the choice of spanning tree \( T \) has no influence on the results that pass by.

#### 4.2.1 Properties of the phase shift polytope

We introduce the set

\[
Y = \{ (\tau, x) \in \mathbb{R}^M \mid l \leq \tilde{B}^T \tau + TEx \leq u, \tau \in \mathbb{R}^N, x \in \mathbb{R}^{M-N} \}.
\] (4.32)
For $Y$ we have

**Proposition 4.1** The set $Y$ is a polytope that is of full dimension in $\mathbb{R}^M$.

**Proof** Write

$$l \leq \tilde{B}^\top \tau + TEx \leq u$$

as

$$l \leq J(\tau, x) \leq u,$$

where $J$ is the $M \times M$ matrix

$$J = \begin{pmatrix} \tilde{B}_1^\top & T I_{M-N} \\ \tilde{B}_2^\top & 0 \end{pmatrix}$$

with $(\tilde{B}_1 \; \tilde{B}_2)$ being the partitioning of $\tilde{B}$ into its first $M - N$ columns and the remaining $N$ columns that correspond to the $N$ arcs of $T$. Since $\tilde{B}_2$ is invertible, $J$ is also invertible, notably with inverse

$$J^{-1} = \begin{pmatrix} 0 & (\tilde{B}_2^{-1})^\top \\ \frac{1}{T} I_{M-N} & -\frac{1}{T} (\tilde{B}_2^{-1} \tilde{B}_1)^\top \end{pmatrix}.$$  

Hence, the solutions to (4.34) are the elements of $\{ J^{-1}z \mid l \leq z \leq u \}$. Then, it suffices to note that $\{ z \mid l \leq z \leq u \}$ is of dimension $M$ because $l < u$. \hfill $\Box$

**Corollary 4.1** The phase shift polytope $\hat{\Lambda}(\mathbb{R})$ is of full dimension in $\mathbb{R}^{M-N}$.

**Proof** If $\hat{\Lambda}(\mathbb{R})$ would not have dimension $M-N$, then there is a non-zero vector $a \in \mathbb{R}^{M-N}$ and number $b \in \mathbb{R}$ such that $a^\top x = b$ for all $x \in \hat{\Lambda}(\mathbb{R})$. However, such an implicit equality is also valid for $Y$, thereby denying that $Y$ is of full dimension in $\mathbb{R}^M$ and, hence, contradicting Proposition 4.1. \hfill $\Box$

In Section 4.3 we will see that $Y$ plays an important role in our method to generate phase shifts, so that its use reaches further than this section alone.

The following propositions provide more insight in the structure of $\hat{\Lambda}(\mathbb{R})$. The first one shows that $\hat{\Lambda}(\mathbb{R})$ is centrally symmetric and the last two deal with the boundary $\partial \hat{\Lambda}(\mathbb{R})$ of $\hat{\Lambda}(\mathbb{R})$. Recall that, if $c \in \{0, \pm 1 \}^M$ is an elementary circuit of $G$, then $\tilde{c} \in \{0, \pm 1 \}^{M-N}$ equals $c$ with its last $N$ entries deleted.

**Proposition 4.2** $\hat{\Lambda}(\mathbb{R})$ is centrally symmetric around $\frac{1}{2T} F(l + u)$.

**Proof** Define

$$m = \frac{1}{2T} (l + u)$$

and let $z \in \mathbb{R}^N$ be such that

$$EFm = m + \tilde{B}^\top z.$$  

(4.38)
Note that such \( z \) exists and is unique, because \( EF - I = -(0 \ S^T) \) so that \( \hat{B}^T z \) is indeed in the cut space of \( \mathcal{G} \). Let \( \hat{x} \in \hat{\Lambda}(\mathbb{R}) \) and choose \( \hat{\tau} \in K(\hat{x}) \). Then,

\[
l \leq \hat{B}^T \hat{\tau} + TE \hat{x} \leq u ,
\]

which is equivalent to

\[
l \leq \hat{B}^T \hat{\tau} + 2TEFm + TE \hat{x} - 2TEFm \leq u .
\]

Using (4.38), we arrive at

\[
l \leq \hat{B}^T (\hat{\tau} + 2Tz) + 2Tm - TE(2Fm - \hat{x}) \leq u .
\]

Combining this with (4.37) yields

\[
l \leq \hat{B}^T (-\hat{\tau} - 2Tz) + TE(2Fm - \hat{x}) \leq u ,
\]

which implies that \( 2Fm - \hat{x} \in \hat{\Lambda}(\mathbb{R}) \). Hence, \( \hat{\Lambda}(\mathbb{R}) \) is centrally symmetric with center \( Fm \).

**Proposition 4.3** For each elementary circuit \( c \) the constraint \( \hat{c}^T x = U(c) \) describes a facet of \( \hat{\Lambda}(\mathbb{R}) \).

**Proof** Let \( c_i \) denote the \( i \)-th elementary circuit of \( \mathcal{G} \) (i.e., \( c_i^T \) is the \( i \)-th row of \( C \)). Now, fix \( i \) and consider the problem

\[
\max \{ \hat{c}_i^T x \mid \hat{C} x \leq U(C), \ x \in \mathbb{R}^{M-N} \}
\]

and its dual

\[
\min \{ U(C)^T y \mid \hat{C}^T y = \hat{c}_i, \ y \geq 0, \ y \in \mathbb{R}^n \},
\]

which is equivalent to

\[
\min \{ U(C)^T y \mid C^T y = c_i, \ y \geq 0, \ y \in \mathbb{R}^n \}.
\]

It is clear that \( c_i \) is dual feasible, with objective function value \( U(c_i) \). It is optimal since for all dual feasible vectors \( y \) we have

\[
T \cdot U(C)^T y = \sum_{j=1}^k y_j (u^T c_j^+ - l^T c_j^-)
\]

\[
\geq u^T \left( \sum_{j=1}^k y_j c_j \right)^+ - l^T \left( \sum_{j=1}^k y_j c_j \right)^-
\]

\[
= T \cdot U(C^T y)
\]

\[
= T \cdot U(c_i)
\]

\[
= T \cdot U(C)^T c_i .
\]
Now suppose that \( y \geq 0 \) is optimal and has more than one nonzero entry. Since \( c_i \) is an elementary circuit, there must be an arc \( m \) of \( \mathcal{G} \) not in \( T \), for which the \( m \)-th entry of the circuit vectors \( c_j \) corresponding to \( y_j > 0 \) cannot all have the same sign (where 0 is taken to have both positive and negative sign). Then, \( u > l \) implies that inequality (4.47) is strict. It follows that \( y = e_i \) is the unique dual optimal solution. Since all but one element of this unique dual optimal solution is zero, the set of solutions to the primal problem with the same value forms a facet of the primal feasible region, which proves the result.

**Proposition 4.4** Let \( x \in \bar{\Lambda}(\mathbb{R}) \). Then, the polytope \( K(x) \) has an implicit equality if and only if \( x \in \partial \bar{\Lambda}(\mathbb{R}) \). Moreover, implicit equalities of \( K(\mathbf{x}) \) are all of the form \( \tau_i - \tau_j = l_{ij} - T(E\mathbf{x})_{ij} \) or \( \tau_i - \tau_j = u_{ij} - T(E\mathbf{x})_{ij} \) for vertices \( \tau_i, \tau_j \) of \( \mathcal{G} \).

**Proof** If \( x \in \partial \bar{\Lambda}(\mathbb{R}) \), then there is an elementary circuit \( c \) of \( \mathcal{G} \) for which \( c^T x = U(c) \), so that

\[
c^T E x = U(c) \, .
\]  

Consider the system

\[
l - T E x \leq \hat{B}^T \tau \leq u - T E x \, .
\]  

Adding

\[
(c^+)^T \hat{B}^T \tau \leq (c^+)^T (u - T E x)
\]  

and

\[
(c^-)^T (l - T E x) \leq (c^-)^T \hat{B}^T \tau
\]  

yields

\[
0 = c^T \hat{B}^T \tau
\]
\[
= (c^+ - c^-)^T \hat{B}^T \tau
\]
\[
\leq (c^+)^T (u - T E x) - (c^-)^T (l - T E x)
\]
\[
= T U(c) - T c^T E x
\]
\[
= 0 \, ,
\]

so that all inequalities in (4.53) and (4.54) hold implicitly at equality.

Conversely, let \( x \in \bar{\Lambda}(\mathbb{R}) \) and suppose that \( K(x) \) has an implicit equality, that is the system (4.52) implies an equality, say

\[
\begin{cases}
y_1^T (u - T E x) - y_2^T (l - T E x) = 0 \\
y_1^T \hat{B}^T - y_2^T \hat{B}^T = 0
\end{cases}
\]  

for some \((y_1, y_2) \in \mathbb{R}^{2M}_+ \), not equal to 0. Define

\[
L = \{ (z_1, z_2) \in \mathbb{R}^{2M}_+ \mid \hat{B}(z_1 - z_2) = 0 \} \, .
\]

In Chapter 5, Lemma 5.3, it is shown that the extremal rays of \( L \) encode the elementary circuits of \( \mathcal{G} \). Hence,

\[
z_1^T (u - T E x) - z_2^T (l - T E x) \geq 0
\]  

for all \((z_1, z_2) \in L \), so that for \((y_1, y_2) \) an extremal ray of \( L \) can be chosen. Therefore, \( x \in \partial \bar{\Lambda}(\mathbb{R}) \). □
4.2.2 A distance measure on the phase shift polytope

We define a distance measure on $\hat{\Lambda}(\mathbb{R})$ that reflects that we are indifferent between vectors whose difference is in the cut space of $\mathcal{G}$. To this end, define the following distance measure.

**Definition 4.1** Let $a, b \in \hat{\Lambda}(\mathbb{R})$. Then $\rho(a, b)$ measures the distance between $a$ and $b$ by

$$\rho(a, b) = \min \{ \| E a - E b + S^T z \|_2 \mid z \in \mathbb{R}^N \},$$

where $\| \cdot \|_2$ is the Euclidian norm. \(\square\)

**Lemma 4.1** $\rho$ is a well defined distance measure on $\hat{\Lambda}(\mathbb{R})$.

**Proof** Trivially, $\rho(a, b) = 0$ if and only if $a = b$, since two vectors in $\hat{\Lambda}(\mathbb{R})$ cannot differ by a cut of $\mathcal{G}$. Furthermore,

$$\rho(a, b) = \min \{ \| E a - E b + S^T z \|_2 \mid z \in \mathbb{R}^N \} \leq \| E a - E b \|_2 \leq \max \{ \| x - y \|_2 \mid x, y \in \hat{\Lambda}(\mathbb{R}) \} < \infty,$$

because $\hat{\Lambda}(\mathbb{R})$ is bounded. Finally, for $a, b, c \in \hat{\Lambda}(\mathbb{R})$ we derive

$$\rho(a, b) = \min \{ \| E a - E b + S^T z \|_2 \mid z \in \mathbb{R}^N \} \leq \min \{ \| E a - E c + S^T z + E c - E b + S^T y \|_2 \mid y, z \in \mathbb{R}^N \} \leq \min \{ \| E a - E c + S^T z \|_2 + \| E c - E b + S^T y \|_2 \mid y, z \in \mathbb{R}^N \} = \rho(a, c) + \rho(c, b),$$

implying the triangle inequality. \(\square\)

This definition remains the same if instead of $S$ we would have used any other $N \times M$ matrix whose rows also span the cut space of $\mathcal{G}$. In particular, we might use $\hat{B}$ for it. Hence, the particular choice of spanning tree $\hat{T}$ is of no influence to the definition of $\rho$. More precise, let $\hat{T}$ be another spanning tree and let $\rho$ and $\hat{\rho}$ be the corresponding definitions of the distance measure in compliance with definition (4.1). Also, denote the corresponding phase shift polytopes with $\hat{\Lambda}_T(\mathbb{R})$ and $\hat{\Lambda}_{\hat{T}}(\mathbb{R})$ and let $a, b \in \hat{\Lambda}_T(\mathbb{R})$ and $\hat{a}, \hat{b} \in \hat{\Lambda}_{\hat{T}}(\mathbb{R})$ such that $E a - \hat{\hat{E}a}$ and $E b - \hat{\hat{E}b}$ are in the cut space of $\mathcal{G}$. Matrix $\hat{E}$ is defined like $E$ and places $N$ zero’s back onto the positions corresponding to $\hat{T}$. Then,

$$\rho(a, b) = \hat{\rho}(\hat{a}, \hat{b}).$$

**Lemma 4.2** Let $a, b \in \hat{\Lambda}(\mathbb{R})$. Then,

$$\rho(a, b) = \| Q E (a - b) \|_2,$$

where

$$Q = I_M - S^T (SS^T)^{-1} S,$$

is an $M \times M$ matrix.
Proof Evaluating (4.63) amounts to projecting the vector $Eb$ onto the affine space $W = \{ Ea + S^Tz \mid z \in \mathbb{R}^N \}$. This is equivalent to projecting the vector $E(b - a)$ onto the linear space $\{ S^Tz \mid z \in \mathbb{R}^N \}$ and adding $Ea$ to the result. Thus, we have that the projection of $Eb$ onto $W$ is equal to $z = Ea + S^T(SS^T)^{-1}SE(b - a)$. The solution to (4.63) is now the Euclidean distance between $z$ and $Eb$, which yields the result. □

4.2.3 The volume function

The robustness distribution, that is to be introduced in Section 4.2.5, is based on the so-called volume function, which we will define in this section. To this end, for $x \in \tilde{\Lambda}(\mathbb{R})$ we denote with $\text{dim}(x)$ the geometric dimension of $K(x)$.

Definition 4.2 Let $X_n = \{ x \in \tilde{\Lambda}(\mathbb{R}) \mid \text{dim}(x) \leq n \}$. Then, the function $v_n : X_n \rightarrow \mathbb{R}$ maps $x \in X_n$ onto the $n$ dimensional geometric volume (Lebesgue measure) of $K(x)$. For $n = 0$ we define $v_0(x) = 1$ for all $x \in X_0$. □

If $\text{dim}(x) < N$, then $v_N(x) = 0$. Clearly, $v_N$ has domain $\tilde{\Lambda}(\mathbb{R})$ and we refer to $v_N$ as the volume function and use it to define the volume distribution and the robustness distribution in the two following sections. Here we derive a few properties of $v_N$.

Proposition 4.5 $v_N^{1/N}$ is concave and $v_N(x) = 0$ if and only if $x \in \partial \tilde{\Lambda}(\mathbb{R})$.

Proof From the definition of the set $K(x)$ it follows that, for all $x, \tilde{x} \in \tilde{\Lambda}(\mathbb{R})$ and $0 \leq \lambda \leq 1$,

\begin{align}
\lambda K(x) + (1 - \lambda)K(\tilde{x}) &= \{ \lambda \tau + (1 - \lambda)\tilde{\tau} \mid l \leq \tilde{B}^T\tau + TE\tilde{x} \leq u \} \\
&\subseteq \{ \lambda \tau + (1 - \lambda)\tilde{\tau} \mid l \leq \tilde{B}^T(\lambda \tau + (1 - \lambda)\tilde{\tau}) + T(\lambda E\tilde{x} + (1 - \lambda)E\tilde{x}) \leq u \} \\
&= \{ \tau \mid l \leq \tilde{B}^T\tau + T(\lambda E\tilde{x} + (1 - \lambda)E\tilde{x}) \leq u \} \\
&= K(\lambda x + (1 - \lambda)\tilde{x}).
\end{align}

We then have

\begin{align}
v_N^{1/N}(\lambda x + (1 - \lambda)\tilde{x}) &= \mu^{1/N}(K(\lambda x + (1 - \lambda)\tilde{x})) \\
&\geq \mu^{1/N}(\lambda K(x) + (1 - \lambda)K(\tilde{x})) \\
&\geq \lambda \mu^{1/N}(K(x)) + (1 - \lambda)\mu^{1/N}(K(\tilde{x})) \\
&= \lambda v_N^{1/N}(x) + (1 - \lambda)v_N^{1/N}(\tilde{x}),
\end{align}

where $\mu$ denotes the Lebesgue measure and (4.81) follows from the Brunn-Minkowski Theorem [19].

The second part of the proposition follows directly from Proposition 4.4, once we realize that $v_N(x) = 0$ if and only if $K(x)$ has an implicit equality. □

Corollary 4.2 $v_N$ is a bounded, continuous function on $\tilde{\Lambda}(\mathbb{R})$. 
Proof Continuity follows from the concavity of $v_N^{1/N}$. Since $v_N$ is a continuous function on a compact subset of $\mathbb{R}^{M-N}$, it is also bounded. □

In conclusion of this section, define

$$x_{\text{max}} = \arg\max \{ v_N(x) \mid x \in \hat{\Lambda}(\mathbb{R}) \} .$$

(4.83)

The following results yield an analytical expression for $x_{\text{max}}$.

Proposition 4.6 $v_N$ is centrally symmetric around $\frac{1}{2T}F(l+u)$.

Proof Like in the proof of Proposition 4.2, define

$$m = \frac{1}{2T}(l+u)$$

(4.84)

and let $z \in \mathbb{R}^N$ be such that

$$EFm = m + \hat{B}^Tz .$$

(4.85)

Then, for $x \in \hat{\Lambda}(\mathbb{R})$ we have

$$K(x) = \{ \tau \mid l \leq \hat{B}^T \tau + TEx \leq u \}$$

(4.86)

$$= \{ \tau \mid l \leq \hat{B}^T \tau + 2TEFm + TEx - 2TEFm \leq u \}$$

(4.87)

$$= \{ \tau \mid l \leq \hat{B}^T(\tau + 2Tz) + 2Tm - TE(2Fm - x) \leq u \}$$

(4.88)

$$= \{ \tau \mid l \leq \hat{B}^T(\tau - 2Tz) + TE(2Fm - x) \leq u \}$$

(4.89)

$$= -K(2Fm - x) - 2Tz .$$

(4.90)

Hence, $v_N(x) = v_N(2Fm - x)$, implying the result. □

Corollary 4.3

$$x_{\text{max}} = \frac{1}{2T}F(l+u) .$$

(4.91)

Proof Recall from Proposition 4.5 that $v_N^{1/N}$ is concave, so that Proposition 4.6 implies the result. □

4.2.4 The volume distribution

We introduce the volume distribution $V$ on $\hat{\Lambda}(\mathbb{R})$, which is formally defined by

Definition 4.3 For all measurable sets $A \subseteq \hat{\Lambda}(\mathbb{R})$, we define the probability density

$$V(A) = \frac{\int_A v_N(y) \, dy}{\int_{\hat{\Lambda}(\mathbb{R})} v_N(y) \, dy} ,$$

(4.92)

and call it the volume distribution on $\hat{\Lambda}(\mathbb{R})$. □
Note that \( f_A v_N(y) \, dy \) is well defined on account of previous results about \( \hat{\Lambda}(\mathbb{R}) \) and \( v_N \).

Like with distance measure \( \rho \), the definition of \( \mathcal{V} \) is not sensitive to the choice of the spanning tree \( T \). Below, by means of Proposition 4.7, we will explain what we mean by this. For this purpose, like in Section 4.2.2, let \( \tilde{T} \) be another spanning tree and make all definitions in this chapter that depend on \( T \) also for \( \tilde{T} \). To distinguish between them, we may capitate symbols with a hat or add subscripts \( T \) or \( \tilde{T} \). For example, we use the notations \( \mathcal{V} \) and \( \hat{\mathcal{V}} \) to denote the volume distributions on the phase shift polytopes \( \hat{\Lambda}(\mathbb{R}) \) and \( \hat{\Lambda}(\mathbb{R}) \), respectively. Moreover, \( F \) and \( \hat{F} \) are the corresponding projection matrices as defined in Section 4.1.1. These transformations are in fact mappings from \( \mathbb{R}^M \) to \( \mathbb{R}^{M-N} \) and for \( T \) we introduced the matrix \( E \) that brings \( x \in \Lambda(\mathbb{R}) \) back to \( \Lambda(\mathbb{R}) \) by appending \( N \) zero's to \( x \) to denote explicitly that for a phase shift its entries corresponding to \( T \) are set to zero. We define \( \hat{E} \) similarly. Also, we introduce the \( M \times M \) permutation matrix \( \hat{P} \) that, given \( x \in \Lambda(\mathbb{R}) \) moves the entries of \( x \) corresponding to tree \( \tilde{T} \) to the last \( N \) positions. Likewise, we introduce matrix \( P \) for the spanning tree \( T \).

We have

\[ W = \hat{F} \hat{P} E. \tag{4.93} \]

Then \( W \) is the matrix of a one-to-one mapping from \( \hat{\Lambda}(\mathbb{R}) \) onto \( \hat{\Lambda}(\mathbb{R}) \) with inverse

\[ W^{-1} = FP\hat{E}. \tag{4.94} \]

**Proof** Clearly, if \( x \in \hat{\Lambda}(\mathbb{R}) \), then \( Wx \in \hat{\Lambda}(\mathbb{R}) \) and, if \( \hat{x} \in \hat{\Lambda}(\mathbb{R}) \), then \( W^{-1}\hat{x} \in \hat{\Lambda}(\mathbb{R}) \). Furthermore, both the transformations \( W \) and \( W^{-1} \) only add a vector in the cut space of \( G \) to the original. Since no two vectors in \( \hat{\Lambda}(\mathbb{R}) \) differ by a vector in the cut space of \( G \) and, since the same holds true for \( \hat{\Lambda}(\mathbb{R}) \), it follows that \( W^{-1}W = WW^{-1} = I_{M-N} \). \( \Box \)

From this lemma we learn that, if \( A \subseteq \hat{\Lambda}(\mathbb{R}) \), then \( WA \subseteq \hat{\Lambda}(\mathbb{R}) \). Moreover, \( x \in A \) if and only if \( Wx \in WA \). The following proposition states that the probability mass that \( \mathcal{V} \) assigns to \( A \) (assuming it is measurable) is the same as \( \hat{\mathcal{V}} \) assigns to \( WA \). Hence, the choice of the spanning tree is immaterial.

**Proposition 4.7** Let \( A \subseteq \hat{\Lambda}(\mathbb{R}) \) be measurable. Then,

\[ \hat{\mathcal{V}}(WA) = \mathcal{V}(A). \tag{4.95} \]

**Proof** Using

\[ \hat{\Lambda}(\mathbb{R}) = W\Lambda(\mathbb{R}) \tag{4.96} \]

we have

\[ \hat{\mathcal{V}}(WA) = \frac{f_{WA} v_N(y) \, dy}{f_{W\hat{\Lambda}(\mathbb{R})} v_N(y) \, dy} \tag{4.97} \]
where we use that
\[ v_N(Wy) = v_N(y) \] (4.101)
in the third equality. Also, in the second equality we use that the Jacobian of \( W \) is factored out in the division.

\[ \square \]

### 4.2.5 The robustness distribution

Using the volume distribution as defined in the previous section we define a discrete probability distribution \( \mathcal{R} \) on \( \tilde{\Lambda}(\mathbb{Z}) \). In particular, \( \mathcal{R} \) will be the conditionalization of \( \mathcal{V} \) onto \( \tilde{\Lambda}(\mathbb{Z}) \). To this end, for \( x \in \tilde{\Lambda}(\mathbb{R}) \) and small \( \epsilon \geq 0 \), let
\[ B_\epsilon(x) = \{ y \in \tilde{\Lambda}(\mathbb{R}) \mid \rho(x, y) \leq \epsilon \} \] (4.102)
be the intersection of \( \tilde{\Lambda}(\mathbb{R}) \) and the ball around \( x \) with radius \( \epsilon \), or, equivalently,
\[ B_\epsilon(x) = \{ y \in \tilde{\Lambda}(\mathbb{R}) \mid (x - y)^T E^T Q^T E(x - y) \leq \epsilon^2 \} \] (4.103)

**Definition 4.4** The robustness distribution \( \mathcal{R} \) on \( \tilde{\Lambda}(\mathbb{Z}) \) is defined by
\[ \mathcal{R}(p) = \lim_{\epsilon \downarrow 0} \frac{\mathcal{V}(B_\epsilon(p))}{\sum_{\tilde{p} \in \tilde{\Lambda}(\mathbb{Z})} \mathcal{V}(B_\epsilon(\tilde{p}))} \] (4.104)
for all \( p \in \tilde{\Lambda}(\mathbb{Z}) \).

Like with \( \mathcal{V} \), the choice of \( T \) does not matter. To see this, recall that \( \rho \) is insensitive to the particular choice of \( T \) and then take \( A = B_\epsilon(x) \) in Proposition 4.7.

We will show that \( \mathcal{R} \) only assigns positive probability to phase shift \( p \) if it has the largest value for \( \dim(p) \) among all phase shifts. Recall that \( \dim(p) \) is the dimension of \( K(p) \). The practical meaning of this stems from the idea that a timetable planner always seeks for timetables that are robust with respect to small changes of a timetable while planning or executing it. Put in a mathematical context, if \( \dim(p) = N \), then for a timetable \( \tau \) that is chosen from the interior of \( K(p) \) none of the interval constraints is binding in a small neighborhood of \( \tau \), so that small perturbations of \( \tau \) are still in \( K(p) \). It is for this reason that we call \( \mathcal{R} \) the robustness distribution.

Define
\[ \eta = \max \{ \dim(p) \mid p \in \tilde{\Lambda}(\mathbb{Z}) \} \] (4.105)
Thus, we will show for \( p \in \Lambda(\mathcal{Z}) \) that \( R(p) > 0 \) if and only if \( \dim(p) = \eta \).

Let \( p \in \Lambda(\mathcal{Z}) \). By switching to polar coordinates, we can write

\[
V(B_r(p)) = \frac{\int_0^r \int_{\Theta_p} v_N(p + r\theta) r^{M-N-1} \omega(\theta) \, d\theta \, dr}{\int_{\Lambda(\mathcal{R})} v_N(y) \, dy},
\]

where \( \omega(\theta) \, d\theta \) is a surface element of \( \{ y \in \mathbb{R}^{M-N} \mid \rho(y, 0) \leq 1 \} \) and

\[
\Theta_p = \{ \theta \in \mathbb{R}^{M-N} \mid \rho(\theta, 0) = 1 \text{ and } p + \delta \theta \in \Lambda(\mathcal{R}) \text{ for some } \delta > 0 \}
\]

is the set of (normalized) directions in which we can perturb \( p \) such that the result is still in \( \Lambda(\mathcal{R}) \). Note that by virtue of Corollary 4.1, \( \Theta_p \) is a curvature of dimension \( M-N-1 \). Also, we define the interior \( \text{int}(\Theta_p) \) of \( \Theta_p \) by

\[
\text{int}(\Theta_p) = \{ \theta \in \Theta_p \mid \text{there is a } \delta > 0 \text{ such that } \|\theta - \hat{\theta}\|_2 < \delta \text{ and } \rho(\hat{\theta}, 0) = 1 \text{ imply } \hat{\theta} \in \Theta_p \},
\]

Note that there exists a \( \delta > 0 \) such that \( v_N(p + \epsilon \theta) > 0 \) for all \( 0 < \epsilon < \delta \) if and only if \( \theta \in \text{int}(\Theta_p) \).

Before we continue, recall or understand that the following statements are equivalent

1. \( p \in \Lambda(\mathcal{Z}) \) lies in the interior of \( \Lambda(\mathcal{R}) \),
2. \( \dim(p) = N \),
3. \( v_N(p) > 0 \),
4. \( \Theta_p = \{ \theta \in \mathbb{R}^{M-N} \mid \rho(\theta, 0) = 1 \} \),
5. \( \text{int}(\Theta_p) = \Theta_p \).

By l’Hôpital’s rule, we obtain that the robustness distribution \( R \) equals

\[
R(p) = \lim_{\epsilon \downarrow 0} \frac{\phi_p(\epsilon)}{\sum_{\hat{p} \in \hat{\Lambda}(\mathcal{Z})} \phi_{\hat{p}}(\epsilon)},
\]

where

\[
\phi_p(\epsilon) = \int_{\Theta_p} v_N(p + \epsilon \theta) \omega(\theta) \, d\theta.
\]

From (4.109) and (4.110) we see that, if \( \eta = N \), then

\[
R(p) = \frac{v_N(p)}{\sum_{\hat{p} \in \hat{\Lambda}(\mathcal{Z})} v_N(\hat{p})}
\]

for all \( p \in \Lambda(\mathcal{Z}) \). In particular, in this case we find that \( R(p) > 0 \) if and only if \( \dim(p) = \eta \), precisely as we planned to show. In general, however, it may occur that \( v_N(p) = 0 \) for all
\( p \in \hat{\Lambda}(\mathbb{Z}) \). The following proposition formalizes the interpretation of \( \mathcal{R} \) in this case. In the proof of it we make use of a couple of lemma's that state results that are simple and intuitively clear. However, their proofs are quite involving and of substantial length, so that we will present them after the proposition. The casual reader, then, may take these lemma’s for granted and proceed with the next section. We have

**Proposition 4.8** Let \( p \in \hat{\Lambda}(\mathbb{Z}) \) and assume that \( \eta < N \). Then, it holds true that \( \mathcal{R}(p) > 0 \) if and only if \( \text{dim}(p) = \eta \).

**Proof** By Lemma 4.6 it follows that for small enough \( \epsilon \geq 0 \)

\[
\frac{\phi_\epsilon(p)}{\epsilon^{N-\text{dim}(p)-1}} = \frac{1}{\epsilon^{N-\text{dim}(p)-1}} \int_{\Theta_p} v_N(p + \epsilon \theta) \omega(\theta) \, d\theta \tag{4.112}
\]

\[
\leq \epsilon \int_{\Theta_p} \left( v_{\text{dim}(p)}(p) \left( 3T \theta_{\text{max}} N^3 \right)^{N-\text{dim}(p)} + o(\epsilon) \right) \omega(\theta) \, d\theta \tag{4.113}
\]

\[
\rightarrow 0 \quad (\epsilon \downarrow 0), \tag{4.114}
\]

where

\[
\theta_{\text{max}} = \max\{ | \theta_i | \mid i = 1, \ldots, M - N \}, \tag{4.115}
\]

which is bounded because \( \rho(0, \theta) = 1 \) for all \( \theta \in \Theta_p \).

Next, we show that

\[
\lim_{\epsilon \to 0} \frac{\phi_\epsilon(p)}{\epsilon^{N-\text{dim}(p)}} > 0, \tag{4.116}
\]

which settles the proposition. To this end, define the set \( \mathcal{A}_i \subseteq \{1, \ldots, M - N\} \) as the set of arcs of \( \mathcal{G} \) that are not in spanning tree \( T \) and for which the corresponding interval constraints \( l_{ij} - Tp_{ij} \leq \tau_i - \tau_j \leq u - Tp_{ij} \) are implicitly equivalent to \( \tau_i - \tau_j = l_{ij} - Tp_{ij} \) in \( K(p) \). That is, the inequality \( \tau_i - \tau_j \geq l_{ij} - Tp_{ij} \) is binding for all \( \tau \in K(p) \). Likewise, we define \( \mathcal{A}_u \) for implicit equalities of type \( \tau_i - \tau_j = u_{ij} - Tp_{ij} \). Then, define the vectors \( d, \hat{\theta} \in \{0, \pm 1\}^{M-N} \) by

\[
d_m = \begin{cases} -1 & \text{if } m \in \mathcal{A}_u \\ 1 & \text{if } m \in \mathcal{A}_i \\ 0 & \text{otherwise} \end{cases} \tag{4.117}
\]

for all \( m = 1, \ldots, M - N \) and

\[
\hat{\theta} = \frac{d}{\rho(0, d)}. \tag{4.118}
\]

By Lemma 4.4, \( \hat{\theta} \in \text{int}(\Theta_p) \). Moreover, as shown in Lemma 4.5, there are numbers \( D > 0 \) and \( C, \delta \geq 0 \) such that for all \( 0 < \epsilon \leq \delta \) it holds true that

\[
v_N(p + \epsilon \hat{\theta}) \geq D v_{\text{dim}(p)}(p) - C \frac{\epsilon T}{\rho(0, d)} \text{vol}(\partial K(p)) \left( \frac{T}{2N\rho(0, d)} \right)^{N-\text{dim}(p)}, \tag{4.119}
\]

where \( \text{vol}(\partial K(p)) \) is the surface area of \( K(p) \) and is defined 1 if \( \text{dim}(p) \in \{0, 1\} \).
Define the neighborhood $N(\hat{\theta})$ of $\hat{\theta}$ by

$$N(\hat{\theta}) = \{ \theta \in \Theta_p \mid \frac{v_N(p + \epsilon \theta)}{\epsilon^{N-\dim(p)}} \geq D \left( v_{\dim(p)}(p) - \frac{2\epsilon T}{\rho(0, d)} \text{vol}(\partial K(p)) \right) \left( \frac{T}{4N \rho(0, d)} \right)^{N-\dim(p)} \}.$$  \hspace{1cm} (4.120)

for all $0 < \epsilon \leq \frac{1}{2} \delta$.  \hspace{1cm} (4.121)

By examining the proof of Lemma 4.5 it follows that $\Theta_p \cap B \subseteq N(\hat{\theta})$ for some small ball $B \subseteq \mathbb{R}^{M-N}$ around $\hat{\theta}$ with positive radius (also recall that $v_N$ is continuous on $\tilde{A}(\mathbb{R})$ and that $\hat{\theta} \in \text{int}(\Theta_p)$).  Hence,

$$\int_{N(\hat{\theta})} \omega(\theta)\,d\theta > 0$$  \hspace{1cm} (4.123)

and we conclude that

$$\frac{1}{\epsilon^{N-\dim(p)}} \int_{\Theta_p} v_N(p + \epsilon \theta)\omega(\theta)\,d\theta \geq \frac{1}{\epsilon^{N-\dim(p)}} \int_{N(\hat{\theta})} v_N(p + \epsilon \theta)\omega(\theta)\,d\theta \geq \int_{N(\hat{\theta})} D \left( v_{\dim(p)}(p) - \frac{2\epsilon T C}{\rho(0, d)} \text{vol}(\partial K(p)) \right) \left( \frac{T}{4N \rho(0, d)} \right)^{N-\dim(p)} \omega(\theta)\,d\theta \geq Dv_{\dim(p)}(p) \left( \frac{T}{4N \rho(0, d)} \right)^{N-\dim(p)} \cdot \int_{N(\hat{\theta})} \omega(\theta)\,d\theta \quad (\epsilon \downarrow 0)$$  \hspace{1cm} (4.124)

thereby implying the result.  \hspace{1cm} \square

In the remainder of this section we state the lemma's that we referred to in the proof of Proposition 4.8. The proofs of these lemma's involve quite some details and may be difficult to follow when reading them for the first time. Therefore, we will first outline the results and proofs before we go into the details. Once again, the casual reader might choose to skip the remainder of this section.

Before we present the details we make a couple of notational conventions. We fix $p \in \tilde{A}(\mathbb{R})$ and assume that $\dim(p) < N$. Also, we let $W_p$ be the affine $\dim(p)$ dimensional subspace of $\mathbb{R}^N$ containing $K(p)$ and we let $W_p^\perp$ be the $N-\dim(p)$ dimensional subspace of $\mathbb{R}^N$ orthogonal to $W_p$. Also, $\partial K(p)$ is the boundary of $K(p)$ relative to $W_p$ and $\text{vol}(\partial K(p))$ is its $\dim(p)-1$ dimensional volume (surface area), which we define equal to 1 in case $\dim(p) \in \{0, 1\}$. Likewise, if $\dim(p) = 0$, then we define $v_{\dim(p)}(p) = 1$. Note that $K(p)$ is of full dimension in $W_p$.
In Lemma 4.5 we derive a lower bound on \( v_N(p + e\hat{\theta}) \), where \( \hat{\theta} \) is a vector in \( \text{int}(\Theta_p) \) and was introduced in the proof of Proposition 4.8. The lower bound reads

\[
v_N(p + e\hat{\theta}) \geq D \left( v_{\text{dim}(p)}(p) - C \frac{eT}{\rho(0, d)} \text{vol}(\partial K(p)) \right) \left( \frac{eT}{2N\rho(0, d)} \right)^{N-\text{dim}(p)}
\]

(4.129)

for all \( 0 \leq \epsilon \leq \delta \) for some numbers \( D > 0 \) and \( C, \delta \geq 0 \), where \( d \) is defined in the proof of Proposition 4.8. Conceptually, the proof is quite simple. That is, for some of the arcs of \( G \) we strengthen their corresponding interval constraints a little bit, such that the resulting set of timetables \( K^-(p) \) is contained in the interior of \( K(p) \) and such that \( K^-(p) \) has the same dimension as \( K(p) \), namely \( \text{dim}(p) \). Then, we introduce a basis \( w^1, \ldots, w^{N-\text{dim}(p)} \) of \( W_p^\perp \) and show that for any timetable \( \tau \in K^-(p) \) we have

\[
\tau + \frac{eT}{2\rho(0, d)} w^i \in K(p + e\hat{\theta})
\]

(4.130)

for all \( i = 1, \ldots, N - \text{dim}(p) \). Then, the result follows immediately. However, if \( \text{dim}(p) = 0 \), there is a pitfall. That is, in this case \( K(p) \) is a singleton and does not have an interior, so that the existence of the vectors \( w^1, \ldots, w^{N-\text{dim}(p)} \) is not clear. We repair this flaw by constructing another set of \( N \) linearly independent vectors in \( \mathbb{R}^N \) and show that we can move in their directions and still be in \( K(p + e\theta) \), just like in the more general case.

In Lemma 4.6 we show that

\[
v_N(p + e\theta) \leq v_{\text{dim}(p)}(p) \left( 3eT\theta_{\text{max}}N^3 \right)^{N-\text{dim}(p)} + o(e^{N-\text{dim}(p)})
\]

(4.131)

for all \( \theta \in \Theta_p \) and for small enough \( \epsilon \geq 0 \). In the proof of this lemma we expand \( K(p) \) a little bit, so as to create \( K^+(p) \). This is done such that \( K(p) \subseteq K^+(p) \). We introduce an orthonormal system \( w^1, \ldots, w^{N-\text{dim}(p)} \) of \( N - \text{dim}(p) \) vectors in \( W_p^\perp \) and show that any timetable in \( K(p + e\theta) \) can be written as the sum of a vector in \( K^+(p) \) and a linear combination of the vector \( w^1, \ldots, w^{N-\text{dim}(p)} \) with coefficients that are no bigger than \( 3eT\theta_{\text{max}}N^3 \) in absolute value, where \( \theta_{\text{max}} \) is defined in the proof of Proposition 4.8. This yields the result.

At this point we are ready to present the lemma’s and their proofs in full detail. We introduce some more notations. For one thing, we partition the index set of arcs of \( G \) as

\[
\{1, \ldots, M\} = T_t \cup T_u \cup A_t \cup A_u \cup A_\neq,
\]

(4.132)

where \( T_t, T_u \subseteq \{M-N+1, \ldots, M\} \) correspond to arcs in the spanning tree \( T \) and \( A_t, A_u \subseteq \{1, \ldots, M-N\} \). Furthermore, \( T_t \cup A_t \) is the set of arcs of \( G \) for which the corresponding interval constraints \( l_{ij} - T(Ep)_{ij} \leq \tau_i - \tau_j \leq u - T(Ep)_{ij} \) are equivalent to \( \tau_i - \tau_j = l - T(Ep)_{ij} \) in \( K(p) \). That is, the inequality \( \tau_i - \tau_j \leq l - T(Ep)_{ij} \) is binding for all \( \tau \in K(p) \). Likewise we define \( T_u \cup A_u \) for implicit equalities of type \( \tau_i - \tau_j = u - T(Ep)_{ij} \). The set \( A_\neq \subseteq \{1, \ldots, M\} \) contains all remaining arcs. Thus, for \( m \in A_\neq \) there are timetables \( \tau^1, \tau^2 \in K(p) \) such that \( \tau^1_i - \tau^1_j > l_{ij} - T(Ep)_{ij} \) and \( \tau^2_i - \tau^2_j < u_{ij} - T(Ep)_{ij} \).
Recall the definition of the vectors \( d, \hat{\theta} \in \{0, \pm 1\}^{M-N} \) as defined in the proof of Proposition 4.8. That is, they are defined by

\[
d_m = \begin{cases} 
-1 & \text{if } m \in A_u \\
1 & \text{if } m \in A_l \\
0 & \text{otherwise}
\end{cases}
\]  

for all \( m = 1, \ldots, M - N \) and

\[
\hat{\theta} = \frac{d}{\rho(0,d)}.
\]

Lemma 4.4 For \( \hat{\theta} \) defined by (4.134) we have

\[
\hat{\theta} \in \text{int}(\Theta_p).
\]

Proof Note that \( d \neq 0 \) because \( \dim(p) < N \), so that there must be an arc \( m \not\in T \) for which \( d_m \neq 0 \). Also, note that \( \rho(0,d) > 0 \) because \( (Ed)_m = 0 \) for all tree arcs \( m = M - N + 1, \ldots, M \), so that \( Ed \) is not a cut of \( G \). Furthermore, \( \rho(0,\hat{\theta}) = 1 \) and, moreover, \( Kp + \epsilon \hat{\theta} \) is of full dimension for small \( \epsilon > 0 \). To see the latter statement, recall from Proposition 4.4 that implicit equalities arise from elementary circuits \( c \) for which \( c^T p = U(c) \). Since adding \( \epsilon \hat{\theta} \) to \( p \) for small \( \epsilon > 0 \) comes down to relaxing at least one arc interval that corresponds to \( c \) (note that not all arcs of \( c \) can be in \( T \)), such an identity is relaxed to \( c^T p < U(c) \), so that the implied equalities are lost. Therefore, we conclude that \( \hat{\theta} \in \text{int}(\Theta_p) \).

Proof We first assume that \( \dim(p) > 0 \) and deal with the case \( \dim(p) = 0 \) later. Note that, by Lemma 4.4, \( \hat{\theta} \in \text{int}(\Theta_p) \) and \( v_N(p + \epsilon \hat{\theta}) > 0 \) for all \( \epsilon > 0 \) that are small enough.

For \( 0 \leq \epsilon \leq \delta \) define the vectors \( \hat{u}, \hat{u} \in \mathbb{R}^M \) by setting for all \( m = 1, \ldots, M \)

\[
\hat{u}_m = \begin{cases} 
u_m - \frac{\epsilon T}{\rho(0,d)} & \text{if } m \in A_\pm \\
\nu_m & \text{otherwise}
\end{cases}
\]  

and

\[
\hat{l}_m = \begin{cases} \nu_m + \frac{\epsilon T}{\rho(0,d)} & \text{if } m \in A_\pm \\
\nu_m & \text{otherwise}
\end{cases}
\]
Then, consider the set
\[ K^-(p) = \{ \tau \in K(p) \mid \tilde{l} - T E p \leq \tilde{B}^T \tau \leq \hat{u} - T E p \} \]  
(4.140)

If \( c \) is an elementary circuit of \( G \) with \( U(c) - \hat{c}^T p \geq \frac{1}{T} \) (recall that \( l \) and \( u \) are integral and that \( \hat{c} \) equals \( c \) with its last \( N \) entries deleted), then
\[ \frac{1}{T} (\hat{u}^T c^+ - \hat{l}^T c^-) \geq \frac{1}{T} \left( (u - \frac{\epsilon T}{\rho(0,d)} c)^T c^+ - (l + \frac{\epsilon T}{\rho(0,d)} c)^T c^- \right) \]  
(4.141)
\[ \geq \frac{1}{T} \left( (u - \frac{1}{N+2} c)^T c^+ - (l + \frac{1}{N+2} c)^T c^- \right) \]  
(4.142)
\[ \geq U(c) - \frac{1}{T} \frac{N+1}{N+2} \]  
(4.143)
\[ \geq \hat{c}^T p \]  
(4.144)

On the other hand, if \( U(c) - \hat{c}^T p = 0 \), then all arcs incident to \( c \) are not in \( A_\# \) and
\[ \frac{1}{T} (\hat{u}^T c^+ - \hat{l}^T c^-) = \frac{1}{T} (u^T c^+ - l^T c^-) \]  
(4.145)
\[ = U(c) \]  
(4.146)
\[ = \hat{c}^T p \]  
(4.147)

Therefore, the dimension of \( K^-(p) \) is determined by the same implicit equalities as the equalities that determine \( K(p) \). Consequently, \( K^-(p) \subset W_p \) is of the same dimension as \( K(p) \). Moreover, since all interval constraints corresponding to the indices in \( A_\# \) are perturbed, \( K^-(p) \) is in the interior of \( K(p) \) relative to \( W_p \), i.e.
\[ K^-(p) \subset \text{relint}(K(p)) \]  
(4.148)

and
\[ \text{vol}(K^-(p)) \geq v_{\text{dim}(p)}(p) - C \frac{\epsilon T}{\rho(0,d)} \text{vol}(\partial K(p)) \]  
(4.149)

for some number \( C \geq 0 \) that does not depend on \( \epsilon \), where \( \text{vol}(K^-(p)) \) is the \( \text{dim}(p) \) dimensional volume of \( K^-(p) \). Note that \( C \) does not depend on \( \epsilon \) because the vector products of the normals of the hyperplanes that define \( K(p) \) and \( K^-(p) \) are restricted to \( \{-2,-1,0,1,2\} \).

Let \( \tau \in K(p) \). If \( m \in A_u \), then
\[ (\hat{B}^T \tau)_m = (u - T E p)_m \]  
(4.150)

Moreover,
\[ 0 < -\epsilon T \hat{\theta}_m \]  
(4.151)
\[ = \frac{\epsilon T}{\rho(0,d)} \]  
(4.152)
\[ \leq \frac{1}{N+2} \]  
(4.153)
\[ < 1 \]  
(4.154)
Therefore,
\[(l - TEp - \epsilon T E \hat{\theta})_m \leq (u - \epsilon - TEp - \epsilon T E \hat{\theta})_m \leq (\hat{B}^T \tau - \epsilon - \epsilon T E \hat{\theta})_m \leq (\hat{B}^T \tau)_m \leq (u - TEp - \epsilon T E \hat{\theta})_m .\] 

Likewise, we find that
\[(l - TEp - \epsilon T E \hat{\theta})_m \leq (\hat{B}^T \tau)_m \leq (u - TEp - \epsilon T E \hat{\theta})_m \]
for all \(m \in \mathcal{A}_u\). For all other arcs the arc intervals are not affected by adding \(\epsilon \hat{\theta}\) to \(p\). Hence,
\[\tau \in K(p + \epsilon \hat{\theta}) ,\]
so that
\[K(p) \subset K(p + \epsilon \hat{\theta}) \]
for all \(0 \leq \epsilon \leq \delta\). Since \(K(p + \epsilon \hat{\theta})\) is convex and of dimension \(N\), (4.148) and (4.161) give that there exists a basis \(\{w^i_1, \ldots, w^{N - \dim(p)}_p\}\) of \(W^+_p\) such that for all \(\tilde{\tau} \in K^{-}(p)\) there are strictly positive numbers \(\alpha_1, \ldots, \alpha_{N - \dim(p)}\) for which \(\tilde{\tau} + \alpha_i w^i \in K(p + \epsilon \hat{\theta})\) for all \(i = 1, \ldots, N - \dim(p)\). We will show that
\[\alpha_i \geq \frac{\epsilon T}{2\rho(0,d)} \]
holds for all \(\tilde{\tau} \in K^{-}(p)\) and for all \(i = 1, \ldots, N - \dim(p)\). This, then, implies that
\[\{ \tilde{\tau} + \sum_{i=1}^{N - \dim(p)} \alpha_i w^i \mid 0 \leq \alpha_i \leq \frac{\epsilon T}{2N \rho(0,d)} , \ i = 1,\ldots, N - \dim(p) \} \subseteq K(p + \epsilon \hat{\theta}) ,\]
by convexity of \(K(p + \epsilon \hat{\theta})\), so that
\[v_N(p + \epsilon \hat{\theta}) \geq \gamma \operatorname{vol}(K^{-}(p)) \left(\frac{\epsilon T}{2N \rho(0,d)}\right)^{N - \dim(p)} \geq \gamma \left(v_{\dim(p)}(p) - C \frac{\epsilon T}{\rho(0,d)} \operatorname{vol}(\partial K(p))\right) \left(\frac{\epsilon T}{2N \rho(0,d)}\right)^{N - \dim(p)} ,\]
for some number \(\gamma > 0\), thereby implying the lemma.

To show (4.162), let \(\tilde{\tau} \in K^{-}(p)\) and choose \(i\) in \(\{1, \ldots, N - \dim(p)\}\). Let \(m \in \{1, \ldots, M\}\) and consider a few cases. Firstly, suppose that \(m \in \mathcal{A}_u\). Then,
\[\left(\tilde{B}^T \left(\tilde{\tau} + \frac{\epsilon T}{2\rho(0,d)} w^i\right)\right)_m = \left(u - TEp + \frac{\epsilon T}{2\rho(0,d)} \tilde{B}^T w^i\right)_m \leq (u - TEp)_m + \frac{\epsilon T}{\rho(0,d)} \]
\[= (u - TEp - \epsilon T E \hat{\theta})_m .\]
The inequality follows from $\|w^i\|_2 = 1$. Note that, since $u \geq l + \epsilon$, for $0 \leq \epsilon \leq \delta$

\[
\left( \bar{B}^T (\tilde{\tau} + \frac{\epsilon T}{2 \rho(0, d)} w^i) \right)_m \geq (l - T E p + \epsilon)_m - \frac{\epsilon T}{\rho(0, d)} \geq (l - E T p)_m .
\]

If $m \in A_a$, the same conclusion results. Secondly, assume that $m \in T_a$. If $(\bar{B}^T w^i)_m > 0$, then

\[
\left( \bar{B}^T (\tilde{\tau} + \beta w^i) \right)_m > (u - T E p)_m ,
\]

for all $\beta > 0$. This contradicts the earlier made observation that $\bar{B}^T (\tilde{\tau} + \alpha w^i) \in K(p + \epsilon \hat{\theta})$ for some $\alpha > 0$. Hence, $(\bar{B}^T w^i)_m \leq 0$, so that

\[
\left( \bar{B}^T (\tilde{\tau} + \alpha w^i) \right)_m \leq (u - T E p)_m .
\]

If $m \in T_i$, we draw the same conclusion. Finally, consider the case that $m \in A_{\bar{\tau}}$. For this case we derive

\[
\left( \bar{B}^T (\tilde{\tau} + \frac{\epsilon T}{2 \rho(0, d)} w^i) \right)_m \leq \left( \tilde{u} - T E p + \frac{\epsilon T}{2 \rho(0, d)} \bar{B}^T w^i \right)_m
\]

\[
= \left( u - T E p + \frac{\epsilon T}{2 \rho(0, d)} \bar{B}^T w^i \right)_m - \frac{\epsilon T}{\rho(0, d)} + \frac{\epsilon T}{\rho(0, d)}
\]

\[
\leq (u - T E p)_m - \frac{\epsilon T}{\rho(0, d)} + \frac{\epsilon T}{\rho(0, d)}
\]

\[
= (u - T E p)_m .
\]

Putting all cases together, we conclude that

\[
\tilde{\tau} + \frac{\epsilon T}{2 \rho(0, d)} w^i \in K(p + \epsilon \hat{\theta})
\]

and (4.162) follows as $\tilde{\tau}$ and $i$ were chosen arbitrarily.

Next, we consider the case that $\text{dim}(p) = 0$. Then,

\[
K(p) = K^{-}(p) = \{ \tilde{\tau} \} \in \mathbb{Z}^N .
\]

In particular, the arguments used before do not apply because $K(p)$ does not have an interior. To circumvent this problem we follow another line of proof. To this end, construct a spanning tree $\hat{T}$ of $G$ by taking $T_u$ and $T_i$ and turning it into a spanning tree by adding some arcs from $A_u$ and $A_i$. This is possible because the arcs in $A_u \cup A_i \cup T_u \cup T_i$ connect all vertices of $G$ as a consequence of Proposition 4.4.

Now, let $m \in \hat{T} \cap (A_u \cup T_u)$ and let $\tau \in \mathbb{R}^N$ satisfy

\[
(\bar{B}^T \tau)_k = \begin{cases} (u - T E p)_k - \frac{\epsilon T}{\rho(0, d)} & \text{if } k = m \\ (\bar{B}^T \tilde{\tau})_k & \text{if } k \in \hat{T} \setminus \{m\} \end{cases}
\]
where \( \hat{B} \) is the \( N \times N \) matrix of full rank that consists of the \( N \) columns of \( \hat{B} \) that correspond to the arcs in \( \hat{T} \). Note that \( \tau \) can be constructed from \( \hat{\tau} \) by partitioning the vertices of \( G \) according to the \( m \)-th fundamental cut of \( G \) relative to \( \hat{T} \) and, then, leaving all \( \tau_i \) on the side of vertex \( \tau_0 \) as they are and incrementing or decrementing (depending on the direction of the \( m \)-th arc) all \( \tau \), 'on the other side' by \( \epsilon T / \rho(0, d) \). Thus, we can write

\[
\tau = \hat{\tau} + \frac{\epsilon T}{\rho(0, d)} g_m ,
\]  

(4.180)

where \( g_m \in \{0, 1\}^N \) or \( g_m \in \{0, -1\}^N \), and we have

\[
\hat{B}^T g_m = \pm \epsilon_m ,
\]  

(4.181)

where \( \epsilon_m \) is the \( m \)-th unit vector in \( \mathbb{R}^N \). By inspection of all arcs of \( G \), it is not difficult to verify that \( \tau \in K(p + \epsilon \hat{\theta}) \). To see this, note that, if the \( k \)-th arc of \( G \) has both endpoints on one side of the \( m \)-th fundamental cut, then

\[
(\hat{B}^T \tau)_k = (\hat{B}^T \hat{\tau})_k ,
\]  

(4.182)

so that

\[
(l - TEp - \epsilon T E \hat{\theta})_k \leq (\hat{B}^T \tau)_k \leq (u - TEp - \epsilon T E \hat{\theta})_k .
\]  

(4.183)

For arc \( m \) itself (4.183) follows directly from (4.179). Finally, if \( k \neq m \) and the \( k \)-th arc of \( G \) has an endpoint on both sides of the \( m \)-th fundamental cut, then \( k \in A_u \cup A_i \cup A_\neq \) and

\[
| (\hat{B}(\tau - \hat{\tau}))_k | \leq \frac{\epsilon T}{\rho(0, d)} ,
\]  

(4.184)

so that

\[
(l - TEp)_k - \frac{\epsilon T}{\rho(0, d)} \leq (\hat{B}^T \tau)_k \leq (u - TEp)_k + \frac{\epsilon T}{\rho(0, d)} ,
\]  

(4.185)

which implies (4.183) as well. If \( m \in \hat{T} \cap (A_i \cup A_\neq) \) we construct \( \tau \in \mathbb{R}^N \) from

\[
(\hat{B}^T \tau)_k = \left\{ \begin{array}{ll}
(l - TEp)_k + \frac{\epsilon T}{\rho(0, d)} & \text{if } k = m \\
(\hat{B}^T \hat{\tau})_k & \text{if } k \in \hat{T} \setminus \{m\}
\end{array} \right.
\]  

(4.186)

and make the same conclusions. As a result we find

\[
\{ \hat{\tau} + \sum_{m=1}^N \alpha_m g_m \mid 0 \leq \alpha_m \leq \frac{\epsilon T}{N \rho(0, d)} , \quad m = 1, \ldots, N \} \subseteq K(p + \epsilon \hat{\theta}) ,
\]  

(4.187)

because \( K(p + \epsilon \hat{\theta}) \) is convex. Moreover, since \( \det(\hat{B}) = 1 \), we have

\[
\text{vol}(\{ \sum_{m=1}^N \alpha_m g_m \mid 0 \leq \alpha_m \leq \frac{\epsilon T}{N \rho(0, d)} , \quad m = 1, \ldots, N \})
\]  

(4.188)
\[
\begin{align*}
= \operatorname{vol}(\{ \sum_{m=1}^{N} \alpha_m \hat{B}^T g_m \mid 0 \leq \alpha_m \leq \frac{\epsilon T}{N \rho(0, d)}, \ m = 1, \ldots, N \}) & \quad \text{(4.189)} \\
= \operatorname{vol}(\{ \sum_{m=1}^{N} \alpha_m e_m \mid 0 \leq \alpha_m \leq \frac{\epsilon T}{N \rho(0, d)}, \ m = 1, \ldots, N \}) & \quad \text{(4.190)} \\
= \left(\frac{\epsilon l}{N \rho(0, d)}\right)^N, & \quad \text{(4.191)}
\end{align*}
\]
so that the lemma follows as well if \( \dim(p) = 0. \)

**Lemma 4.6** Let \( p \in \hat{\Lambda}(\mathbb{H}) \) with \( \dim(p) < N. \) Also, let \( \theta \in \Theta_p \) and define
\[
\theta_{\max} = \max\{ \mid \theta_i \mid : i = 1, \ldots, M - N \},
\]
(4.192)

Set
\[
\delta = \frac{1}{2 T \theta_{\max}}.
\]
(4.193)

Then, for all \( 0 \leq \epsilon \leq \delta \) we have
\[
v_N(p + \epsilon \theta) \leq v_{\dim(p)}(p) \left(3 \epsilon T \theta_{\max} N^3\right)^{N - \dim(p)} + o(\epsilon^{N - \dim(p)}).
\]
(4.194)

**Proof** First of all, note that, since \( \rho(0, \theta) = 1 \), we have
\[
0 < a \leq \theta_{\max} \leq b
\]
(4.195)

for numbers \( a \) and \( b \) that are constant on \( \Theta_p. \)

Let \( \{ w^1, \ldots, w^{N - \dim(p)} \} \in W_p^{\perp} \) be an orthonormal system of \( N - \dim(p) \) vectors. Then, if \( \tau \in K(p + \epsilon \theta) \), we can write
\[
\tau = \hat{\tau} + \sum_{i=1}^{N - \dim(p)} \alpha_i w^i
\]
(4.196)

for unique \( \hat{\tau} \in W_p \) and unique numbers \( \alpha_1, \ldots, \alpha_{N - \dim(p)}. \) Also, define the sets
\[
K^+(p) = \{ \tau \in W_p \mid (l - TE p)_m - \epsilon T \theta_{\max} (1 + 6N^4) \leq \left(\hat{B}^T \tau\right)_m \leq (u - TE p)_m + \epsilon T \theta_{\max} (1 + 6N^4), \ m \in \mathbb{A}_p \}
\]
(4.197)

and
\[
\hat{K}(p) = \{ \tau + \sum_{i=1}^{N - \dim(p)} \alpha_i w^i \mid \tau \in K^+(p) \text{ and } |\alpha_i| \leq 3 \epsilon T \theta_{\max} N^3, \ i = 1, \ldots, N - \dim(p) \}.
\]
(4.198)

We will show that
\[
K(p + \epsilon \theta) \subseteq \hat{K}(p).
\]
(4.199)
If so, then
\[
\begin{align*}
v_N(p + \epsilon \theta) & \leq \text{vol}(\hat{K}(p)) \\
& = \text{vol}(K^+(p)) \left(3\epsilon T \theta_{\max} N^2\right)^{N - \dim(p)} \\
& \leq \left(\nu_{\text{dim}(p)}(p) + \epsilon D \text{vol}(\partial K(p)) + o(\epsilon)\right) \left(3\epsilon T \theta_{\max} N^3\right)^{N - \dim(p)}
\end{align*}
\]

for some number $D \geq 0$ that does not depend on $\theta$ and $\epsilon$, which settles the lemma. Note that $D$ does not depend on $\theta$ because $\rho(0, \theta) = 1$ and because the vector products of the normals of the hyperplanes that define $K(p)$ and $K^+(p)$ are restricted to $\{-2, -1, 0, 1, 2\}$.

To show (4.199), let $\tau \in K(p + \epsilon \theta)$ and write $\tau$ as in (4.196). We will first show that
\[
|\alpha_\tau| \leq 3\epsilon T \theta_{\max} N^3
\]
for all $i = 1, \ldots, N - \dim(p)$. Thereafter, we will show that (4.196) and (4.203) imply that $\hat{\tau} \in K^+(p)$, thereby yielding (4.199).

Let $m \in A_u \cup T_u$. Then,
\[
(\hat{B}^T \hat{\tau})_m = (u - T E p)_m,
\]
so that the corresponding $m$-th interval constraint for $\tau$ implies
\[
\left(\sum_{i=1}^{N - \dim(p)} \alpha_i \hat{B}^T w^i\right)_m \leq (-\epsilon T E \theta)_m.
\]

Similarly, if $m \in A_i \cup T_i$, we find
\[
(-\epsilon T E \theta)_m \leq \left(\sum_{i=1}^{N - \dim(p)} \alpha_i \hat{B}^T w^i\right)_m.
\]

For $\epsilon = 0$ these inequalities define $W_p$. Thus, $\epsilon = 0$ implies $\alpha_i = 0$ for all $i = 1, \ldots, N - \dim(p)$. Furthermore, if $W$ is the $N \times (N - \dim(p))$ matrix $W$, the columns of which are the vectors $w^i$ and, if $A_u \cup A_i \cup T_u \cup T_i$ has $h$ elements, we can write (4.205) and (4.206) as
\[
\hat{B}^T W \alpha \leq q,
\]
where $\hat{B}$ is the $N \times h$ matrix that can be constructed by taking the $h$ columns from $\hat{B}$ that correspond to $A_u \cup A_i \cup T_u \cup T_i$ and multiply a column with $-1$ if it corresponds to an arc in $A_i \cup T_i$. Furthermore, in (4.207) $q \in \mathbb{R}^h$ is defined by
\[
q_m = \begin{cases} 
-\epsilon T \theta_m & \text{if } m \in A_u \\
\epsilon T \theta_m & \text{if } m \in A_i \\
0 & \text{otherwise},
\end{cases}
\]

for all $m \in A_u \cup A_i \cup T_u \cup T_i$. For $i = 1, \ldots, N - \dim(p)$ we derive
\[
\max \{ \alpha_i \mid \hat{B}^T W \alpha \leq q \} = \min \{ q^T y \mid W^T \hat{B} y = e_i, \ y \geq 0 \} = \min \{ q^T y \mid \hat{B} y = w^i, \ y \geq 0 \} \leq c T \theta_{\max} \min \{ e^T y \mid \hat{B} y = w^i, \ y \geq 0 \} \leq 3\epsilon T \theta_{\max} N^3.
\]
and, likewise,
\[
\min\{ \alpha_i \mid \hat{B}^T W \alpha \leq q \} \geq -\epsilon T \theta_{\text{max}} \min\{ \epsilon^T y \mid \hat{B} y = -w_i, \ y \geq 0 \} \label{eq:4.213}
\]
\[
\geq -3 \epsilon T \theta_{\text{max}} N^3 \label{eq:4.214}.
\]

Below we will show that the inequalities in (4.212) and (4.214) are valid.

Consider the graph with vertex set \{0, \ldots, N\} and set of arcs \(A_u \cup A_i \cup T_u \cup T_i\). That is, take \(\hat{G}\) and delete all arcs in \(A_u\). If \(m \in A_i \cup T_i\), then reverse the direction of arc \(m\) and its interval (i.e. multiply (4.206) by \(-1\)). Call the resulting graph \(\hat{G}\) and make the following sequence of observations.

1. By Proposition 4.4, \(\hat{G}\) can be constructed by considering all elementary circuits \(c\) of \(G\) for which \(c^T p = U(c)\) and adding the arcs of \(c\) to \(\hat{G}\). However, if \(m \in A_i \cup T_i\), then the direction of arc \(m\) (and its corresponding interval) should be added in reversed order.

2. In \(\hat{G}\), all arcs in each of these circuits are oriented such that they can be traversed by following the arc directions, i.e. the circuits are directed. This is a direct consequence from the construction of \(\hat{G}\).

3. \(\hat{G}\) has vertex incidence matrix \(\hat{B}\). Like with \(\hat{B}\) the row for vertex 0 is omitted.

4. Let \(b^j\) be the \(j\)-th column of \(\hat{B}\). Then, there are at most \(N\) other columns of \(\hat{B}\) such that the sum of these columns equals \(-b^j\). This follows immediately from the first two observations.

5. Each vector in \(W_p \perp\) is a linear combination of the columns of \(\hat{B}\).

6. Each component of \(\hat{G}\) has a spanning tree of which the corresponding columns of \(\hat{B}\) span the subspace of \(W_p \perp\) that is generated by the columns of \(\hat{B}\) that correspond to the particular graph component. Let \(\hat{B}_*\) be the submatrix of \(\hat{B}\) consisting of the columns of \(\hat{B}\) that correspond to these spanning trees. Also, let \(A\) be the square submatrix of \(\hat{B}_*\) that arises when the rows of \(\hat{B}_*\) that correspond to isolated vertices of \(\hat{G}\) are deleted. Note that \(A\) is the totally unimodular vertex-arc incidence matrix of a tree.

Consider the system
\[
\hat{B} y = w_i \label{eq:4.215}.
\]

By observation (6), a solution \(\hat{y}\) to this system can be constructed by solving
\[
\hat{B}_* y = w_i \label{eq:4.216}.
\]

Then, since \(\|w_i\|_2 = 1\) and \(A^{-1}\) is a matrix with entries in \(\{0, \pm 1\}\),
\[
|\hat{y}_j| \leq N \label{eq:4.217}.
\]
for all entries $j$ of $\hat{y}$ that correspond to the columns of $\hat{B}_*$, and, hence,
\[
e^T\hat{y} \leq N^2 . \tag{4.218}
\]

Now, some of the $\hat{y}_j$ may be negative. For such index $j$, use observation (4) to increase at most $N$ other entries of $\hat{y}$ by $-\hat{y}_j$ and reset $\hat{y}_j = 0$. This way, $e^T\hat{y}$ increases by at most $-\hat{y}_j(N + 1) \leq N^2 + N$. Note that no entry of $\hat{y}$ is decreased by this operation. Thus, we have to do this for at most $N$ entries of $\hat{y}$, so that, finally, $\hat{y} \geq 0$, $\hat{B}\hat{y} = w'$, and
\[
e^T\hat{y} \leq N^2 + N(N^2 + N) \tag{4.219}
= 3N^3 . \tag{4.220}
\]

The inequality in (4.214) follows the same way.

Next, we will show that, under (4.203), we have $\hat{\tau} \in K^+(p)$, so that $\tau \in \tilde{K}(p)$, as was to be shown. By virtue of the foregoing, if (4.203) is true, then
\[
(l - TEp - cTE\theta)_m \leq (\hat{B}^T\tau)_m \leq (u - TEp - cTE\theta)_m , \tag{4.221}
\]
for all $m \notin \mathcal{A}_\phi$. If, on the other hand, $m \in \mathcal{A}_\phi$, then
\[
(\hat{B}^T\hat{r})_m \leq \left( u - TEp - cTE\theta - \sum_{i=1}^{N - \dim(p)} \alpha_i \hat{B}^Tw^i \right)_m \tag{4.222}
\leq (u - TEPp)_m + cT\theta_{\max}(1 + 6N^4) , \tag{4.223}
\]
and, similarly,
\[
(\hat{B}^T\hat{r})_m \geq (l - TEp)_m - cT\theta_{\max}(1 + 6N^4) , \tag{4.224}
\]
so that $\hat{\tau} \in K^+(p)$.

**4.3 The heuristic**

The obvious approach to pick a phase shift in compliance with the robustness distribution is to enumerate all phase shifts, compute $\mathcal{R}$, and sample straightforwardly. Unfortunately, enumerating all phase shifts is computationally intractable and, hence, we will turn to a heuristic approach.

The heuristic to sample a phase shift from $\mathcal{R}$ starts with the random generation of a vector $x$ from the volume distribution. In the second step $x$ is rounded to a phase shift in $\hat{\Lambda}(\mathbb{Z})$ using distance measure $\rho$. Thus, we actually sample from the distribution $\mathcal{R}^\infty$, which we define below, and use it to approximate $\mathcal{R}$.

**Definition 4.5** By $r$ we denote the mapping $r : \hat{\Lambda}(\mathbb{R}) \rightarrow \hat{\Lambda}(\mathbb{Z})$ that rounds a vector in $\hat{\Lambda}(\mathbb{R})$ to the nearest phase shift in $\hat{\Lambda}(\mathbb{Z})$ using distance measure $\rho$. That is,
\[
r(x) = \arg\min\{ \rho(x, p) \mid p \in \hat{\Lambda}(\mathbb{Z}) \} , \tag{4.225}
\]
for all $x \in \hat{\Lambda}(\mathbb{R})$. □
Note that, by definition of $\rho$,
\[
    r(x) = \arg\min \{ \| QE(x - p) \|_2 \mid p \in \tilde{\Lambda}(\mathbb{Z}) \},
\]
for all $x \in \tilde{\Lambda}(\mathbb{R})$.

**Definition 4.6** The probability distribution $\mathcal{R}^\infty$ on $\tilde{\Lambda}(\mathbb{Z})$ is defined by
\[
    \mathcal{R}^\infty(p) = \mathcal{V}(r^{-1}(p))
\]
for all $p \in \tilde{\Lambda}(\mathbb{Z})$, where
\[
    r^{-1}(p) = \{ y \in \tilde{\Lambda}(\mathbb{R}) \mid \rho(p, y) \leq \rho(\hat{p}, y) \text{ for all } \hat{p} \in \tilde{\Lambda}(\mathbb{Z}) \}.
\]

The sets $r^{-1}(p)$ in the definition of $\mathcal{R}^\infty$ have nonzero measure, witness the following proposition.

**Proposition 4.9** For all $p \in \tilde{\Lambda}(\mathbb{Z})$, $r^{-1}(p)$ is a full dimensional polytope contained in $\tilde{\Lambda}(\mathbb{R})$. Furthermore,
\[
    \tilde{\Lambda}(\mathbb{R}) = \bigcup_{p \in \tilde{\Lambda}(\mathbb{Z})} r^{-1}(p),
\]
and if $p, \hat{p} \in \tilde{\Lambda}(\mathbb{Z})$, $p \neq \hat{p}$, then $r^{-1}(p)$ and $r^{-1}(\hat{p})$ have at most a facet in common.

**Proof** Let $x \in \tilde{\Lambda}(\mathbb{R})$, $p = r^{-1}(x)$, and $A = E^TQ^TQE$, where $Q$ is defined by (4.73). Also, let $\hat{p} \in \tilde{\Lambda}(\mathbb{Z}) \setminus \{p\}$. Then, $\rho(x, p) \leq \rho(x, \hat{p})$, which is equivalent to
\[
    2(\hat{p} - p)^T Ax \leq \hat{p}^T A\hat{p} - p^T Ap.
\]
Therefore,
\[
    r^{-1}(p) = \{ x \in \tilde{\Lambda}(\mathbb{R}) \mid 2(\hat{p} - p)^T Ax \leq \hat{p}^T A\hat{p} - p^T Ap \text{ for all } \hat{p} \in \tilde{\Lambda}(\mathbb{Z}) \setminus \{p\} \}.
\]
To see that $r^{-1}(p)$ is of full dimension in $\mathbb{R}$, note that $p \in r^{-1}(p)$. Then, taking $x = p$ in (4.230) gives
\[
    0 \leq (p - \hat{p})^T A(p - \hat{p}),
\]
the right-hand side of which equals $\rho^2(p, \hat{p})$. Since $\rho^2(p, \hat{p}) > 0$, the inequality in (4.232) holds strictly. Therefore, $p$ is bounded away from the boundary of $r^{-1}(p)$.

Although the above concludes the proof, we will show below that $r^{-1}(p)$ is of full dimension by giving another characterization of $r^{-1}(p)$ that gives more insight in the structure of $r^{-1}(p)$. To this end, let $\theta \in \Theta_p$, where $\Theta_p$ is defined as in Section 4.2.5. Let $\delta \geq 0$ and take $x = p + \delta\theta$ in (4.230). This gives
\[
    2\delta(\hat{p} - p)^T A\theta \leq (\hat{p} - p)^T A(\hat{p} - p) = \rho^2(\hat{p}, p).
\]
Hence,

\[ r^{-1}(p) = \{ p + \delta \theta \in \hat{\Lambda}(\mathbb{R}) \mid \theta \in \Theta_p, \delta \geq 0, \text{ and} \]
\[ 2\delta(\hat{p} - p)^T A\theta \leq \rho^2(\hat{p}, p) \text{ for all } \hat{p} \in \hat{\Lambda}(\mathbb{Z}) \setminus \{p\} \}. \tag{4.235} \]

Furthermore, we derive that for all \( \theta \in \Theta_p \)

\[ (\hat{p} - p)^T A\theta = (\hat{p} - p)^T E^T Q^T Q E \theta \]
\[ \leq ||Q E(\hat{p} - p)||_2 ||Q E\theta||_2 \]
\[ = \rho(p, \hat{p}) \rho(0, \theta) \]
\[ = \rho(p, \hat{p}), \tag{4.238} \]

so that

\[ \{ p + \delta \theta \in \hat{\Lambda}(\mathbb{R}) \mid \theta \in \Theta_p \text{ and } 0 \leq \delta \leq \frac{1}{2} \min_{\hat{p} \in \hat{\Lambda}(\mathbb{Z}) \setminus \{p\}} \rho(p, \hat{p}) \} \subseteq r^{-1}(p), \tag{4.240} \]

implying that \( r^{-1}(p) \) is of dimension \( M - N \).

In the previous section we implicitly motivated this approach. We showed that, if \( p, \hat{p} \in \hat{\Lambda}(\mathbb{Z}) \) and \( \dim(p) > \dim(\hat{p}) \), then \( \phi_p(\epsilon) \) is at least an order of a magnitude larger than \( \phi_{\hat{p}}(\epsilon) \) for small \( \epsilon \). The bias we introduce by replacing \( \mathcal{R} \) with \( \mathcal{R}^\approx \) is determined by the way \( r^{-1} \) cuts \( \hat{\Lambda}(\mathbb{R}) \) into pieces. In Section 4.1 we will see that the bias can be undesirably large in theory. As a contrast, we also show that for the example presented in Chapter 3 the approximation is quite reasonable.

In the remainder of this section we will explain the heuristic in detail. That is, we will first show that sampling from \( \mathcal{V} \) is easily done by means of a transformation technique and thereafter we say something more about the rounding mapping \( r \).

Sampling from \( \mathcal{V} \) is easy as it can be reduced to the uniform generation of a vector in

\[ Y = \{ (\tau, x) \in \mathbb{R}^N \times \mathbb{R}^{M-N} \mid l \leq \hat{B}^T \tau + TE \leq u, \ \tau \in \mathbb{R}^N, \ x \in \mathbb{R}^{M-N} \}, \tag{4.211} \]

which we defined earlier in Section 4.2.1 and can be written as

\[ Y = \{ (\tau, x) \in \mathbb{R}^N \times \mathbb{R}^{M-N} \mid x \in \hat{\Lambda}(\mathbb{R}), \ \tau \in \hat{K}(x) \}. \tag{4.242} \]

**Lemma 4.7** Let \( (\tau, x) \) be uniformly distributed in \( Y \). Then, \( x \) is distributed in \( \hat{\Lambda}(\mathbb{R}) \) according to \( \mathcal{V} \).

**Proof** Let \( A \subseteq \hat{\Lambda}(\mathbb{R}) \) be measurable and let \( D \subseteq Y \) be such that \( A \) is its projection on \( \hat{\Lambda}(\mathbb{R}) \), i.e.

\[ D = \{ (\tau, x) \in \mathbb{R}^M \mid x \in A, \ \tau \in \hat{K}(x) \}. \tag{4.243} \]
Then

\[
P r(x \in A) = Pr((\tau, z) \in D) = \frac{\int_D d(\tau, y)}{\int_Y d(\tau, y)} = \frac{\int_A \int_{K(y)} d\tau \, dy}{\int_{\tilde{\Lambda}(\mathbb{R})} \int_{K(y)} d\tau \, dy} = \frac{\int_A v_N(y) \, dy}{\int_{\tilde{\Lambda}(\mathbb{R})} v_N(y) \, dy} = V(A).
\]

Uniform sampling from \( Y \) can be done by means of the transformation technique, see e.g. [114, 131]. The idea of this technique is very simple. Firstly, sample a vector \( z \) from the hypercube

\[
H = \{ z \in \mathbb{R}^M \mid l \leq z \leq u \}.
\]  

(4.244)

Secondly, transform this vector using the matrix \( J^{-1} \), see (4.36). The vector consisting of the last \( M - N \) elements of \( J^{-1}z \) is then distributed in \( \tilde{\Lambda}(\mathbb{R}) \) according to \( \mathcal{V} \). Formally, that is

**Proposition 4.10** Let \( z \) be distributed uniformly on the hypercube \( H \). Then,

\[
z = \frac{1}{T} F \tilde{z}
\]  

(4.215)

follows the volume distribution on \( \tilde{\Lambda}(\mathbb{R}) \).

**Proof** Since transformation \( J^{-1} \) preserves uniformity, \( J^{-1}z \) is uniformly distributed on \( Y \) if \( z \) is uniformly distributed on \( H \). Furthermore,

\[
J^{-1} = \begin{pmatrix} 0 & (\tilde{B}_2^{-1})^T \\ \frac{1}{T} I_{M-N} & -\frac{1}{T}(\tilde{B}_2^{-1} \tilde{B}_1)^T \end{pmatrix} = \begin{pmatrix} 0 & (\tilde{B}_2^{-1})^T \\ \frac{1}{T} F \end{pmatrix}.
\]  

(4.246)

Hence, by virtue of Lemma 4.7, \( \frac{1}{T} F \tilde{z} \) follows \( \mathcal{V} \). \( \square \)

In the rounding step we solve the program

\[
QP: \quad \min \{ \rho(x, p) \mid p \in \tilde{\Lambda}(\mathbb{Z}) \}
\]

(4.247)

to evaluate \( r(x) \). In words, the rounding means that we map the system

\[
l - Tx \leq \tilde{B}^T \tau \leq u - Tx
\]

(4.248)
onto

\[ l - T p \leq \hat{B}^\top \tau \leq u - T p , \]  

(4.249)

such that \( p \in \hat{\Lambda}(\mathbb{Z}) \) and \( \rho(p, x) \) is minimal. As the rows of \( \hat{B} \) span the cut space of \( \mathcal{G} \), we see that translating \( x \) (or \( p \)) over cuts of \( \mathcal{G} \) has no effect on the shape of \( K(p) \) (or \( K(\tilde{p}) \)). Moreover, using \( \rho \) preserves the shape of \( K(x) \) as much as possible in the sense that the right and left-hand sides of (4.248) and (4.249) are as close as possible.

Since sampling from the volume distribution, i.e. the first step in the heuristic, is easy, solving \( QP \) forms the hard part of the heuristic. Considering that searching for a phase shift in \( \Lambda(\mathbb{Z}) \) is a very hard problem, let alone optimizing over \( \hat{\Lambda}(\mathbb{Z}) \), the applicability of the heuristic is determined by the extent to which we are able to handle \( QP \). For the special case that \( M - N = 1 \) we have

**Proposition 4.11** If \( \hat{\Lambda}(\mathbb{R}) \subset \mathbb{R} \), then \( r \) corresponds to rounding to the nearest integer.

**Proof 1** Using the definition of \( r \) and \( \rho \) we find for \( x \in \hat{\Lambda}(\mathbb{R}) \) that

\[ r(x) = \text{argmin}\{ \rho(p, x) \mid p \in \hat{\Lambda}(\mathbb{Z}) \} \]

(4.250)

\[ = \text{argmin}\{ ((p - x)^\top E^\top Q^\top QE(p - x))^{1/2} \mid p \in \hat{\Lambda}(\mathbb{Z}) \} \]

(4.251)

\[ = \text{argmin}\{ |p - x| \mid p \in \hat{\Lambda}(\mathbb{Z}) \} , \]

(4.252)

because \( E \in M \times 1 \) and \( p - x \in \mathbb{R} \).

**proof 2** Note that for \( x = \frac{1}{2}(p + \tilde{p}) \) the left-hand side and right-hand side in (4.230) are equal. \( \Box \)

### 4.4 Examples

To illustrate the concepts discussed in this chapter we study three examples, the first two of which are artificial. The two artificial examples are small to keep things simple and intuitively clear and are constructed in such a way that they possess much symmetry, which makes it easy to analyze them analytically to a large extent. In particular, analysis of \( r \) and \( \mathcal{R}^\mathbb{R} \) is of interest because they embody the heuristic aspects of the sampling approach. It turns out that in the first example \( \mathcal{R} \) has a nice monotonicity property implying that we can live with \( \mathcal{R}^\mathbb{R} \) as a replacement of \( \mathcal{R} \). In the second example, on the other hand, \( \mathcal{R} \) does not have such a favorable property. The third example is a real-life example taken from Chapter 3.

#### 4.4.1 First artificial example

The first example is that of a timetable structure that has a minimal number of constraints given its \( N \) vertices and given that having less than \( N \) arcs allows for a decomposition of \( \mathcal{G} \) and, hence, of the timetable structure. Its graphical representation is shown in Figure 4.1. For notational convenience we number the vertices 1 through \( N \). We set all associated
intervals to $[-N, N]$ and the period $T$ to $2N + 1$. For spanning tree $T$ we may take any set of $N - 1$ arcs, say all arcs except the first one pointing from vertex $N$ to vertex 1. Since this is the only arc not in the tree, we find that \( \hat{\Lambda}(\mathbb{R}) \subseteq \mathbb{R} \), so that Proposition 4.11 applies.

It follows easily that

\[
\hat{\Lambda}(\mathbb{R}) = \{ x \in \mathbb{R} \mid -\alpha \leq x \leq \alpha \}
\]  

and

\[
\hat{\Lambda}(\mathbb{Z}) = \{ -\lfloor \alpha \rfloor, \ldots, \lfloor \alpha \rfloor \},
\]

where

\[
\alpha = \frac{N^2}{2N + 1}.
\]

By Corollary 4.3, we know that the the maximum of the volume function $v_N$ is attained at $p = 0 \in \hat{\Lambda}(\mathbb{Z})$. As an illustration, Figure 4.2 shows $v_N$ for $N = 3$. The thick vertical lines correspond to phase shifts in $\hat{\Lambda}(\mathbb{Z})$. Using Proposition 4.11, the thin vertical lines at $-\frac{1}{2}$ and $\frac{1}{2}$ and the domain boundaries $-\alpha$ and $\alpha$ indicate how $r$ maps onto $\hat{\Lambda}(\mathbb{Z})$.

Since for $N = 3$, $\eta = 2$, $\mathcal{R}$ is given by (4.111). Recall that in this example we count the vertices $1, \ldots, N$ and not $0, \ldots, N$ like in all other parts of this chapter. Thus, we find

\[
\mathcal{R}(0) = \frac{v_2(0)}{v_2(-1) + v_2(0) + v_2(1)}
\]
Figure 4.2: The volume function $v_N$ for $N = 3$. 
Figure 4.3: Graphical representation of the timetable structure of the second example.

\[ R(-1) = R(1) = \frac{v_2(1)}{v_2(-1) + v_2(0) + v_2(1)} \]  \hspace{1cm} (4.257)

Then, \( v_2(0) > v_2(-1) = v_2(1) \) gives us that \( R(0) > R(-1) = R(1) \). Furthermore, the area under the graph restricted to \( [\frac{1}{2}, \alpha] \) is smaller than the corresponding area for \( [-\frac{1}{2}, \frac{1}{2}] \), so that also \( R^\infty(0) > R^\infty(-1) = R^\infty(1) \) follows. Hence, the probability mass that \( R^\infty \) assigns to a phase shift \( p \in \Lambda(Z) \) is monotone in \( R \). In the absence of a tractably computable exact alternative for \( R \), this property is important because it guarantees that \( R^\infty \) will not behave in some undesirable manner. If \( R^\infty \) has this property we say that \( R^\infty \) is monotone in \( R \).

### 4.4.2 Second artificial example

The second example is shown in Figure 4.3. Its graph \( G \) consists of \( N + 1 \geq 4 \) vertices (numbered 0, 1, \ldots, \( N \)), \( M = 2N \) arcs, and the period \( T \) equals \( 2N - 2 \). Note that in this example \( M - N = N \). All arcs emanating from positively numbered vertices, which we will call the circular arcs, have interval \([0, 2] \) and are numbered \( 1, \ldots, N \). The remaining arcs (emanating from vertex 0), which we will refer to as the radial arcs, have interval \([0, 2N - 3] \) and are numbered \( N + 1, \ldots, 2N \). We choose the set of radial arcs as the spanning tree \( T \).

Clearly, an elementary circuit can only be constructed by selecting a consecutive range of circular arcs and complementing it with two radial arcs. One particular set of elementary...
circuit is the set of fundamental circuits. For this set of circuits we have for all \( x \in \Lambda(\mathbb{R}) \) that
\[
- \frac{2N - 3}{2N - 2} \leq x_m \leq \frac{2N - 1}{2N - 2},
\]
for all \( m = 1, \ldots, N \). This implies that \( p \) is binary for all \( p \in \Lambda(\mathbb{Z}) \). Moreover, by also considering the elementary circuits involving all circular arcs, we see that
\[
0 \leq c^T x \leq \frac{2N}{2N - 2},
\]
\[
= 1 + \frac{1}{N - 1},
\]
for all \( x \in \Lambda(\mathbb{R}) \), yielding
\[
\Lambda(\mathbb{Z}) = \{0\} \cup \{c_i \in \mathbb{R}^N \mid i = 1, \ldots, N\}.
\]
Note that, if \( x \in \Lambda(\mathbb{R}) \), then also \( P x \in \Lambda(\mathbb{R}) \), where \( P \) is a permutation matrix that permutes the \( N \) entries of \( x \) cyclically. That is \( (P x)_i = x_{i-1} \) for \( i = 2, \ldots, N \) and \( (P x)_1 = x_N \), for all \( x \in \mathbb{R}^N \). We refer to this property as permutation invariance and \( \Lambda(\mathbb{Z}) \) inherits it from \( \Lambda(\mathbb{R}) \). Moreover, \( v_N \) is also invariant to such cyclic permutations.

From (4.259) we see that the phase shift \( p = 0 \) is on the boundary of \( \Lambda(\mathbb{R}) \), so that \( v_N(0) = 0 \). Furthermore, by examining all elementary circuits, except the one that was used in (4.259), we find
\[
- \frac{2N - 3}{2N - 2} \leq x_1 + \ldots, + x_k \leq \frac{2N - 3 + 2k}{2N - 2}
\]
for all \( k \leq N - 1 \), where the fraction on the left side is less than 0 and the fraction on the right side is larger than 1. Hence, all phase shifts in \( \Lambda(\mathbb{Z}) \setminus \{0\} \) are in the interior of \( \Lambda(\mathbb{R}) \) and, consequently, \( v_N(p) > 0 \) for all these phase shifts. Using permutation invariance and following the definition of \( \mathcal{R} \), we conclude that \( \mathcal{R}(0) = 0 \) and \( \mathcal{R}(p) = 1/N \) for all \( p \in \Lambda(\mathbb{Z}) \setminus \{0\} \). Moreover, permutation invariance also implies that \( \mathcal{R}_\infty(p) = C \) for all \( p \in \Lambda(\mathbb{Z}) \setminus \{0\} \) for some constant \( C \geq 0 \). Therefore, \( \mathcal{R}_\infty \) is monotone in \( \mathcal{R} \) if and only if \( \mathcal{R}_\infty(0) \leq 1/(N + 1) \). Unfortunately, we will observe that this is not the case, at least not if \( 5 \leq N \leq 100 \).

Figure 4.4 shows, for \( N \) ranging from 3 to 100, the numbers \( d_N \) on the vertical axis, which are defined as follows. For each dot in the graph, \( 400 \cdot (N + 1) \) vectors in \( \Lambda(\mathbb{R}) \) were sampled from the volume distribution \( \mathcal{V} \) and rounded according to the heuristic. Hence, it is hard to deny that \( \mathcal{R}_\infty(0) \leq 1/(N + 1) \) if no more than about 400 samples are mapped by \( r \) onto 0. In the experiment, however, \( 400d_N \) points ended up in 0. Thus, \( d_N \) can be interpreted as a measure of deviation from the desired monotonicity of \( \mathcal{R}_\infty \). That is, \( d_N \leq 1 \) is an indication of monotonicity, and the smaller smaller \( d_N \), the better \( \mathcal{R}_\infty \) approximates \( \mathcal{R} \).
We can give the following informal explanation for the absence of monotonicity, which relies on three arguments: (a) \( v_N(0) = 0 \) and \( v_N(\epsilon_i) > 0 \) for all \( i = 1, \ldots, N \), (b) \( r \) maps the area in the vicinity of \( \ell \cap \hat{A}(R) \) onto 0, where \( \ell \) is the line

\[
\ell = \{ \alpha \epsilon \in R^N \mid \alpha \in R \},
\]

and (c) the volume distribution assigns a large proportion of the probability mass to this area. On the basis of (a)-(c), we may conclude that \( \mathcal{R}^N(0) \) is relatively large, whereas \( \mathcal{R}(0) = 0 \). Next, we will comment on point (b) and (c).

Concerning point (b) note that \( \ell \) is in the circuit space of \( \mathcal{G} \) so that \( \rho(0, x) \) is just the Euclidian distance between 0 and \( x \) for all \( x \in \ell \). Also, by inspection of the elementary circuits (4.259) and (4.263) of \( \mathcal{G} \) we obtain that \( \ell \cap \hat{A}(R) = \{ \alpha \epsilon \in R^M \mid \alpha \in [0, 2/T] \} \). Then, point (b) follows from the following lemma.

**Lemma 4.8** \( \rho(\epsilon_i, \delta \epsilon) > \rho(0, \delta \epsilon) \) for all \( 0 \leq \delta \leq \frac{\epsilon}{T}, N \geq 8, \) and \( i = 1, \ldots, N. \)

**Proof** Since \( \epsilon \) is in the circuit space of \( \mathcal{G} \) we find that

\[
\rho(0, \delta \epsilon) = \delta \sqrt{N},
\]

for all \( 0 \leq \delta \leq \frac{\epsilon}{T} \). Note that, by permutation invariance, it suffices to consider \( i = 1 \) only.

By definition we have

\[
\rho(\epsilon_1, \delta \epsilon) = \min\{ \| E(\epsilon_1 - \delta \epsilon) + S^T z \|_2 \mid z \in R^N \},
\]

for all \( 0 \leq \delta \leq \frac{\epsilon}{T} \).
where the columns of $S^T$ are the fundamental cuts of $\mathcal{G}$, i.e.

$$
S^T = \begin{pmatrix}
1 & 0 & 0 & \cdots & 0 & 0 & 0 & -1 \\
-1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & -1 & 1 \\
1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\
\end{pmatrix} \quad (4.267)
$$

Hence,

$$
\rho^2(\epsilon_1, \delta \epsilon) = \min \{ (z_1 - z_N - \delta + 1)^2 + \sum_{i=1}^{N-1} (z_{i+1} - z_i - \delta)^2 + z^T z \mid z \in \mathbb{R}^N \} \quad (4.268)
$$

$$
= N\delta^2 - 2\delta + 1 + \min \{ 2(z_1 - z_N) + \sum_{i=1}^{N} (z_{i+1} - z_i)^2 + z^T z \mid z \in \mathbb{R}^N \} \quad (4.269)
$$

$$
\geq N\delta^2 - 2\delta + 1 + \min \{ 2(z_1 - z_N) + (z_1 - z_N)^2 + z_1^2 + z_N^2 \mid z_1, z_N \in \mathbb{R} \} \quad (4.270)
$$

$$
= N\delta^2 - 2\delta + \frac{1}{3} \quad (4.271)
$$

$$
\geq N\delta^2 \quad (4.272)
$$

$$
= \rho^2(0, \delta \epsilon) \quad , (4.273)
$$

where $z_{N+1}$ is defined as $z_1$. Note that

$$
\frac{1}{3} - 2\delta \geq \frac{1}{3} - \frac{4}{T} \quad (4.274)
$$

$$
= \frac{N - 7}{3N - 3} \quad (4.275)
$$

$$
> 0 \quad (4.276)
$$

for $N \geq 8$.

As to point (c), we have the following lemma, which we will prove in two ways for the purpose of demonstration.
Lemma 4.9 \( v_N \) attains its maximum on \( \hat{\Lambda}(\mathbb{R}) \) at \( x_{\text{max}} = \epsilon/T \in \ell \).

**Proof 1** First, we show that \( x_{\text{max}} \in \ell \). To this end, let \( x \in \hat{\Lambda}(\mathbb{R}) \) and construct

\[
y = \frac{1}{N} \sum_{n=0}^{N-1} P^n x ,
\]

where \( P \) is still the permutation matrix and \( P^0 = I \). By permutation invariance and convexity of \( \hat{\Lambda}(\mathbb{R}) \) it is clear that \( y \in \hat{\Lambda}(\mathbb{R}) \). Moreover, \( \sum_{n=0}^{N-1} P^n \) is an \( N \times N \) matrix all of whose entries are equal to 1. Hence, \( y_i = \frac{1}{N} e^\top x \) for all \( i = 1, \ldots, N \), so that \( y \in \ell \) as well. Since \( v_N^{1/N} \) is concave, it follows that

\[
v_N^{1/N}(y) = v_N^{1/N}\left( \frac{1}{N} \sum_{n=0}^{N-1} P^n x \right)
\geq \frac{1}{N} \sum_{n=0}^{N-1} v_N^{1/N}(P^n x)
= v_N^{1/N}(x).
\]

Since this is true for all \( x \in \hat{\Lambda}(\mathbb{R}) \), \( x_{\text{max}} \in \ell \).

Secondly, let \( x \in \ell \cap \hat{\Lambda}(\mathbb{R}) \), i.e. \( x = \delta e \) for some \( 0 \leq \delta \leq 2/T \), and consider the interval constraints for \( x \) corresponding with the circular arcs. These are

\[
0 \leq \tau_i + 1 - \tau_i + \delta T \leq 2 \tag{4.282}
\]

for \( i = 1, \ldots, N \), where \( \tau_{N+1} \) should be read as \( \tau_1 \). By reversing the direction of these arcs the volume does not change. Hence, \( v_N(x) \) remains the same if the circular arcs would read

\[
0 \leq \tau_i - \tau_{i+1} + \delta T \leq 2 ,
\]

which is equivalent to

\[
0 \leq \tau_{i+1} - \tau_i + \left( \frac{2}{T} - \delta \right) T \leq 2 .
\]

Thus, we see that \( v_N(\delta e) = v_N((2/T - \delta)e) \), so that, by concavity of \( v_N^{1/N} \), we conclude that \( x_{\text{max}} = \epsilon/T \). \( \square \)

**Proof 2** More directly, we can use Corollary 4.3. To this end, we calculate that

\[
l + u = \begin{pmatrix} 2e \\ (2N - 3)e \end{pmatrix} .
\]

Also, a fundamental circuit consists of a circular arc, corresponding to one of the first \( N \) entries of \( l + u \), plus two radial arcs, both corresponding to one of the last \( N \) entries of \( l + u \). Furthermore, these two radial arcs appear in the fundamental circuit with opposite sign. Hence, if \( f \) is a fundamental circuit, then

\[
f^\top(l + u) = 2 .
\]
Since this is true for all fundamental circuits, it follows that
\[ \frac{1}{2T} F(l + u) = \frac{1}{T} e. \]
(4.287)

The result, then, follows directly from Corollary 4.3.

\[ \square \]

### 4.4.3 A real-life example

In this section we use the situation at the Dutch railway station Arnhem CS as an example. The timetable structure is described in detail by Tables 3.1 through 3.4 in Chapter 3, see also [100], and it has \( N = 23 \) free event times and \( M = 54 \) constraints.

Exhaustive enumeration of the elements in \( \Lambda(\mathbb{Z}) \) yields that there are 1,703 different phase shifts. Furthermore, if we take a look at the circuit \( c \) in the constraint graph corresponding to constraints \( \{2, 8, 11, 18\} \) in Tables 3.1 through 3.3, we see that \( c^T p = 0 \) for all \( p \in \Lambda(\mathbb{Z}) \). Substituting this identity in the timetable structure yields the interval constraints
\[
\begin{cases}
2 \leq \tau_2 - \tau_1 \leq 6 \\
2 \leq \tau_3 - \tau_2 \leq 5 \\
5 \leq \tau_4 - \tau_3 \leq 6 \\
-9 \leq \tau_1 - \tau_4 \leq -6 .
\end{cases}
\]
(4.288)

Adding them up gives \( 0 \leq 0 \leq 11 \), so that the lower bounds hold at equality. Hence, for all \( p \in \Lambda(\mathbb{Z}) \) we have that \( \text{dim}(p) \leq N - 3 \). It was found that out of the 1,703 phase shifts precisely 1,168 phase shifts have a corresponding timetable set of dimension \( N - 3 \).

With \( \Lambda_+ \) we denote the set of these phase shifts. The remaining 535 timetable sets have dimension less than \( N - 3 \) and are united in the set \( \Lambda_- \). Thus, for the phase shifts in \( \Lambda_- \) \( R \) assigns no probability mass.

We generated 17,030 samples from \( R^2 \) and performed a frequency count for each of the 1,703 phase shifts. It turned out that 12.59\% of the phase shifts in \( \Lambda_+ \) occurred with frequency zero, whereas for the phase shifts in \( \Lambda_- \) this number was 39.63\%, thereby increasing the confidence in the heuristic.

We also looked at the data from another viewpoint. That is, if \( \eta = N \), then \( R \) assigns probability mass proportional to the \( N \)-dimensional geometric volumes of the underlying timetable sets. The question comes up as to whether some weak or strong version of this property also holds true when \( \eta < N \). However, answering this question for our example requires that 1,703 geometric volumes must be calculated, which is too complex. Therefore, to investigate the question we plotted the sample frequencies against the distances of the phase shifts (using \( \rho \)) to the center \( x_{\text{max}} \) of \( \hat{\Lambda}(\mathbb{R}) \) (as in Proposition 4.2 and Corollary 4.3), see Figure 4.5.

The dots with the smallest diameter correspond to a single phase shift and the bigger a dot, the more phase shifts it corresponds to. Seven phase shifts are not shown, because they had a frequency of over 85 (the highest frequency encountered was 126), although they are involved in all calculations. With \( \mu \) we denote average values.
Figure 4.5: Results for $R^\infty$ applied to Arnhem CS for all 1,703 phase shifts together.
The idea behind showing this plot is based on two assumptions. The first one is that \( R^n \) is indeed a good approximation of \( R \). The second assumption is that, if the hypothesis is true, phase shifts with a large distance from \( x_{\max} \) are assigned less probability mass by \( R \) than phase shifts for which this distance is small. In particular, if \( \eta = N \), this assumption seems plausible because \( v_N^{1/N} \) is concave and \( x_{\max} \) maximizes \( v_N \). Based on these assumptions, the cloud of points in Figure 4.5 should be spread out around a line with negative slope. Performing a linear regression yields that for frequency \( f \)

\[
\rho(f) = 1.419 - 0.00339 f
\]  

with \( t \)-value 34.96 for the slope, thereby strongly rejecting the hypothesis that the slope is zero.

In the scatter plots of Figures 4.6 and 4.7 we show the same data as in Figure 4.5, only this time the 1,168 phase shifts in \( \Lambda_+ \) are set apart from the other 533 phase shifts. Again the phase shifts with a frequency of over 85 were left out (five in Figure 4.6 and two in Figure 4.7). Performing linear regression on both sets of data yields

\[
\rho(f) = 1.399 - 0.00301 f
\]  

with \( t \)-value 30.77 for the data in Figure 4.6 and

\[
\rho(f) = 1.452 - 0.00320 f
\]  

with \( t \)-value 14.98 for the data in Figure 4.7. In both cases the negative slope is quite significant.

More importantly, from these two figures we see that in this example the phase shifts that should be out of sight, i.e. the phase shifts in \( \Lambda_- \), have much lower average frequency and a somewhat greater distance to \( x_{\max} \) than the phase shifts in Figure 4.6, which is precisely what we want.

### 4.5 Computational complexity

In general we do not have a listing of the phase shifts and in the previous sections of this chapter we showed that we do not need such a listing provided we can deal with the complexity of the rounding problem. Alternatively we could turn to a Markov chain sampling approach, see Chapter 2. In such an approach we also do not need a listing of the phase shifts and sample from \( \hat{\Lambda}(\mathbb{Z}) \) by making a random walk on it. Moreover, to define a Markov chain on \( \hat{\Lambda}(\mathbb{Z}) \) we only need one phase shift as a starting point. However, with the Markov chain sampling approach we now need to define a set of step directions in order to walk on \( \hat{\Lambda}(\mathbb{Z}) \) and below we will argue that calculating such a step set is probably very hard.

In order to have access to all phase shifts, given an initial one, we must know at least some structure on \( \hat{\Lambda}(\mathbb{Z}) \) that we can use to reach a phase shift from another phase shift. Loosely speaking, we would like to have something like a test set [120], which is a set of
Figure 4.6: Results for $\mathcal{R}^\infty$ applied to Arnhem CS for the phase shifts with a timetable set of dimension $N - 3$. 
Figure 4.7: Results for $\mathcal{R}^\infty$ applied to Arnhem CS for the phase shifts with a timetable set of dimension less than $N - 3$. 
integral vectors in $\mathbb{Z}^{M-N}$ such that each phase shift can be reached from each other phase shift by using the vectors in the test set as steps. In [8] it is shown how to set up a Markov chain converging to the desired limiting and equilibrium distribution using such set of steps. As to the construction of test sets, in [135], for example, a so-called reduced Gröbner basis is developed using concepts in polynomial ideal theory. In [33, 135] the Buchberger algorithm [28] for calculating Gröbner bases for polynomial ideals is transformed into an algorithm for calculating a Gröbner basis. However, numerical results are limited to relatively small integer programming problems and for the purpose of calculating a test set for the phase shifts in $\hat{\Lambda}(\mathbb{Z})$ this is not sufficient. Other examples of test sets, for which similar remarks about the practical applicability apply, are e.g. the Neighbors of the origin [117, 118] or the one described in [120]. It is still questionable whether useful test sets of modest size will exist for our specific problem, let alone for general problems. This feeling is based on the following series of negative results concerning the structure of $\hat{\Lambda}(\mathbb{Z})$.

Consider the system

$$
\begin{align*}
\hat{C}p & \leq |U(C)| \\
p & \in \mathbb{Z}^{M-N}
\end{align*}
$$

(4.292)

where the rounding is done component-wise. Since $\hat{C}p$ is integral, rounding down the right-hand sides like this does not affect the set of solutions, i.e. the solutions to (4.292) are the phase shifts in $\Lambda(\mathbb{Z})$. Hence, if (almost) uniform generation of solutions to (4.292) were easy, then it would also be easy to (almost) uniformly generate solutions from $\Lambda(\mathbb{Z})$, and vice versa.

In many respects the polyhedral structure of $\hat{\Lambda}(\mathbb{R})$ is susceptible to analysis, witness the results in Section 4.2.1. For the polyhedral structure of

$$
[\hat{\Lambda}(\mathbb{R})] = \{ x \in \mathbb{R}^{M-N} \mid \hat{C}x \leq |U(C)| \} 
$$

(4.293)

the story seems to be completely the opposite, witness the following series of negative results. It is convenient to introduce

$$
L(c) = \frac{1}{T}(t^T c^+ - u^T c^-)
$$

(4.294)

for elementary circuits $c$ of $G$. Note that

$$
L(c) = -U(-c)
$$

(4.295)

Proposition 4.12 A subsystem of $\hat{C}x \leq |U(C)|$ defining an implicit equality contains a constraint, corresponding to an elementary circuit $c$, for which $[L(c)] = [U(c)]$.

Proof Let $C_*$ be the submatrix of $C$ consisting of the elementary circuits of $G$, but such that if $c$ is a row of $C_*$, then $-c$ is not. Define $\hat{C}_*$ as matrix $C_*$ with its last $N$ columns, corresponding to tree $T$, deleted. Note that

$$
[\hat{\Lambda}(\mathbb{R})] = \{ x \in \mathbb{R}^{M-N} \mid [L(C_*)] \leq \hat{C}_* x \leq |U(C_*)| \} 
$$

(4.296)
If the system
\[
[L(C_\ast)] \leq \tilde{C}_\ast x \leq [U(C_\ast)]
\]
defines implicit equalities, there exist nonnegative vectors \( \alpha, \beta \), such that
\[
\alpha^T \tilde{C}_\ast = \beta^T \tilde{C}_\ast
\]
and
\[
\alpha^T [L(C_\ast)] = \beta^T [U(C_\ast)].
\]
Note that, since
\[
C_\ast = \tilde{C}_\ast F,
\]
(4.298) implies
\[
\alpha^T C_\ast = \beta^T C_\ast.
\]
Without loss of generality we assume that
\[
(\alpha + \beta)^T e = 1.
\]
By definition we have
\[
U(C_\ast) + L(C_\ast) = C_\ast (u + l)/T,
\]
so that
\[
\alpha^T (U(C_\ast) + L(C_\ast)) = \beta^T (U(C_\ast) + L(C_\ast))
\]
follows from (4.301). Assume, by way of contradiction, that for all elementary circuits \( c \) we have
\[
[U(c)] - [L(c)] \geq 1.
\]
By virtue of (4.299) we find that
\[
\alpha^T [U(C_\ast)] - \beta^T [L(C_\ast)] \geq (\alpha + \beta)^T e \quad \text{and} \quad 1.
\]
Also, by definition,
\[
\beta^T U(C_\ast) < \beta^T [U(C_\ast)] + \beta^T e
\]
and
\[
\alpha^T L(C_\ast) > \alpha^T [L(C_\ast)] - \alpha^T e.
\]
Hence,
\[
\beta^T U(C_\ast) - \alpha^T L(C_\ast) < \beta^T [U(C_\ast)] - \alpha^T [L(C_\ast)] + 1.
\]
Combining these results yields

\[ 1 \leq \alpha^T [U(C_*)] - \beta^T [L(C_*)] \leq \alpha^T U(C_*) - \beta^T L(C_*) = \beta^T U(C_*) - \alpha^T L(C_*) < \beta^T [U(C_*)] - \alpha^T [L(C_*)] + 1 , \]

so that

\[ \beta^T [U(C_*)] - \alpha^T [L(C_*)] > 0 , \]

which contradicts (4.299).

\[ \square \]

**Corollary 4.4** The problem to decide whether \([\hat{\Lambda}(\mathbb{R})]\) is of full dimension is equivalent to the problem of deciding about the existence of an elementary circuit \(c\) for which \([L(c)] = [U(c)]\).

**Proof** If \([\hat{\Lambda}(\mathbb{R})]\) is not of full dimension, there is an implicit equality, so that on account of Proposition 4.12 there exists an elementary circuit \(c\) for which \([L(c)] = [U(c)]\). The converse is obvious.

Without concluding that this problem is NP-complete we expect it at least to be very difficult from computational point of view. To support this view we mention the following result.

**Proposition 4.13** The problem of deciding whether \(G\) has an elementary circuit \(c\) of length at least three for which \([L(c)] = [U(c)]\) is NP-complete, even if we restrict ourselves to the class of timetable structures for which it is known that \(0 \in \hat{\Lambda}(\mathbb{Z})\).

**Proof** We reduce the problem Partition [56] to the problem described in the proposition. This is the problem of, given a set \(\beta = \{\beta_1, \ldots, \beta_k\}\) of natural numbers with even sum \(P\), to decide if there exists a bi-partition of it such that for both classes the element sum equals \(P/2\). Given \(\beta\) construct a timetable structure like in Figure 4.8 and set

\[ T = \frac{P + 2}{2} . \]

It is easily seen that each elementary circuit of length at least three goes through arc 1 and through either arc \((i, a)\) or \((i, b)\), but not both, for all \(i \in \{1, \ldots, k\}\).

Let \(c\) be such circuit, say, without loss of generality, clock-wise directed. If \([L(c)] = [U(c)]\), then

\[ \left[ \frac{2}{P + 2}(0^T c^+ - \beta^T c^-) \right] = \left[ \frac{2}{P + 2}(\beta^T c^+ - 0^T c^-) \right] , \]

or, equivalently,

\[ \frac{2}{P + 2}\beta^T c^- + \frac{2}{P + 2}\beta^T c^+ = 0 . \]
Figure 4.8: Reducing a Partition problem to a timetable structure.

Hence,
\[ \frac{2}{P+2} \beta^T e^+ < 1 \]  \hspace{1cm} (4.321)

and
\[ \frac{2}{P+2} \beta^T e^- < 1 . \]  \hspace{1cm} (4.322)

Furthermore, we have
\[ \beta^T c^+ + \beta^T c^- = P , \]  \hspace{1cm} (4.323)

so that
\[ \beta^T c^+ = \frac{1}{2} P \]  \hspace{1cm} (4.324)

and
\[ \beta^T c^- = \frac{1}{2} P \]  \hspace{1cm} (4.325)

must hold, thus implying the aspired partitioning.

Conversely, let \((H, K)\) partition the elements of \(\beta\) such that both subset sums are equal and define elementary circuit \(c\) of the graph in Figure 4.8 by setting \(c_0 = 0\), \(c_i = 1\) if \(\beta_i \in H\), and \(c_i = -1\) if \(\beta_i \in K\). Then
\[ \beta^T c^+ = \beta^T c^- = \frac{1}{2} P \]  \hspace{1cm} (4.326)

and
\[ [L(c)] = \left[ \frac{2}{P+2} (0^T c^+ - \beta^T c^-) \right] \hspace{1cm} (4.327) \]
\[ = 0 , \]  \hspace{1cm} (4.328)

and also
\[ [U(c)] = \left[ \frac{2}{P+2} (\beta^T c^+ - 0^T c^-) \right] \hspace{1cm} (4.329) \]
\[ = 0 , \]  \hspace{1cm} (4.330)
implying the result.

Note that in the proposition the condition that $0 \in \hat{A}(\mathbb{Z})$ is equivalent to the condition that we must explicitly know at least one feasible phase shift in $\hat{A}(\mathbb{Z})$ because we can always translate system (4.2) over such a phase shift to comply with the condition that $0 \in \hat{A}(\mathbb{Z})$. Finally, we have

**Proposition 4.14** The problem of deciding whether there exists an elementary circuit $c$ of $G$ for which $[L(c)] < [U(c)]$ is NP-complete, even if we restrict ourselves to the class of timetable structures for which it is known that $0 \in \hat{A}(\mathbb{Z})$.

**Proof** Reduce the problem Directed Hamiltonian Cycle [56] to the problem described in the proposition in the following way. Take an arbitrary, simple, graph and convert it to a timetable structure by adding interval $[-1, 1]$ to all arcs and setting $T = N + 1$. Then, an elementary circuit $c$ is Hamiltonian if and only if $[L(c)] < [U(c)]$.

Although these results do not formally prove the difficulty of generating phase shifts, they do indicate that $\hat{A}(\mathbb{Z})$ seem to lack any exploitable structure, even if we have one particular solution at hand. Moreover, so far we have not been able to prove, or even conjecture, any enjoyable property of $\hat{A}(\mathbb{Z})$ or $[\hat{A}(\mathbb{R})]$.

### 4.6 Some remarks

We conclude this chapter with two remarks, the first of which deals with generating phase shifts from $\hat{A}(\mathbb{R})$ in compliance with the uniform distribution. Our second remark concerns the phase shift that is optimal in a sense related to robustness of timetables. We stress that some statements in this section are speculative and are merely meant to suggest solution strategies for the particular problems that are addressed in this section.

#### 4.6.1 Uniform generation of phase shifts

By using the robustness distribution to generate phase shifts at random we mean to favor robust timetables over others. Although this complies with the standard practice of compiling timetables, it is not unlikely that there are situations in which the timetable or infrastructure planner is completely indifferent between any of the phase shifts.

Generating phase shifts uniformly at random can in principle be done by deiving a Markov chain whose state space is the set of phase shifts and whose limiting distribution is the uniform one. As we saw in the above, such Markov chain requires something as a test set and test sets are hard to calculate. Thus, suppose that we resort to a heuristic approach like for the robustness distribution. That is, we generate uniformly at random a vector $x$ in $\hat{A}(\mathbb{R})$ and then round $x$ to $p = r(x)$ in $\hat{A}(\mathbb{Z})$.

Unfortunately, whereas with the robustness distribution we could use a simple transformation technique to generate from $\hat{A}(\mathbb{R})$, for the uniform distribution we do not know
of an efficient way to sample from it. Nevertheless, below we will discuss two possible approaches.

The first method we consider to generate a vector uniformly at random from $\mathbb{A}(\mathbb{R})$ is the Markov chain sampling technique, see Chapter 2. In particular, we refer to the ball-walk variant in the remainder of this section, so that the sample follows a distribution that has a continuous probability density. Although results about the computational complexity of this technique are celebrated in theoretical environments, they are not yet in practice. Even exploiting the symmetry of the phase shift polytope by applying the following proposition will not improve the computational complexity much.

**Proposition 4.15** Let $u$ be the probability density of the uniform distribution on a centrally symmetric convex body $A \subset \mathbb{R}^n$ with center $c$ (i.e. $A$ is convex, compact, symmetric, and of full dimension). Also, let $q$ be some other probability density on $A$. If $\underline{x}$ is sampled from $q$ and $\bar{y} = h(\underline{x})$, where

$$ h(\underline{x}) = \begin{cases} x / 2c - x & \text{with probability } \frac{1}{2}, \\ 2c - x & \text{with probability } \frac{1}{2}, \end{cases} $$

then

$$ \int_A |u(x) - q(x)| \, dx \geq \int_A |u(x) - \bar{q}(x)| \, dx, $$

where $\bar{q}$ is the probability density on $A$ after applying the transformation $h$ to a sample from $q$. Moreover, the inequality in (4.332) is tight if $q$ is centrally symmetric in $c$.

**Proof** Note that

$$ \bar{q}(x) = \frac{1}{2} q(x) + \frac{1}{2} q(2c - x), $$(4.333)

for all $x \in A$. Then,

$$ \int_A |u(x) - \bar{q}(x)| \, dx $$

(4.334)

$$ = \int_A |u(x) - \frac{1}{2} q(x) - \frac{1}{2} q(2c - x)| \, dx $$

(4.335)

$$ \leq \frac{1}{2} \int_A |u(x) - q(x)| \, dx + \frac{1}{2} \int_A |u(x) - q(2c - x)| \, dx $$

(4.336)

$$ = \frac{1}{2} \int_A |u(x) - q(x)| \, dx + \frac{1}{2} \int_A |u(2c - x) - q(2c - x)| \, dx $$

(4.337)

$$ = \int_A |u(x) - q(x)| \, dx. $$

(4.338)

If $q$ is centrally symmetric with center $c$, then $q(x) = q(2c - x)$ and the inequality becomes an equality. $\square$

Intuitively, the transformation $h$ effectively 'doubles' the area the Markov chain traverses, thereby reducing the problem that the chain might get stuck in some part of $\mathbb{A}(\mathbb{R})$. Note that, by symmetry of the feasible region, applying the transformation to each iteration
point of the Markov chain is equivalent to just applying it to the final sample point used from a given chain. Furthermore, note that transformation $h$ can simply be implemented as a coin-tossing experiment with a fair coin. Note that when starting the walk in $c$, applying the coin-tossing trick is of no use.

Alternatively, we could sample from $\hat{\Lambda}(\mathbb{R})$ by following a constraint generation approach like in Chapter 3. That is, we first generate $x_0$ uniformly at random from the set

$$\{ x \in \mathbb{R}^{M-N} \mid -U(-F) \leq x \leq U(F) \},$$

(4.339)

which is a box in $\mathbb{R}^{M-N}$ that contains $\hat{\Lambda}(\mathbb{R})$, see Section 4.1.1. Then, if $x_0 \in \hat{\Lambda}(\mathbb{R})$, $x$ follows the uniform distribution on $\hat{\Lambda}(\mathbb{R})$. If, on the other hand, $x_0 \not\in \hat{\Lambda}(\mathbb{R})$, then we use the FDP algorithm in [111] to obtain an elementary circuit $c$ for which $c^T x_0 > U(c)$ to cut off $x$, i.e. we extend the system $-U(-F) \leq x \leq U(F)$ with the constraint $c^T x \leq U(c)$. Then, we generate $x_1$ uniformly at random from this extended system using the Markov chain sampling technique. If $x_1 \not\in \hat{\Lambda}(\mathbb{R})$ we add another constraint, and so on.

However, witnessing Proposition 4.3, this approach may eventually lead to the generation of a constraint for each elementary circuit as each of these constraints describes a facet of $\hat{\Lambda}(\mathbb{R})$. Clearly, this would make the constraint generation approach an unattractive alternative from computational point of view.

### 4.6.2 Phase shift optimization

If $\eta = N$, then $\mathcal{R}$ assigns probability mass to the phase shifts proportional to the $N$-dimensional geometric volume of the underlying sets of timetables. A planner may be interested in solving the program

$$P : \max \{ v_N(p) \mid p \in \hat{\Lambda}(\mathbb{Z}) \},$$

(4.340)

because a large volume of $K(p)$ may be an indicator of robustness.

By virtue of the foregoing, solving $P$ is computationally too hard. A reasonable alternative might be to sample several phase shifts from $\mathcal{R}^S$ and pick the one that is most frequent observed. However, the number of phase shifts that can be generated in a reasonable amount of time will be restricted, so that it may very well happen that not one of the phase shifts will be observed more than once. Therefore, we may have to settle for the generation of only a few phase shifts and try to select the best one by making rough estimates of the associated volumes using e.g. the algorithm for counting timetables in Chapter 5.
Chapter 5
Random Generation and Counting of Railway Timetables

In Chapter 3 we discussed a mathematical model for modeling and compiling railway timetables. It turned out that the set of feasible timetables can be partitioned into a finite number of classes, each of which corresponds to a so-called phase shift. A phase shift is an integral vector that is part of the model and must be determined first before we can search for a timetable. Determining a phase shift for which timetables exist is a difficult combinatorial problem and is addressed in Chapters 3 and 4. In this chapter\(^1\) we assume that a feasible phase shift is given and study the timetable model for fixed phase shift. In particular we are interested in generating timetables from a timetable class uniformly at random, the purpose of which is stated Chapter 1 and will be recapitulated below.

As explained in Chapter 1 we generate several reference timetables in order to derive specifications of railway infrastructure. At a later stage, to check the robustness of the infrastructure design, for each reference timetable we generate several other timetables from the same class that the reference timetable belongs to. This way we can study the sensitivity of the design for small changes in the reference timetables.

In this chapter we present an algorithm to generate timetables from a given timetable class uniformly at random, i.e. each timetable in the class has equal probability of being selected. The basic idea behind the algorithm is easily explained, although the details are very much involved. That is, a Markov chain is constructed whose state space is the set of timetables and the limiting distribution is the uniform distribution on this set. The Markov chain sampling approach to uniformly generate timetables comes down to making a nearest neighbor random walk on the integral points in a polytope. In general this does not work because the state transition graph will not be connected. Remarkably, in our case this walk does work, but only after we have substituted out all implicit equalities in the polyhedral description of the set of feasible timetables. Since with the Markov chain sampling method we can only approximate the uniform distribution (although it can be approached as close

\(^1\)This chapter is based on ‘Michiel A. Odijk: Sensitivity Analysis of a Railway Station Track Layout with Respect to a Given Timetable’, which will appear in the European Journal of Operational Research.
as desired), we call the algorithm an almost uniform generator. Moreover, when we speak of (almost) uniform generation we actually mean (almost) uniform generation at random.

For the purpose of sensitivity analysis of infrastructure designs we prefer to generate timetables uniformly at random over generating only corner points of the forementioned polytope because random generation reflects the uncertainties at the strategic decision level much better. What's more, there seems to be no reason whatsoever to use any other distribution than the uniform one.

The uniform generator turns out to have a nice spin-off. That is, owing to a result in [68], randomized approximate counting of the number of timetables in a timetable class is reducible to almost uniform generation. Thus, the timetable planner can also estimate the size of its 'planning space' in terms of the number of alternative timetables within a timetable class. This facility might be useful when searching for a timetable class that is robust with respect to planning in the sense of Chapter 4. In the present chapter we make the reduction of counting to uniform generation explicit.

The remainder of this chapter is organized as follows. In the next section we describe and discuss the model we use in this chapter. Thereafter, in Section 5.2 we present an algorithm to almost uniformly generate timetables from a timetable class. In Section 5.3 we demonstrate the theory by means of a small example. Using two other examples, in Section 5.4 we comment on the performance of the uniform generator. In the final section we state an algorithm for approximately counting the number of timetables in the timetable class under study.

### 5.1 The model

Here we adopt the railway timetable model used in Chapter 3. In this model a railway timetable is a mapping of a set of arrival and/or departure events (numbered consecutively from 0 to \( N \)) to \( \mathbb{R} \). To model driving times, headway times, connections between trains, and so on, events are coupled by relating their associated event times \( \tau_0, \ldots, \tau_N \) with constraints, called interval constraints, that adopt the form

\[
 l_{ij} \leq \tau_i - \tau_j + T p_{ij} \leq u_{ij} ,
\]

where \( T, l_{ij} \), and \( u_{ij} \) are integers such that \( 0 < u_{ij} - l_{ij} < T \). The event times \( \tau_i \in \mathbb{R} \) and \( \tau_j \in \mathbb{R} \) of event \( i \) and \( j \), respectively, and the \( p_{ij} \in \mathbb{Z} \) are the unknowns. Parameter \( T \) is the period of any feasible timetable and is often set to 60 minutes, thereby implying the desire to construct hourly patterns. In vector notation the timetable model is the mixed integer system of linear inequalities

\[
l \leq B^T \tau + T p \leq u ,
\]

where \( B \in \{0, \pm 1\}^{(N+1) \times M} \) is the vertex-arc incidence matrix of a simple directed graph \( \mathcal{G} \) and \( l, u \) are integral vectors satisfying \( 0 < u - l < T \), where the inequalities are to be understood component-wise. Graph \( \mathcal{G} \) is such that its vertices correspond to the events
(and hence the event times) and such that its arcs correspond to the \( M \) interval constraints. The a-priori unknown integral vector \( p \) expresses the periodicity of the timetables and is called the phase shift. Somewhat sloppy we identify the \( i \)-th event time as well as its vertex in \( G \) with \( \tau_i \).

Whereas Chapters 3 and 4 are concerned with the combinatorial problem of searching for a phase shift \( p \) for which (5.2) is feasible, in this chapter we assume that such a vector exists and we fix one. Thus, from here on we consider the system

\[
l \leq B^T \tau \leq u,
\]

(5.3)

where \( G \) is as in the above and, if one wishes, \( Tp \) is part of \( l \) and \( u \). Note that \( l < u \) is implied. Solving this system is referred to as the Feasible Differential Problem (FDP) [111] and the model is called the FDP model. Here we restrict ourselves to integral solutions \( \tau \) only and refer to the set of solutions with \( K \). This restriction is not severe because \( B \) is totally unimodular and \( l \) and \( u \) are integral. Hence, the convex hull of \( K \) is integral, so that (5.3) is feasible if and only if an integral solution exists. See [111] for an \( \mathcal{O}(N^3) \) algorithm that solves (5.3).

The main purpose of this chapter is to present an algorithm for almost uniform generation of elements in \( K \). As to the railway application of the FDP model, the elements in \( K \) are timetables.

We denote an FDP instance with graph \( G \) and intervals \( [l, u] \) by \((G, l, u)\). We encode circuits of \( G \) with their \( M \)-dimensional incidence vectors denoting arc orientations with 1 if the orientation is consistent with some canonicle direction in which circuits are traversed and \(-1 \) otherwise. Arcs that are not part of a circuit are marked 0 in the associated incidence vector. For a circuit \( c \) we define \( c^+ = \max(0, c) \) and \( c^- = -\min(0, c) \). Elementary circuits are circuits that are not decomposable into smaller circuits.

The following lemma is frequently used in this chapter.

**Lemma 5.1** Let \( a, b \in \mathbb{R}^M \) with \( a \leq b \) and let \( B \) be the vertex-arc incidence matrix of a directed graph. Then \( a \leq B^T \tau \leq b \) is feasible if and only if

\[
b^T c^+ - a^T c^- \geq 0
\]

(5.4)

for all elementary circuits \( c \).

**Proof** See [111]. \( \square \)

Some further remarks are in order. First, adding a multiple of all-one vector \( e \) to a timetable doesn’t change its characteristics and if \( \tau \in K \), then so is \( \tau + \alpha e \) for all \( \alpha \in \mathbb{Z} \). Thus we may restrict ourselves to timetables with \( \tau_0 = 0 \) and replace the definition of \( K \) with

\[
K = \{ \tau \in \mathbb{Z}^{N+1} \mid l \leq B^T \tau \leq u, \; \tau_0 = 0 \}
\]

(5.5)

without losing or creating timetables. By doing this, \( K \) becomes finite and the convex hull of \( K \) becomes a polytope in \( \{ x \in \mathbb{R}^{N+1} \mid x_0 = 0 \} \). As a consequence the uniform
(discrete probability) distribution $U$ on $K$ can be defined by

$$U(\tau) = \frac{1}{\#K}$$ (5.6)

for all $\tau \in K$, where $\#K$ is the cardinality of $K$. Furthermore, the FDP model possesses the self-reducibility property [68, 119], which means that if we fix some of the entries of $\tau$ the problem of extending $\tau$ to a solution to (5.3) or establishing that this is not possible is an FDP instance by itself. To see this, note that fixing entry $\tau_i$ of $\tau$ to $\alpha \in \mathbb{Z}$ comes down to substituting $\tau_0 + \alpha$ for $\tau_i$ in (5.3) because $\tau_0 = 0$. This property will underly several results stated in this chapter. Finally, if for a row of (5.3), say corresponding to vertices $\tau_i$ and $\tau_j$, we have $l_{ij} = u_{ij}$, then $\tau_i + u_{ij}$ can be substituted for $\tau_i$, thereby yielding a smaller and equivalent FDP instance. Hence, there is no generality lost when assuming that $l < u$.

### 5.2 Uniform generation of timetables

In this section we show how to apply the Markov chain sampling technique to generate timetables at random from $K$ in compliance with a distribution that is close to $U$. To this end, we start with introducing a Markov chain and some definitions. Thereafter, still in this section, we prove that $U$ is the limiting distribution of this chain. However, this is only true under a technical, yet intuitively clear, condition on $G$, $l$, and $u$. Therefore, we also outline a procedure to check if this condition holds and, if not, how to enforce it without losing or creating timetables.

To grasp the idea of the approach explained in this section it is not necessary to understand the proofs of the formal results at first reading. That is, as the example in the following section sheds some light on the theory in this section, the reader might skip these proofs, read the following section, and backtrack to this section to read the proofs.

If $a, b, r > 0$, we say that $a$ approximates $b$ within ratio $1 + r$ if $a(1+r)^{-1} \leq b \leq a(1+r)$. Given $\epsilon > 0$, an algorithm that returns a timetable in $K$ such that the probability of returning $\tau$ is approximated by $U(\tau)$ within ratio $1 + \epsilon$ for all $\tau \in K$ is called an almost uniform generator. We also say that such algorithm generates uniformly with tolerance $\epsilon$. Note that this definition is stronger than the one given in [68], where an almost uniform generator is allowed to fail to return a solution with some fixed probability.

We almost uniformly generate a timetable from $K$ by defining a random walk on $K$ converging to $U$ in the following way. Let $\{N_k\}_{k \geq 1}$ be a row of i.i.d. stochastic variables, each of which has a distribution on $\{ \pm \epsilon_i \mid i = 1, \ldots, N \} \cup \{0\}$ with

$$Pr(N_k = 0) = \frac{1}{2},$$

$$Pr(N_k = \epsilon_i) = \frac{1}{4N},$$

and

$$Pr(N_k = -\epsilon_i) = \frac{1}{4N}$$ (5.7) (5.8) (5.9)
for all $k \geq 1$ and $i = 1, \ldots, N$. Recall that $c_i$ is the $i$-th unit vector of appropriate size. These variables constitute the stochastic driver of the random walk

$$\mathcal{W} = \{X_k\}_{k \geq 1}$$

which is defined by

$$\begin{cases} \Pr(X_1 = \hat{\tau}) = 1 \\ \Pr(X_1 \neq \hat{\tau}) = 0 \end{cases}$$

for some arbitrary timetable $\hat{\tau} \in K$, and

$$X_k = \begin{cases} X_{k-1} + N_{k-1} & \text{if } X_{k-1} + N_{k-1} \in K \\ X_{k-1} & \text{otherwise}, \end{cases}$$

for $k \geq 2$. Checking the condition $X_{k-1} + N_{k-1} \in K$ can be done by inspection of the interval constraints in $O(M)$ time. Finding a starting point $\hat{\tau} \in K$ is easy — in [111] an $O(N^3)$ algorithm is given to deal with this. Note that $\mathcal{W}$ is nothing but a nearest neighbor random walk over the standard lattice inside the polytope described by (5.3) with self-loop probability of $\frac{1}{2}$ for each state.

Although in general random walks over the standard lattice within polytopes (even of full dimension) do not converge to the uniform distribution on the enclosed lattice points, this walk does as long as no elementary circuit of $G$ is critical, where

**Definition 5.1** An elementary circuit $c$ of $G$ is called critical if and only if

$$u^T c^+ - l^T c^- = 0.$$  

A circuit for which (5.13) holds is called critical because from Lemma 5.1 we know that the problem instance is feasible if and only if for all elementary circuits of $G$ it holds true that $u^T c^+ - l^T c^- \geq 0$. Note that, by exploiting integrality of $l$, $u$, and $c$, $u^T c^+ - l^T c^- \geq 1$ if and only if $c$ is not critical.

A critical circuit $c$ defines a number of implicit equalities that fix the difference between two event times. More precisely, suppose $c$ has length $v$ and select a subsystem of $l \leq B^T \tau \leq u$ such that constraint $B_m^T \tau \leq u_m$ is selected if and only if $c_m = 1$ and constraint $l_m \leq B_m^T \tau$ is selected if and only if $c_m = -1$. Then, by summing these constraints we derive that

$$0 = \tau^T B c$$

$$\leq u^T c^+ - l^T c^-$$

$$= 0 ,$$

thus implying $B_m^T \tau = u_m$ for all arcs $m$ for which $c_m = 1$ and $B_m^T \tau = l_m$ for all arcs $m$ for which $c_m = -1$. 


Definition 5.2 A vertex $\tau_i$ is called a free vertex if

$$\min\{\tau_i \in \mathbb{Z} \mid \tau \in K\} < \max\{\tau_i \in \mathbb{Z} \mid \tau \in K\}$$

and fixed otherwise.

Note that, if vertex $\tau_i$ is incident to a critical circuit that is also incident to $\tau_0$, then $\tau_i$ is fixed.

At this point we are ready to prove that, if $G$ does not contain critical elementary circuits, then $W$ converges to $U$. This is done in Proposition 5.1 where we use the following lemma.

Lemma 5.2 Let $(G, l, u)$ be an FDP instance without critical elementary circuits and with $l < u$ and let $(\tilde{G}, 1, \tilde{u})$ be another FDP instance with set of timetables $\tilde{K}$ that originates from $(G, l, u)$ by adding a number of additional constraints of the form $\alpha \leq \tau_i - \tau_0 \leq \beta$ with $\alpha \leq \beta$ and $i \in \{1, \ldots, N\}$. Let $\tau_i$ be a free vertex in $(\tilde{G}, 1, \tilde{u})$, say

$$a = \min\{\tau_i \in \mathbb{Z} \mid \tau \in \tilde{K}\} < \max\{\tau_i \in \mathbb{Z} \mid \tau \in \tilde{K}\} = b.$$ 

Then (a) for all $z \in \{a, \ldots, b\}$ there is a timetable $\tau \in \tilde{K}$ with $\tau_i = z$, (b) for each vertex $\tau_j$, $j = 0, \ldots, N$, adjacent to vertex $\tau_i$ we have $l_{ij} \leq u_{ij}$, and (c) no elementary circuit of $\tilde{G}$ that is incident to vertex $\tau_i$ is critical.

Proof By convexity of the convex hull of $K$, for $z \in \{a, \ldots, b\}$ there is a solution to $l \leq B^\top \tau \leq u$, $\tau_0 = 0$, $\tau \in \mathbb{R}^{N+1}$. Then, (a) follows from the observation that the polytope defined by this system is integral. Since $l < u$ is assumed for $(G, l, u)$, if $l_{ij} = u_{ij}$ for some vertex $\tau_j$ adjacent to vertex $\tau_i$, then $j = 0$ must hold. However, this would fix $\tau_i$, thus contradicting the assumption that $\tau_i$ is free. As to (c), by assumption $G$ has no critical circuits so that any critical circuit of $\tilde{G}$ must be incident to vertex $\tau_0$. Thus, a critical circuit can not be incident to $\tau_i$, as this would fix vertex $\tau_i$.

Proposition 5.1 If $G$ does not contain critical elementary circuits, then the random walk $W$ on $K$ converges to the uniform distribution $U$ on $K$.

Proof Let $(K, E)$ be the standard lattice graph restricted to $K$, i.e.

$$E = \{ (\tau, \hat{\tau}) \subseteq K \times K \mid \|\tau - \hat{\tau}\|_2 = 1 \} ,$$

where $\|\cdot\|_2$ denotes the Euclidian distance. Since $K$ is the set of states of $W$, $E$ represents the state transitions (apart from self-loop probabilities) and, hence, $(K, E)$ is the state transition graph of $W$ with the self-loops omitted. Thus, $W$ can be seen as a particle randomly walking on $(K, E)$ visiting a sequence of points in $K$ by walking over the edges such that in each step with probability at least $\frac{1}{2}$ no move is made.

Convergence of $W$ to $U$ is implied by ergodicity and symmetry of $W$. More precisely, if a Markov chain is ergodic, the limiting distribution equals the stationary distribution,
and, if in addition it is symmetric, the stationary distribution is the uniform one. In our case, symmetry follows directly from the definition of $\mathcal{W}$.

Ergodicity of $\mathcal{W}$ means a-periodicity and irreducibility. The former property holds trivially as a result of the self-loop probabilities. Irreducibility of $\mathcal{W}$ holds if and only if $(K, E)$ is connected and we show next that this is the case if no elementary circuit of $\mathcal{G}$ is critical.

Suppose that there are no critical elementary circuits. Now, let $t, \hat{t} \in K$ be two different timetables in $K$. We will show that there is a path in $(K, E)$ from $t$ to $\hat{t}$. Choose index $i \in \{1, \ldots, N\}$ such that $t_i < \hat{t}_i$. Thus, vertex $\tau_i$ is not fixed and can adopt values in the range $t_i, \ldots, \hat{t}_i$. We show that for all $h \in \{t_i, \ldots, \hat{t}_i - 1\}$ there exists a pair of timetables $t^{h, 1}, t^{h, 2} \in K$ such that $t_i^{h, 1} = h$ and $t_i^{h, 2} = h + 1$ and such that $t_i^{h, 2} = t_i^{h, 1} + e_i \in K$. That is, for all $h \in \{t_i, \ldots, \hat{t}_i - 1\}$ it is possible to move from a timetable $t_i^{h, 1} \in K$ with $t_i^{h, 1} = h$ to a timetable $t_i^{h, 2} \in K$ with $t_i^{h, 2} = h + 1$ in a single step. See Figure 5.1.

Let $h \in \{t_i, \ldots, \hat{t}_i - 1\}$ and add the constraint $h \leq \tau_i - \tau_0 \leq h + 1$ and associated arc to $\mathcal{G}$, thereby creating FDP instance $(\hat{\mathcal{G}}, \hat{\bar{t}}, \hat{u})$. Note that this does not fix $\tau_i$ but restricts its value to $\{h, h + 1\}$. Hence, Lemma 5.2 applies to $(\hat{\mathcal{G}}, \hat{\bar{t}}, \hat{u})$. In $(\hat{\mathcal{G}}, \hat{\bar{t}}, \hat{u})$ consider the part of $\hat{\mathcal{G}}$ around vertices $\tau_0$ and $\tau_i$, see Figure 5.2. Also, consider another modified FDP instance, see the lower part of the same figure. That is, the latter instance equals $(\hat{\mathcal{G}}, \hat{\bar{t}}, \hat{u})$ except that, if $\tau_j$ is a vertex adjacent to $\tau_i$, then $u_{ij}$ is decreased by 1 if the corresponding constraint is $l_{ij} \leq \tau_i - \tau_j \leq u_{ij}$ and $l_{ij}$ is increased by 1 if the corresponding constraint reads $l_{ji} \leq \tau_j - \tau_i \leq u_{ji}$. It is easily verified that for any elementary circuit $c$ incident to vertex $\tau_i$ the value of $u^T c^+ - l^T c^-$ decreases by one. Hence, since $u^T c^+ - l^T c^- \geq 1$, in this modified FDP instance $u^T c^+ - l^T c^- \geq 0$ holds. Thus, the modified FDP instance in the lower part of Figure 5.2 is feasible, say that $t_i^{h, 1}$ is a feasible timetable. Clearly, $t_i^{h, 1}$ is also a timetable for $(\hat{\mathcal{G}}, \hat{\bar{t}}, \hat{u})$ and the same holds for $t_i^{h, 2} = t_i^{h, 1} + e_i$. Since the set of timetables $K$ for $(\hat{\mathcal{G}}, \hat{\bar{t}}, \hat{u})$ is contained in $K$, it follows that $\mathcal{W}$ can move in a single step from $\tau_i^{h, 1}$ to $\tau_i^{h, 2}$.

At this point the recursion comes in. That is, it remains to be seen that we can move over the edges of $(K, E)$ from $t_i^{h-1, 2}$ to $t_i^{h, 1}$ for all $h \in \{t_i, \ldots, \hat{t}_i\}$, where we define $t_i^{h-1, 2} = t$ if $h = t_i$ and $t_i^{h, 1} = t$ if $h = \hat{t}_i$. Let $h \in \{t_i, \ldots, \hat{t}_i\}$ and add the constraint and associated arc $h \leq \tau_i - \tau_0 \leq h$ to $\mathcal{G}$. If fixing $\tau_i$ this way implies that some other vertices become fixed as well, we add some more of these arcs to denote such fixing explicitly. The so-constructed graph consists of the original graph $\mathcal{G}$ that has no critical elementary circuits and some additional arcs that all emanate from vertex $\tau_0$. Hence, Lemma 5.2 applies once again. Note that, if $t_i^{h, 1} \neq t_i^{h-1, 2}$, then, obviously, there is a free vertex in the newly constructed graph.

Finally note that, if in the end only one vertex is free to move over a range of event times, then obviously $\mathcal{W}$ connects all these event times. Also note that the recursion cannot go deeper than $N$ levels.

Proposition 5.1 is a remarkable result by itself, since in general resorting to a smaller mesh size is necessary in order to enforce connectivity of the lattice graph enclosed by the convex body at hand, see e.g. [8, 44, 85]. What's more, from the proof of the proposition we learn
Figure 5.1: Path from timetable $t$ to timetable $\tilde{t}$ projected onto co-ordinate direction $i$. 

\{$\hat{t} \in K \mid \hat{t}_i = t_i$\}
Figure 5.2: Part of the FDP instance in the proof of the proposition.
that $W$ converges to $U$ if and only if no critical elementary circuit is incident to a free vertex. This condition is equivalent to saying that the subgraph of $G$ that corresponds to the arcs that belong to one of the critical elementary circuits is connected and contains vertex $\tau_0$.

In general some elementary circuits will be critical, so that the question comes up as to what the above theory is worth in practice. Fortunately, there is an algorithm for transforming a problem into a reduced problem with no critical circuits.

**Definition 5.3** An FDP instance $(\hat{G}, \hat{l}, \hat{u})$ with solution set $\hat{K}$ is a reduction of FDP instance $(G, l, u)$ with solution set $K$ if and only if (a) $\hat{N} < N$, where $\hat{N} + 1$ and $N + 1$ are the number of vertices of $G$ and $\hat{G}$, respectively, and (b) there exists an injective mapping $h : \{1, \ldots, \hat{N}\} \rightarrow \{1, \ldots, N\}$ such that $q : K \rightarrow \hat{K}$ defined by

$$q(\tau) = (\tau_0, \tau_{h(1)}, \ldots, \tau_{h(\hat{N})})$$

is a bijection. \hfill \Box

Note that if $(\hat{G}, \hat{l}, \hat{u})$ is a reduction of $(G, l, u)$, then $\#K = \#\hat{K}$, so that (almost) uniform generation from $K$ is equivalent to (almost) uniform generation from $\hat{K}$. Reductions arise in particular if in $(G, l, u)$ equalities implied by critical circuits are substituted out.

The algorithm to reduce $(G, l, u)$ to an FDP instance without critical elementary circuits starts by searching for a critical circuit $c$. If no such circuit is found, we are done. On the other hand, if one is found, the implied equalities are used to substitute out some variables. It is not difficult to see that this leads to a reduced FDP instance for which $l < u$ still holds. If necessary, relabel the vertices after having substituted out the equalities in order to establish that the vertices in the reduced instance are numbered consecutively. By iteratively repeating this procedure at most $N$ times we end up with a reduced problem without critical elementary circuits so that $W$ converges to $U$ by virtue of Proposition 5.1. By memorizing the implicit equalities we found in the course of the execution of the algorithm we can convert the timetable that $W$ returns for the reduced problem into a timetable for the original problem. This leaves us to show how to detect and identify a critical circuit.

Detecting a critical circuit begins with making the intervals $[l, u]$ a little bit tighter, i.e. define

$$\begin{cases}
  l_\delta = l + \delta e \\
  u_\delta = u - \delta e
\end{cases}$$

where

$$\delta = \frac{1}{N + 2},$$

and solve

$$P : \begin{array}{ll}
  \min & u_\delta^T y_1 - l_\delta^T y_2 \\
  \text{s.t.} & B(y_1 - y_2) = 0 \\
  & y_1, y_2 \geq 0
\end{array}$$

In Proposition 5.2 we will show that, if the value of $P$ equals 0, then there are no critical elementary circuits of $G$, and, if its value is less than 0, then for the solution $(y_1, y_2)$ that
is returned $y_1 - y_2$ is the incidence vector of a critical elementary circuit of $\mathcal{G}$. To this end, we first need to state the following lemma.

**Lemma 5.3** The set of extremal rays of the polyhedral cone

$$L = \{ (y_1, y_2) \in \mathbb{R}^{2M}_+ \mid B(y_1 - y_2) = 0 \}$$  \hfill (5.24)

is the union of

$$\{ (c^+, c^-) \in \{0, 1\}^{2M} \mid c \text{ is an elementary circuit of } \mathcal{G} \}$$  \hfill (5.25)

and

$$\{ (e_i, e_i) \in \{0, 1\}^{2M} \mid i = 1, \ldots, M \}$$  \hfill (5.26)

**Proof** Rewrite (5.24) as

$$L = \{ (y_1, y_2) \in \mathbb{R}^{2M}_+ \mid y_1 - y_2 = F^T x, \ x \in \mathbb{R}^{M-N} \} ,$$  \hfill (5.27)

where the rows of $F \in \{0, \pm 1\}^{(M-N) \times M}$ are the $M - N$ fundamental circuits of $\mathcal{G}$ with respect to an arbitrary, though fixed, spanning tree of $\mathcal{G}$. That is, any circuit $c$ of $\mathcal{G}$ can be written as $c = F^T x$ with $x \in \mathbb{R}^{M-N}$. Given $x \in \mathbb{R}^{M-N}$, the general solution to $y_1 - y_2 = F^T x$, $(y_1, y_2) \geq 0$ is

$$\begin{cases} 
  y_1 = (F^T x)^T + z \\
  y_2 = (F^T x)^- + z,
\end{cases}$$  \hfill (5.28)

where $z \in \mathbb{R}^M_+$. Hence, we may write

$$L = \{ ((F^T x)^+,(F^T x)^-) + (z, z) \mid x \in \mathbb{R}^{M-N}, \ z \in \mathbb{R}^M_+ \} .$$  \hfill (5.29)

Since the supports of $(F^Tx)^+$ and $(F^Tx)^-$ are disjoint, the solutions $(e_i, e_i)$, for $i = 1, \ldots, M$ can only be constructed if $x = 0$ (note that $F^Tx = 0$ if and only if $x = 0$ because $F$ is of full rank). Hence, $(e_i, e_i)$ is an extremal ray of $L$ for all $i = 1, \ldots, M$.

To show that for an arbitrary elementary circuit $c \in \{0, \pm 1\}^M$ the solution $(c^+, c^-)$ is also an extremal ray of $L$, suppose that $(c^+, c^-)$ is a positive combination of other solutions to $B(y_1 - y_2) = 0$, $(y_1, y_2) \geq 0$. Since the supports of $c^+$ and $c^-$ are disjoint, the vectors $e_i$, $i = 1, \ldots, M$, can not take part in such a combination. Suppose that for some $k \geq 2$

$$(c^+, c^-) = \sum_{i=1,\ldots,k} \lambda_i (c_i^+, c_i^-) ,$$  \hfill (5.30)

where for all $i = 1, \ldots, k$, $\lambda_i > 0$ and $B(c_i^+, c_i^-) = 0$. Now, let $m$ be an arc of $\mathcal{G}$ not incident to $c$. Then $(c_m^+, c_m^-) = (0, 0)$. Furthermore, $\sum_{i=1,\ldots,k} \lambda_i ((c_i^+)_m, (c_i^-)_m) = (0, 0)$ if and only if no circuit $c_i$ in the summation is incident to arc $m$. Hence, all arcs that are incident to one of the $c_i$ must also be incident to $c$. Since $c$ is an elementary circuit and the $c_i$ are circuits, this means that all these circuits must be a multiple of $c$. Therefore, we
conclude that \( c \) is an extremal ray of \( L \) and, hence, that all elementary circuits of \( \mathcal{G} \) are extremal rays of \( L \).

This leaves us to show that any solution to \( B(y_1 - y_2) = 0, \ (y_1, y_2) \geq 0 \) can be written as a positive combination of the extremal rays \( \{ (e_i, e_i) | i = 1, \ldots, M \} \) and \( \{ (e^+, c^-) | c \ \text{elementary circuit} \} \). To this end, let \( (y_1, y_2) \) be such solution. Write \( d = y_1 - y_2 \) and note that there exist \( z \in \mathbb{R}^M_+ \) such that we can write
\[
\begin{align*}
y_1 &= d^+ + z \\
y_2 &= d^- + z,
\end{align*}
\] (5.31)

Thus, we are done once we have shown that \( (d^+, d^-) \), which is also a solution, can be written as a positive combination of the extremal rays in \( \{ (e^+, c^-) | c \ \text{elementary circuit} \} \). Let \( d = d^+ - d^- \) and define
\[
m = \arg\min \{ |d_k| \ | k = 1, \ldots, M, \ d_k \neq 0 \}.
\] (5.32)

Since in each vertex of \( \mathcal{G} \), \( d \) preserves flow, i.e. \( Bd = 0 \), and since \( d_m \) has minimal flow, there exists an elementary circuit \( g \), a number \( \alpha \), and a circuit \( \tilde{d} \), such that
\[
(d^+, d^-) = |d_m| (g^+, g^-) + \alpha(d^+, d^-)
\] (5.33)
and
\[
d_m = 0.
\] (5.34)

By recursively applying this to \( \tilde{d} \) we arrive at the result. Note that, since in each step at least one arc is eliminated, the recursion goes at most \( M \) levels deep. \( \square \)

**Proposition 5.2** Assume that \( l < u \). If the value of the linear program \( P \) stated in (5.23) is \( 0 \), then there are no critical elementary circuits of \( \mathcal{G} \), and, if its value is less than \( 0 \), then the simplex method returns an extremal ray \( (e^+, c^-) \) for which \( c = c^+ - c^- \) is a critical elementary circuit of \( \mathcal{G} \).

**Proof** If the value of \( P \) is \( 0 \), then for all elementary circuits \( e \) of \( \mathcal{G} \) we have
\[
u^T c^+ - l^T c^- \geq u^T c^+ - l^T c^- \geq 0,
\] (5.35)
and thereby establishing the absence of critical circuits.

On the other hand, if the value of \( P \) is strictly negative, the simplex method returns an extremal ray \( (e^+, c^-) \) for which \( u^T c^+ - l^T c^- < 0 \). Since by assumption \( u_m - l_m \geq 1 \), we find that \( u^T c_i - l^T c_i \geq 1 - 2\delta > 0 \) for all \( i = 1, \ldots, M \). Hence, \( c^+ - c^- \) is an elementary circuit of \( \mathcal{G} \) for which
\[
0 \leq u^T c^+ - l^T c^- \leq (N + 1)\delta
\] (5.37)
\[
< (N + 1)\delta
\] (5.38)
\[
< 1.
\] (5.39)

Then, using the integrality of \( l, u, \) and \( c \), we see that
\[
u^T c^+ - l^T c^- = 0.
\] (5.40) \( \square \)
5.3 A small example

In this section we give a small example of an FDP instance to illustrate the theory developed in Section 5.2. In the example \( G \) consists of 8 vertices and 10 arcs \((N = 7, M = 10)\). The upper part of Figure 5.3 shows \( G \) together with the intervals on each arc.

There are 12 elementary circuits (we make distinction between a circuit \( c \) and its negative counterpart \(-c\)) and they are listed in Table 5.1. In that table for each circuit \( c \) we also list the value of \( u^T c^+ - l^T c^- \) in the second column. Note that by virtue of Lemma 5.1 this FDP instance is feasible if and only if \( u^T c^+ - l^T c^- \geq 0 \) for each of the 12 elementary circuits. For example, for circuit \( d \) we find (starting in vertex \( \tau_1 \) and running clock-wise) that

\[
u^T d^+ - l^T d^- = 4 - 0 - 0 + 5 - 0 - 0 = 9 \tag{5.41}\]

and for \(-d\) we have (running anti clock-wise starting in \( \tau_1 \)) that

\[
u^T (-d)^+ - l^T (-d)^- = 1 + 1 - 3 + 1 + 2 - 2 = 0 . \tag{5.42}\]

Hence, \(-d\) is a critical elementary circuit and \( d \) is not critical. Note that

\[
u^T (-d)^+ - l^T (-d)^- = -(l^T d^+ - u^T d^-) , \tag{5.43}\]

so that we could also have done the calculation for \(-d\) by running clock-wise (starting in \( \tau_1 \)) and summing \( 2 - 2 - 1 + 3 - 1 - 1 = 0 \) and reversing the sign.

We see that \(-d\) and \(-f\) are the only critical circuits because only those two circuits have a zero in the second column and there are no others zero's. Hence, we can use either of them to substitute out some implicit equalities, say we choose \(-f\). Starting in \( \tau_2 \), \(-f\) traverses the third and fourth arc in positive direction. Hence, we have the equalities

\[
\begin{align*}
\tau_2 - \tau_3 &= 2 \\
\tau_3 - \tau_4 &= 1 .
\end{align*} \tag{5.44}\]

The first and second arc are traversed in negative direction, yielding

\[
\begin{align*}
\tau_2 - \tau_5 &= 0 \\
\tau_5 - \tau_4 &= 3 .
\end{align*} \tag{5.45}\]

Then we may use the equalities coming from the second, third and fourth arc in \(-f\) to store the relations between variables \( \tau_2, \tau_3, \tau_4, \) and \( \tau_5 \), and use the equality

\[
\tau_2 - \tau_5 = 0 \tag{5.46}\]

to identify vertex \( \tau_2 \) and \( \tau_5 \) in \( G \), thereby contracting \( G \) to the second graph \( \hat{G} \) of Figure 5.3.

We make two remarks before we proceed. First, the equalities induced by elementary circuit \(-f\) can be understood as follows. By inspecting its first two arcs (again run anti clock-wise starting in \( \tau_2 \)), we see that any value for \( \tau_2 \) must be at least \( 3 + 0 = 3 \) bigger than a value for \( \tau_4 \). Similarly we find that any value for \( \tau_2 \) can be at most \( 1 + 2 = 3 \) bigger
Figure 5.3: Sequence of graphs of an FDP instance.
<table>
<thead>
<tr>
<th>Circuit</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>( (\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7) )</td>
<td>14</td>
</tr>
<tr>
<td>-a</td>
<td>( (\tau_0, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, \tau_7) )</td>
<td>4</td>
</tr>
<tr>
<td>b</td>
<td>( (\tau_0, \tau_1, \tau_2, \tau_5, \tau_6, \tau_7) )</td>
<td>9</td>
</tr>
<tr>
<td>-b</td>
<td>( (\tau_0, \tau_7, \tau_6, \tau_5, \tau_2, \tau_1) )</td>
<td>7</td>
</tr>
<tr>
<td>c</td>
<td>( (\tau_0, \tau_1, \tau_6, \tau_7) )</td>
<td>6</td>
</tr>
<tr>
<td>-c</td>
<td>( (\tau_0, \tau_7, \tau_6, \tau_1) )</td>
<td>5</td>
</tr>
<tr>
<td>d</td>
<td>( (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6) )</td>
<td>9</td>
</tr>
<tr>
<td>-d</td>
<td>( (\tau_1, \tau_6, \tau_5, \tau_4, \tau_3, \tau_2) )</td>
<td>0  ← critical</td>
</tr>
<tr>
<td>e</td>
<td>( (\tau_1, \tau_2, \tau_5, \tau_6) )</td>
<td>4</td>
</tr>
<tr>
<td>-e</td>
<td>( (\tau_1, \tau_6, \tau_5, \tau_2) )</td>
<td>3</td>
</tr>
<tr>
<td>f</td>
<td>( (\tau_2, \tau_3, \tau_4, \tau_5) )</td>
<td>8</td>
</tr>
<tr>
<td>-f</td>
<td>( (\tau_2, \tau_3, \tau_4, \tau_5) )</td>
<td>0  ← critical</td>
</tr>
</tbody>
</table>

**in first modification:**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>( (\tau_0, \tau_1, \tau_6, \tau_7) )</td>
<td>6</td>
</tr>
<tr>
<td>-c</td>
<td>( (\tau_0, \tau_7, \tau_6, \tau_1) )</td>
<td>5</td>
</tr>
<tr>
<td>g</td>
<td>( (\tau_0, \tau_1, \tau_5, \tau_6, \tau_7) )</td>
<td>9</td>
</tr>
<tr>
<td>-g</td>
<td>( (\tau_0, \tau_7, \tau_6, \tau_5, \tau_1) )</td>
<td>4</td>
</tr>
<tr>
<td>h</td>
<td>( (\tau_1, \tau_5, \tau_6) )</td>
<td>4</td>
</tr>
<tr>
<td>-h</td>
<td>( (\tau_1, \tau_6, \tau_5) )</td>
<td>0  ← critical</td>
</tr>
</tbody>
</table>

**in second modification:**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>( (\tau_0, \tau_6, \tau_7) )</td>
<td>6</td>
</tr>
<tr>
<td>-k</td>
<td>( (\tau_0, \tau_7, \tau_6) )</td>
<td>4</td>
</tr>
</tbody>
</table>

*Table 5.1: Elementary circuits in the process of reducing the FDP instance.*
Figure 5.4: Original FDP instance with intervals slightly perturbed to detect a critical circuit.
than a value for \( \tau_4 \). Hence, for the first arcs the intervals must hold at their lower bounds and for the last two arcs the intervals are fixed at their upper bounds, thereby establishing four equalities.

Our second remark concerns the algorithm we stated for detecting that \(-d\) or \(-f\) are critical. By making each interval a bit tighter by lowering the upper bounds by

\[
\delta = \frac{1}{N + 2} \approx 0.1
\]  

and raising the lower bounds by the same amount, we arrive at the slightly modified FDP instance depicted in Figure 5.4 with circuits and numbers \( u^+ c^+ \) and \( u^- c^- \) written underneath. Since the latter value is exactly \( \delta \) times the number of arcs in the circuit smaller than the former value and since this deficit is smaller than 1, we see that an elementary circuit in the original instance can be detected by looking at the signs in the third column of the table in Figure 5.4, or equivalently looking for an extremal ray of the feasible region of the program (5.23) with negative value.

Proceeding with \( \tilde{G} \) we list the elementary circuits again. This is also done in Table 5.1. This time \(-h\) is the only critical circuit, yielding

\[
\begin{align*}
\tau_5 - \tau_1 &= 2 \\
\tau_5 - \tau_6 &= 1 \\
\tau_6 - \tau_1 &= 1
\end{align*}
\]  

Using the latter equality we can contract \( \tilde{G} \) to end up with a reduced FDP instance with its graph depicted in the lower part of Figure 5.3. Note that the \([1, 2]\) interval in this graph is the sum of the \([0, 1]\) interval between \( \tau_0 \) and \( \tau_1 \) in \( \tilde{G} \) and 1, the latter number stemming from the equality we used to contract \( \tilde{G} \).

The final graph of Figure 5.3 has only two elementary circuits, both of which are not critical, so that there are no further implicit equalities. Note that we could have arrived at this system in a single step by choosing critical circuit \(-d\) in the original graph.

Anyway, after substituting out all implicit equalities we end up with the reduced FDP instance

\[
\begin{align*}
0 &\leq \tau_6 - \tau_7 \leq 1 \\
1 &\leq \tau_6 - \tau_0 \leq 2 \\
-4 &\leq \tau_7 - \tau_0 \leq 4 \\
\tau_0 &= 0
\end{align*}
\]  

the solutions of which are the corner points of the shaded area in Figure 5.5. Clearly, a random walk as defined in Section 5.2 is irreducible, whereas for the original FDP instance the walk would not work (for example the walk could never increase or decrease \( \tau_4 \) by one without changing the values of the other variables as well).
5.4 Computational complexity

In this section we study the performance of the almost uniform generator by considering the number of steps that \( \mathcal{W} \) needs to make in order to approach the uniform distribution close enough. We do this by examining two examples, one of which we use to show that \( \mathcal{W} \) does not mix rapidly, i.e. the almost uniform generator does not run in polynomial time. The other one is meant to show that such bad behavior is not likely to occur in practice. Note that from the point of view of computational complexity ‘computation time’ and ‘number of steps’ are equivalent because each step takes polynomial computation time. That is, in each step a random number must be sampled and at the most \( 2M \) linear inequalities have to be checked.

Let \( p_{\tau, \tilde{\tau}}^{(t)} \) be the probability that \( \mathcal{W} \) is in \( \tilde{\tau} \) after \( t \) steps when starting in \( \tau \). In compliance with the definition of approximation within some ratio, we apply the ratio measure \( \Delta(t) \), defined by

\[
\Delta(t) = \min\{ r \geq 0 \mid \# K^{-1} (1 + r)^{-1} \leq p_{\tau, \tilde{\tau}}^{(t)} \leq \# K^{-1} (1 + r) \text{ for all } \tau, \tilde{\tau} \in K \}, \tag{5.51}
\]

to quantify the distance between the \( t \)-step probability distribution of the state of \( \mathcal{W} \) and the uniform distribution. Given this measure and tolerance \( \epsilon \), we consider the quantity

\[
T_{\Delta}(\epsilon) = \min\{ t \in \mathbb{Z}_+ \mid \Delta(t) \leq \epsilon \}, \tag{5.52}
\]

which is the minimum number of steps to take in order to assure that, no matter the starting point of \( \mathcal{W} \), all \( \tau \in K \) have become about equiprobable. \( \mathcal{W} \), then, is said to be rapidly mixing if and only if \( T_{\Delta}(\epsilon) \) can be bounded from above by a polynomial in the input length (determined by \( N, M, l, \) and \( u \)) and \( \ln(\epsilon^{-1}) \). To use the theory in [128, 130] in order to analyze this bound it is convenient to relate the ratio measure to the relative pointwise distant measure \( \Psi(t) \), which is defined by

\[
\Psi(t) = \max\{ \frac{|p_{\tau, \tilde{\tau}}^{(t)} - \# K^{-1}|}{\# K^{-1}} \mid \tau, \tilde{\tau} \in K \}. \tag{5.53}
\]
Figure 5.6 : An example for which the random walk does not mix rapidly.

It is not difficult to see that

$$\Psi(t) \leq \Delta(t) \leq 2\Psi(t) \quad (5.54)$$

if $$\Psi(t) \leq \frac{1}{8}$$. Like for the ratio measure we define

$$T_\Psi(\epsilon) = \min\{ t \in \mathbb{Z}_+ \mid \Psi(t) \leq \epsilon \} \quad (5.55)$$

as the number of steps $$W$$ needs to take to get close to the uniform distribution.

By means of our first example in this section we show that, in general, $$T_\Psi(\epsilon)$$ cannot be bounded from above by a polynomial in the input size and $$\ln(\epsilon^{-1})$$. Thereafter we argue on the basis of another example that in practice for most FDP instances the random walk will mix rapidly.

Consider two sets $$X$$ and $$Y$$, both consisting of $$N$$ 0-1 variables, and define a partial order on $$X \cup Y$$ such that (a) all variables in $$X$$ are incomparable, (b) all variables in $$Y$$ are incomparable, and (c) each variable in $$Y$$ is at least as big as each variable in $$X$$, see Figure 5.6 where the arcs denote the order relation. We want to (almost) uniformly generate assignments that are consistent with the partial order. This problem is easily converted into the problem of uniformly generating solutions to the system

$$\begin{cases} 
0 \leq B^T \tau \leq \epsilon \\
0 \leq \tau - \tau_0 \epsilon \leq \epsilon \\
\tau_0 = 0 \end{cases} \quad (5.56)$$

where $$B$$ is the vertex-arc incidence matrix of the bipartite graph shown in Figure 5.6. The last two constraints reflect that this graph is extended with arcs from an additional vertex
(with index 0) to each of the $2N$ vertices in $X \cup Y$, thereby enforcing $0 \leq \tau_i \leq 1$ for all $i = 1, \ldots, 2N$. For this example we have

$$K = K_X \cup K_Y \cup \{\tilde{\tau}\} ,$$

where

$$K_X = \{ \tau \in \{0,1\}^{2N} \mid \tau \neq \tilde{\tau}, \ \tau_i = 1, \ i = N + 1, \ldots, 2N \} ,$$

$$K_Y = \{ \tau \in \{0,1\}^{2N} \mid \tau \neq \tilde{\tau}, \ \tau_i = 0, \ i = 1, \ldots, N \} ,$$

and

$$\tilde{\tau} = (0, e) ,$$

with $0, e \in \mathbb{Z}^N$. For sake of brevity we omitted $\tau_0$ in $\tau$. Recall that $(0, e) = (0^T, e^T)^T$ by convention. Furthermore, in the state transition graph $\tilde{\tau}$ is a cut vertex and both $K_X$ and $K_Y$ have $2^N - 1$ elements. Hence, starting in $K_X$, the random walk $W$ will reside an exponential time in $K_X$ until it jumps to $K_Y$. As a result $W$ does not mix rapidly.

More formally, the probability $\Phi(S)$ that, conditioned on $W$ being in $S \subseteq K$, to move out of $S$ in a single step reads

$$\Phi(S) = \frac{\sum_{\tau \in S, \tau \in K \setminus S} \pi_{\tau} P_{\tau, \tau}}{\sum_{\tau \in S} \pi_{\tau}}$$

$$= \frac{\sum_{\tau \in S, \tau \in K \setminus S} P_{\tau, \tau}}{\#S}$$

$$= \frac{1}{8N} \frac{\#(S, K \setminus S)}{\#S} ,$$

where $\pi_{\tau} = \#K^{-1}$ for all $\tau \in K$ and $(S, K \setminus S)$ is the set of undirected arcs between states in $S$ and $K \setminus S$ in the state transition graph. This conditional probability is known as the conductance of $S$ [71, 128, 130] and yields a measure for the time it takes to ‘escape’ from $S$ once $W$ gets there. Thus, the larger

$$\Phi = \min \{ \Phi(S) \mid S \subseteq K, \ #S \leq \frac{1}{2} \#K \} ,$$

the more rapid $W$ mixes. In [128, 130] it is shown that, if $\Phi \leq \frac{1}{2}$, which is generally the case, then

$$(1 - 2\Phi)^t \leq \Psi(t) \leq \#K(1 - \frac{1}{2} \Phi^2)^t .$$

In our example we calculate

$$\Phi(K_X) = \frac{1}{8N} \frac{\#(K_X, K_Y \cup \{\tilde{\tau}\})}{\#K_X}$$

$$= \frac{1}{8(2^N - 1)}.$$
and it follows that
\[ \Phi \leq \frac{1}{8(2^N - 1)} , \] (5.68)
so that
\[ \Psi(t) \geq (1 - \frac{2}{8(2^N - 1)})^t . \] (5.69)
Hence,
\[ T_\Psi(\epsilon) \geq 2^N \ln(\epsilon^{-1}) , \] (5.70)
thereby establishing exponential running time of the almost uniform generator in the worst case.

For this special example we could circumvent the exponential run time of the uniform generator by adding some additional step directions. More precisely, given that \( W \) is in a state \((0, \tau) \in \{0, 1\}^{2N}\), for some \( \tau \in \{0, 1\}^N \), we could allow \( W \) to step directly to state \((\tau, \epsilon) \in \{0, 1\}^{2N}\) and vice versa. This walk will mix rapidly because it can be seen as a random walk on the corners of the unit hypercube of dimension \( N \) where each corner \( \tau \) represents both the states \((0, \tau)\) and \((\tau, \epsilon)\). From [39] we know that this walk converges in \( O(N^2(\ln(\epsilon^{-1}) + N)) \) time. Of course, adding these specific steps does not repair the general problem of exponential running time and it seems unlikely that there exists a universal set of steps of polynomial size for which any random walk mixes rapidly.

In [106] the same example is mentioned. There, exact uniform generation is considered by coupling several Markov chains. More precisely, to generate uniformly at random exactly, in the approach described in [106], two random walks are simulated simultaneously, one starting in \((0, 0)\) and the other one starting in \((\epsilon, \epsilon)\). The simulations stop whenever the states of the two walks are the same. It can be shown that the coalescent state, then, is an exact uniform sample. However, like in our case, this approach requires that the Markov chains be simulated for an exponential number of steps. See also Chapter 2.

Although the exponential running time of the random walk may discourage the use of the uniform generation algorithm, we find that bad behavior is not likely to occur in practice. To support this claim on theoretical ground we give an example for which the convex hull of \( K \) (recall that this hull is integral) is needle-like, so that at first sight convergence of the random walk is expected to be slow, yet on second thought turns out to be relatively fast.

Consider the FDP instance shown in Figure 5.7. It is not difficult to verify that there are \( PN + 1 \) solutions to the corresponding system of inequalities and that the state transition graph constitutes a linear chain of these states. Hence, the random walk is a 2-dimensional random walk with reflecting barriers on both sides. For \( \Psi(t) \) we have
\[ \Psi(t) = \max\{ \sum_{\tau=1}^{PN} \frac{x_j^{(\tau)}y_k^{(\tau)} \cos(t(\pi r(PN + 1)))}{1 - \cos(\pi r(PN + 1))} \mid j = f(\tau), k = f(\hat{\tau}), \tau, \hat{\tau} \in K \} , \] (5.71)
where \( f \) maps a solution \( \tau \in K \) to its position, numbered from 1 to \( PN + 1 \), in the chain that the state transition graph forms (states 1 and \( PN + 1 \) are the reflecting barriers), and
where
\[ x_i^{(r)} = y_i^{(r)} = \sin(\pi r (PN + 1)) - \sin(\pi r(i - 1) (PN + 1)), \]
for \( r = 1, \ldots, PN \) and \( i = 1, \ldots, PN + 1 \), see [47]. From this formula it is a somewhat cumbersome exercise to bound \( T_\psi(\epsilon) \). Fortunately, to show that \( T_\psi(\epsilon) \) can be bounded from above by a polynomial in \( N \) and \( \ln(\epsilon^{-1}) \), we can also study the conductance \( \Phi \) of the state transition graph, like with the previous example. Obviously, the subset \( S \) of \( K \) that has the smallest conductance consists of the first \( \lfloor (PN + 1)/2 \rfloor \) states of the state transition graph. That is, \( S \) contains almost half of the solutions in \( K \) that are on one side of the chain. Then
\[ \frac{1}{2N(PN + 1)} \leq \Phi(S) \leq \frac{1}{2PN^2}, \]
so that
\[ T_\psi(\epsilon) \leq \left( 9N^2(PN + 1)^2 - 1 \right) \left( \ln(PN + 1) + \ln(\epsilon^{-1}) \right). \]
Taking that \( P \leq 59 \) in practice, we see that \( T_\psi(\epsilon) = \mathcal{O}(N^4(\ln(N) + \ln(\epsilon^{-1}))) \) with a small constant. This is not bad at all given the needle-like shape of the convex hull of \( K \).

To speed up the convergence of \( \mathcal{W} \) it seems better to take larger steps than just steps of unit size because this decreases the diameter of the state transition graph (i.e. the longest shortest path between any two states) and increases the maximum vertex degree with factor \( P \) a most. In this walk, which is called the walk with Rook’s moves in [8], a step is made by selecting a direction at random and move to a feasible integral point along that direction such that all feasible points on that line segment are equiprobable. In fact it is the discrete version of the hit-and-run algorithm, see [16]. It is easily verified that a-periodicity, irreducibility and symmetry are preserved by this modification. However,
introducing Rook’s moves does not always improve the mixing rate, witness our previous example where Rook’s moves coincide with the original set of steps.

5.5 Counting timetables

We conclude this chapter by describing an algorithm for approximating the number $\#K$ of elements of the set of solutions $K$ to an FDP instance with parameters $B \in \{0, \pm 1\}^{(N+1) \times M}$, $l$, $u$, and $G$. As to the complexity of this problem we have

**Proposition 5.3** The counting problem for the FDP is $\#P$-complete.

**Proof** The problem of counting the number of satisfying assignments to implicative boolean formulae, which is known to be $\#P$-complete [83], reduces to the counting problem for the FDP. To see this, let $x$ and $y$ be two boolean variables in $\{0,1\}$ and assume that $x \Rightarrow y$ is part of the problem description of a particular instance of the satisfiability problem. Reducing this to an FDP instance can be done by including the system

$$
\begin{align}
0 &\leq \tau_y - \tau_x \leq 1 \\
0 &\leq \tau_x - \tau_0 \leq 1 \\
0 &\leq \tau_y - \tau_0 \leq 1 \\
\tau_0 &= 0
\end{align}
$$

(5.75)

where $\tau_x$ and $\tau_y$ correspond to $x$ and $y$, respectively, and where $\tau_0$ can be re-used to translate the other implications in the same way. □

This negative result motivates the search for approximate counting algorithms. An algorithm that, given $k$, $\epsilon > 0$, and $0 < \delta < 1$, outputs a number $\#K$ that with probability at least $1 - \delta$ approximates $\#K$ within ratio $1 + \epsilon$ is called a randomized approximation scheme for $\#K$.

In [68] it is shown that for self-reducible problems [68, 119] randomized approximate counting is polynomially equivalent to almost uniform generation. Here we use the proof of that result to devise a randomized approximation scheme that makes $O(P^{3N^2 \epsilon^{-2} \ln(\delta^{-1})})$ calls to an almost uniform generator with sufficiently small tolerance to approximate $\#K$ within ratio $1 + \epsilon$ with probability at least $1 - \delta$, where $P$ is the width of the widest interval among all arcs of $G$.

Suppose an almost uniform generator $U$ is given, say as an oracle. We state an algorithm that approximates $\#K$ within ratio $1 + \epsilon$ with probability at least $\frac{3}{4}$. Then, repeating this algorithm $1 + 12[\ln(1\delta)]$ times and taking the median of the results as the final outcome yields the desired randomized approximation scheme for $\#K$.

We make $t$ calls to $U$ with a small tolerance and consider the $i$-th coordinate of the $t$ outputs. Let $m$ be the value that is most frequently observed, say $\alpha$ times. Then we can estimate $\#K$ by equating it (approximately) to $t\alpha^{-1}\#K$, where $\#K$ is the number of solutions to $\{\tau \in \mathbb{Z}^{N+1} | l \leq B^T \tau \leq u, \tau_0 = 0, \tau_i = m\}$. Hence, by way of iteration (recall the self-reducibility property) we get an estimate of $\#K$. 
To state the counting algorithm more precisely, let $0 < \epsilon, \delta < 1$ and a positive integer $t$ be given and initialize by setting
\[
\begin{aligned}
C_{n-1} &= 1 \\
l_n &= l \\
u_n &= u \\
B_n &= B \\
G_n &= G \\
K_n &= K \\
n &= 1,
\end{aligned}
\] (5.76)

We assume that $G_n$ has no critical circuits. Then, all vertices of $G_n$ are free in the sense of Definition 5.2. If this assumption is false, then just transform the system $l_1 \leq B_1^T \tau \leq u_1$ to a reduced system without critical elementary circuits, using the procedure described in Section 5.2. As long as $K_n$ is not a singleton keep repeating the following steps recursively.

Choose an index $i$ in $\{1, \ldots, N\}$ and define the numbers
\[
\begin{aligned}
d^+ = \max \{ \tau_i | \tau \in K_n \} \\
d^- = \min \{ \tau_i | \tau \in K_n \}.
\end{aligned}
\] (5.77)

That is, $\tau_i$ can take values in the range $d^-, \ldots, d^+$ and only in this range. Make $t$ calls to $U$ with tolerance $\epsilon/9N$, resulting in the tabulates $\tau^1, \ldots, \tau^t$. For $j = d^-, \ldots, d^+$ let
\[
X_j = \frac{1}{t} \# \{ h = 1, \ldots, t | \tau_i^h = j \}
\] (5.78)
and compute
\[
k = \arg\max \{ X_j | j = d^-, \ldots, d^+ \}.
\] (5.79)

Among the $t$ tabulates $X_j$ is fraction of those tabulates for which vertex $\tau_i$ has value $j$ and $X_k$ maximizes these fractions over the range $d^-, \ldots, d^+$. Then, set
\[
C_n = \frac{1}{X_k} C_{n-1},
\] (5.80)

increase $n$ by 1, and construct $l_n, u_n, B_n, G_n$, and $K_n$ by substituting $\tau_i = k$ in $l_{n-1} \leq B_{n-1}^T \tau \leq u_{n-1}$. Reduce it to a system without critical elementary circuits (and adjust $N$) and move over to the next iteration. If $K_n$ is a singleton, then $C_{n-1}$ approximates $\#K$ within ratio $\epsilon$ with probability at least $\frac{3}{4}$. We have

**Proposition 5.4** Let $0 < \epsilon, \delta < 1$ and let $U$ be an almost uniform generator with tolerance $\epsilon/9N$. Then, by applying the algorithm $1 + 12[\ln(\frac{1}{\delta})]$ times and taking the median of the results as the outcome we have a randomized approximation scheme for $\#K$ that needs to make at most
\[
36 \left( 1 + 12[\ln(\delta^{-1})] \right) \frac{(PN + 1)^3 N^4}{\epsilon^2}
\] (5.81)
calls to $U$. 
**Proof** We give a proof along the lines of [68] and use the definitions and setting as in the above. Hence, we have an index $i$ in $\{1, \ldots, N\}$ and define $d^+$ and $d^-$ accordingly. In addition we define

$$P = \max\{u_m - l_m \mid m = 1, \ldots, M\} \quad (5.82)$$

and

$$d = d^+ - d^- + 1 \quad (5.83)$$

and set

$$t = 36d^3N^3\epsilon^{-2} \quad (5.84)$$

Note that

$$d \leq PN \quad (5.85)$$

because the diameter of $G$ is not larger than $N$. Furthermore, let an almost uniform generator $U$ be given by way of an oracle.

Make $t$ calls to $U$ with tolerance $\epsilon/9N$ and name the outcomes $\{r^1, \ldots, r^t\}$. Define for $j = d^-, \ldots, d^+$ the stochastic variables

$$X_j = \frac{1}{t} \# \{ h = 1, \ldots, t \mid r^h = j\} \quad (5.86)$$

and their means

$$\mu_j = E(X_j) \quad (5.87)$$

Since $X_j$ is the average of $t$ independent 0-1 random variables we have

$$\text{var}(X_j) \leq \frac{1}{4t} \quad (5.88)$$

and, hence, by virtue of Chebyshev's inequality,

$$Pr(\ |X_j - \mu_j\ | \leq \frac{\epsilon}{4dN} ) \geq 1 - \frac{1}{4t} \left(\frac{4dN}{\epsilon}\right)^2$$

$$= 1 - \frac{1}{9dN} \quad (5.89)$$

implying

$$Pr(\ |X_j - \mu_j\ | > \frac{\epsilon}{4dN} ) \leq \frac{1}{9dN} \quad (5.90)$$

Let

$$k = \arg\max\{ X_j \mid j = d^-, \ldots, d^+ \} \quad . \quad (5.91)$$

Then,

$$Pr(\ |X_k - \mu_k\ | \leq \frac{\epsilon}{4dN} ) \geq Pr(\ |X_j - \mu_j\ | \leq \frac{\epsilon}{4dN} \text{ for all } j = d^-, \ldots, d^+ ) \quad (5.92)$$

$$\geq Pr(\ |X_j - \mu_j\ | \leq \frac{\epsilon}{4dN} ) \quad (5.93)$$
\[
= 1 - \Pr( |X_j - \mu_j| > \frac{\epsilon}{4dN} \text{ for some } j = d^-, \ldots, d^+ ) \quad (5.95)
\]
\[
\geq 1 - \sum_{j=d^-, \ldots, d^+} \Pr( |X_j - \mu_j| > \frac{\epsilon}{4dN} ) \quad (5.96)
\]
\[
\geq 1 - \frac{d}{9dN} \quad (5.97)
\]
\[
= 1 - \frac{1}{9N} \quad (5.98)
\]

By the pigeon hole principle we have
\[
X_k \geq \frac{1}{d}, \quad (5.99)
\]

thus yielding
\[
\Pr( \frac{|X_k - \mu_k|}{X_k} \leq \frac{\epsilon}{4N} ) \geq \Pr( \frac{|X_k - \mu_k|}{X_k} \leq \frac{\epsilon}{4Nd} ) \quad (5.100)
\]
\[
\geq 1 - \frac{1}{9N} \quad (5.101)
\]

and, hence,
\[
\Pr( \mu_k(1 + \frac{\epsilon}{3N})^{-1} \leq X_k \leq \mu_k(1 + \frac{\epsilon}{3N}) ) \geq 1 - \frac{1}{9N}. \quad (5.102)
\]

Let
\[
\alpha_j = \frac{\# \{ \tau \in K | \tau_i = j \} }{\# K}, \quad (5.103)
\]
i.e. \( \alpha_j \) is the fraction of the timetables in \( K \) for which the \( i \)-th vertex has value \( j \). \( U \) was called with tolerance \( \epsilon/9N \), which implies that
\[
\alpha_k(1 + \frac{\epsilon}{9N})^{-1} \leq \mu_k \leq \alpha_k(1 + \frac{\epsilon}{9N}) \cdot (5.104)
\]

Combining this with (5.102) gives
\[
\Pr( \alpha_k(1 + \frac{\epsilon}{3N})^{-1}(1 + \frac{\epsilon}{9N})^{-1} \leq X_k \leq \alpha_k(1 + \frac{\epsilon}{3N})(1 + \frac{\epsilon}{9N}) ) \geq 1 - \frac{1}{9N}, \quad (5.105)
\]

which implies
\[
\Pr( \alpha_k(1 + \frac{\epsilon}{2N})^{-1} \leq X_k \leq \alpha_k(1 + \frac{\epsilon}{2N}) ) \geq 1 - \frac{1}{9N}. \quad (5.106)
\]

Clearly the algorithm needs at most \( N \) iterations like this, so that by executing the algorithm we end up with an approximation of \( \# K \) within ratio
\[
(1 + \frac{\epsilon}{2N})^p < 1 + \epsilon \quad (5.107)
\]
with probability at least
\[ (1 - \frac{1}{9N})^p > \frac{3}{4}, \]  
for some \( p \leq N \). Thus the algorithm reaches the desired result in at most
\[ t N = 36d^3 N^3 \epsilon^{-2} N \]  
\[ \leq 36(PN + 1)^3 N^4 \epsilon^{-2} \]
calls to \( \mathcal{U} \). Applying the Powering lemma as in [68], which means repeating the algorithm \( 1 + 12\lceil \ln(\frac{1}{\delta}) \rceil \) times and accepting the median of the respective results as the final outcome, gives the result. \( \Box \)
Part III

Miscellaneous
Chapter 6

Improved Solutions to the Steiner Triple Set Covering Problem

In this chapter\textsuperscript{1} we present improved covers for a set covering problem, introduced by Fulkerson et al. in 1974 [55], that arises from computing the 1-width of incidence matrices of Steiner triple systems. In turn, their work was based on that of Fulkerson and Ryser in the early sixties [52, 53, 54] who discussed the more general problem of computing widths and heights of 0-1 matrices. We will refer to the problem of computing the 1-width of incidence matrices of Steiner triple systems as the Steiner Triple Set Covering Problem (ST-SCP). Apart from presenting covers of lower cardinality than known before we also discuss the algorithm with which we obtained those results.

The natural 0-1 integer linear programming formulation of the ST-SCP is widely recognized as a particularly difficult problem, witness a result from Avis [9] in which he shows that, for a special class of problem instances, any branch and bound algorithm with linear programming relaxation and/or elimination by dominance requires that a super polynomial number of partial solutions must be inspected. Moreover, the results presented in this chapter form new evidence of its complexity.

The ST-SCP is of particular interest for benchmarking purposes [74], because the problem is hard and has yet a relatively low number of variables. Moreover, classes of instances can simply be generated [54, 55, 62] and, as we will see in Section 6.1, for some test problems optimal or good covers are known.

In linear programming format the ST-SCP adopts the form

\[
\begin{align*}
\min_{x} & \quad e^\top x \\
\text{s.t.} & \quad Ax \geq e \\
& \quad x \in \{0, 1\}^n ,
\end{align*}
\]  

(6.1)

where \(e\) is the all-one vector of appropriate size and \(A\) is an \(m \times n\) 0-1 matrix that has precisely three ones per row. In addition, for every pair of columns \(i\) and \(j\) there is exactly

\textsuperscript{1}This chapter is based on 'M.A. Odijk and H. van Maaren: Improved Solutions to the Steiner Triple Covering Problem', which will appear in Information Processing Letters. The authors are grateful to Joost Warners for bringing the Steiner Triple Set Covering Problem to their attention.
one row \( k \) with \( a_{ki} = a_{kj} = 1 \). For every row of \( A \) the set of column indices corresponding to the ones in that row is called a Steiner triple. Steiner triple systems are in fact a special case of block designs [62]. The optimal value of the problem (i.e. the smallest cover cardinality) is called the 1-width of \( A \).

As to the generation of problem instances, in [74] a recursive technique is described to generate problems for which \( n = 3^k \) and \( n = 15 \cdot 3^{k-1} \), where \( k = 1, 2, 3, \ldots \). That is, using seed matrices

\[
A_3 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}
\]

and

\[
A_{15} = \begin{bmatrix} Z & E & 0 \\ 0 & Z & E \\ E & 0 & Z \\ I & I & I \end{bmatrix},
\]

where \( I \) is the identity matrix of appropriate size and

\[
Z = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}
\]

and

\[
E = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}
\]

problems of size \( n = 45, 135, \ldots \) and \( n = 9, 27, 81, \ldots \) can be generated applying the recursion

\[
A_{3n} = \begin{bmatrix} A_n & 0 & 0 \\ 0 & A_n & 0 \\ 0 & 0 & A_n \\ I & I & I \\ \{ & \hat{A}_{3n} & \} \end{bmatrix},
\]
where \( \tilde{A}_{3n} \) consists of 6 blocks of rows corresponding to the permutations of \( \{1, 2, 3\} \). That is, if we index the columns of \( A_{3n} \) with \( \{ (i, j) \mid i = 1, \ldots, n, \ j = 1, 2, 3 \} \), then \( \{(i, r), (j, s), (k, t)\} \) is a triple of \( \tilde{A}_{3n} \) if and only if \( \{i, j, k\} \) is a triple of \( A_n \) and \( \{r, s, t\} = \{1, 2, 3\} \). Clearly, seed matrix \( A_3 \) is a Steiner triple system. \( A_{15} \) was introduced by Fulkerson and Ryser in 1963 [54]. All results in this chapter apply to the ST-SCP instances \( A_9, A_{15}, A_{27}, A_{45}, A_{81}, A_{135}, A_{243} \) that were generated using the above recursion. The problems can be retrieved from [103].

### 6.1 Known and new results

In general, for all \( n \equiv 1, 3 \mod 6 \), Steiner triple systems exist. \( A_9 \) is the unique system on 9 elements but for \( n = 15 \) already 80 nonisomorphic systems were found by Cole et al. [32]. Fulkerson and Ryser [54] asserted that all of them allow for a cover of size 9 or less and showed that at least one of them has a cover of size 7.

Let \( w_n \) be the optimal cover size for \( A_n \). Fulkerson and Ryser [54] also showed that \( \frac{1}{2}(n - 1) \) is a lower bound on the size of any cover of a Steiner triple system of order \( n \). From (6.6) it is not difficult to see that for our test problems this bound can be uplifted to \( \frac{3}{2}n \) for \( n \geq 27 \), because \( w_{27} = 18, w_{45} = 30, \) and \( w_{3n} \geq 3w_n \) for \( n = 3, 9, 15, 27, \ldots \). Avis [9] showed that \( n - 2^k \) is an upper bound on \( w_n \), where \( k = \log_3(n) \), for \( n = 3^k, k = 1, 2, 3, \ldots \).

For \( A_3, \ldots, A_{81} \) optimal covers are known. Fulkerson et al. [55] used a cutting plane method to solve \( A_9 \) and an implicit enumeration algorithm to solve \( A_{15} \) and \( A_{27} \). They failed, however, to solve \( A_{45} \), which was reported by Avis [9] to be solved by Ratliff in 1979.

For \( A_{81} \) a 61 cover was found in 1989 by Feo and Resende [48] using a randomized version of Chvátal's heuristic [30]. Recently, in 1995, Mannino and Sassano [87] found that this cover is indeed optimal by executing a branch and bound scheme.

A 105 cover for the \( A_{135} \) was found in 1991 by Karmarkar et al. [74] using an interior point approach. Recently, in 1995, Feo and Resende [49] found a 104 cover using a greedy randomized adaptive search procedure (grasp). Mannino and Sassano [87] also produced a 104 cover for the \( A_{135} \) with their branch and bound algorithm.

For the \( A_{243} \) in 1989 a cover of size 204 [48] and later, in 1995, a 203 cover [49] were found by Feo and Resende with their grasp. The 204 cover was also found by Karmarkar et al. [74] using the above-mentioned interior point approach. Mannino and Sassano [87] found a 202 cover in 1995 by executing their branch and bound algorithm.

The main contribution of this chapter is the presentation of a 103 cover for the \( A_{135} \) problem and a 198 cover for the \( A_{243} \) problem, see Table 6.1. Optimality of these covers is still an open problem. However, in Section 6.3 we will present reasons to believe that the 198 cover is optimal, thereby challenging Feo and Resende's conjecture that the \( A_{243} \) is beyond today's (1989) methods [48]. The indices in Table 6.1 correspond to the descriptions of \( A_{135} \) and \( A_{243} \) in [103]. Table 6.2 summarizes the above.

As to these new covers, we mention that Nonobe and Ibaraki [93] also found a 198 cover for the \( A_{243} \) in 1996, independently from us, and Nonobe [94] communicated in 1997 a 103 cover for the \( A_{135} \). They found their covers using a tabu search approach.
Zero's in the 103 cover for $A_{135}$

<table>
<thead>
<tr>
<th>2</th>
<th>3</th>
<th>6</th>
<th>9</th>
<th>11</th>
<th>14</th>
<th>20</th>
<th>30</th>
<th>31</th>
<th>34</th>
<th>37</th>
<th>38</th>
<th>42</th>
<th>43</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>80</td>
<td>90</td>
<td>91</td>
<td>94</td>
<td>97</td>
<td>98</td>
<td>102</td>
<td>103</td>
<td>110</td>
<td>120</td>
<td>122</td>
<td>123</td>
<td>126</td>
<td>129</td>
</tr>
<tr>
<td>131</td>
<td>134</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Zero's in the 198 cover for $A_{243}$

<table>
<thead>
<tr>
<th>3</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>16</th>
<th>27</th>
<th>28</th>
<th>34</th>
<th>41</th>
<th>43</th>
<th>57</th>
<th>58</th>
<th>60</th>
<th>61</th>
</tr>
</thead>
<tbody>
<tr>
<td>67</td>
<td>72</td>
<td>81</td>
<td>83</td>
<td>85</td>
<td>86</td>
<td>94</td>
<td>95</td>
<td>98</td>
<td>99</td>
<td>125</td>
<td>126</td>
<td>128</td>
<td>133</td>
<td>141</td>
</tr>
<tr>
<td>142</td>
<td>148</td>
<td>149</td>
<td>159</td>
<td>161</td>
<td>162</td>
<td>165</td>
<td>167</td>
<td>168</td>
<td>185</td>
<td>203</td>
<td>220</td>
<td>224</td>
<td>232</td>
<td>236</td>
</tr>
</tbody>
</table>

Table 6.1: Improved covers for the ST-SCP instances $A_{135}$ and $A_{243}$. The problems can be retrieved by anonymous ftp.

### 6.2 The algorithm

The algorithm that we used to search for the covers is based on Selman et al.'s GSAT algorithm for tackling difficult satisfiability problems [124]. The algorithm starts with a randomly selected vertex of the $n$-dimensional unit hypercube $H$ and proceeds by walking over the edges, thereby visiting a sequence of vertices of which the best one is saved. Such vertices can be seen as representations of solutions by taking a one as the presence of the corresponding element in the cover and a zero as an absence. By repeating this procedure a couple of times and taking the best result as final output, good covers for all problem instances up to $n = 243$ were found within a couple of minutes (seconds for the smaller problems).

The inner loop (i.e. a single walk starting at random) of the algorithm is stated in pseudo-code in Table 6.3. Lines 4-7 let the walk start in vertex $x$, which is randomly selected. Throughout, vertex $s$ is the current best vertex encountered during the walk covering $a$ rows of $A_n$ and containing $b$ zero's. Thus, initially, $s = x$.

After initialization the walk makes $t$ steps, each of which is described by the lines 9-23. A step is in fact a binary flip of one of the $n$ variables and can be made in two ways. With probability $1 - p$ a step is made to the best neighboring vertex of $H$ (two vertices are neighbors if they differ in exactly one variable). Comparison between two vertices is done on the basis of a partial ordering defined as follows. A vertex $x$ covering $a_x$ rows of $A_n$ and having $b_x$ zero's has bigger value than vertex $y$ covering $a_y$ rows of $A_n$ and having $b_y$ zero's if and only if either $a_x > a_y$ or if $a_x = a_y$ and $b_x > b_y$. In rows 20 and 22 this ordering is referred to as lexicography. According to line 20, if two or more neighbors of the current vertex $x$ have the best value among all $n$ neighbors, the tie is broken randomly. Note that a step is always made, even if this means moving to a vertex with lower value.
<table>
<thead>
<tr>
<th>problem</th>
<th>order/triples</th>
<th>cover size</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_9$</td>
<td>9/12</td>
<td>5</td>
<td>Fulkerson and Ryser 1963 [54] optimal</td>
</tr>
<tr>
<td>$A_{15}$</td>
<td>15/35</td>
<td>9</td>
<td>Fulkerson et al. 1974 [55] optimal</td>
</tr>
<tr>
<td>$A_{27}$</td>
<td>27/117</td>
<td>18</td>
<td>Fulkerson et al. 1974 [55] optimal</td>
</tr>
<tr>
<td>$A_{45}$</td>
<td>45/330</td>
<td>30</td>
<td>Ratliff 1979 (reported in [9]) optimal</td>
</tr>
<tr>
<td>$A_{81}$</td>
<td>81/1080</td>
<td>61</td>
<td>Feo and Resende 1989 [48] conjectured optimal 1991 (reported in [74]) proven optimal 1995 [87]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>104</td>
<td>Odijk and van Maaren 1996 [this thesis] Nonobe 1997 [94] not known to be optimal</td>
</tr>
</tbody>
</table>

Table 6.2: Problems and covers found.
Input: $A_n$, number $p \in [0, 1]$
Output: $n$-dimensional 0-1 vector

begin

$x \leftarrow$ randomly generated vertex of $H$

$a \leftarrow$ number of rows of $A_n$ covered by $x$

$b \leftarrow$ number of zero's of $x$

$s \leftarrow x$

for $j:=1$ to $t$ do

with probability $p$ do

$k \leftarrow$ random number from $1, \ldots, n$

$x \leftarrow x$ with $k$-th variable flipped

$c_k \leftarrow$ number of rows of $A_n$ covered by $x$

$d_k \leftarrow$ number of zero's of $x$

else do

for $i:=1$ to $n$ do

$z_i \leftarrow x$ with $i$-th variable flipped

$c_i \leftarrow$ number of rows of $A_n$ covered by $z_i$

$d_i \leftarrow$ number of zero's of $z_i$

end for

$k \leftarrow \text{argmax}\{c_i, d_i \mid i = 1, \ldots, n\}$ (random tie-break)

$x \leftarrow x$ with $k$-th variable flipped

if $(c_k, d_k) >_{\text{lex}} (a, b)$

then $(s, a, b) \leftarrow (x, c_k, d_k)$

end for

return $s$

end

Table 6.3: Inner loop of the algorithm that was used to find the new covers.
The other type of step the walk can make occurs with complementary probability $p$ and involves a 'blind step'. Such step is taken by moving to a neighboring vertex of $H$ at random. It is best to set $p$ to a low value, say 0.1. Allowing the walk to make blind steps is very important as it decreases the probability that the walk starts cycling seemingly.

### 6.3 Discussion

The strategy the algorithm follows is applied before by Selman et al. [124], under the name of the GSAT algorithm, to solve hard random 3-SAT problems that are an order of magnitude larger than such problems that can be solved by the classical Davis-Putnam algorithm [35]. However, the GSAT algorithm seeks to satisfy as much clauses as possible without bothering about the number of zero's or one's in the solutions, whereas we know beforehand that all 'clauses' (i.e. triples) can trivially be 'satisfied' (i.e. covered) leaving us to search for a solution with as few one's as possible (i.e. seek for a minimal cover size). For this purpose we modified the GSAT algorithm by introducing a lexicographical ordering of the solutions by first covering as many triples as possible and secondarily doing so with as few one's as possible.

It should be noted, however, that the GSAT algorithm is incomplete, whereas the Davis-Putnam algorithm truly decides about satisfiability. Our algorithm inherits this incompleteness in the sense that optimality of the best covers found remains undecided.

It is very hard to say why the concept of GSAT works so well. As pointed out in [124], a very important feature of the algorithm is that during a walk always precisely one variable of the current solution is flipped, even if this yields a worse solution. Thus, the algorithm is greedy, but not too greedy to get trapped in a local optimum. Furthermore, adding noise by way of blind steps adds to the effectiveness a great deal, witness empirical results in [124]. Indeed, from our own experiments we confirm these conclusions. Taking $p = 0.1$ is about optimal.

To further illustrate the effectiveness of the algorithm, we can take a look at Table 6.4 which shows the expected number of 'cover checks' required to hit an optimal cover (in case of the $A_{135}$ covers of size 104 were considered because covers of size 103 were found sporadic and in case of the $A_{243}$ covers of size 198 were considered because covers of lower cardinality were not encountered). With a cover check we refer to deciding about the covering of an individual triple by a given solution and we take the time to perform such a check as a unit amount of computation time.

Since each column of $A_n$ has exactly $1/2(n - 1)$ one's, flipping one variable of $x$ in the algorithm requires $1/2(n - 1)$ cover checks. Hence, the expected number of cover checks in a single iteration of the algorithm (during a particular walk) equals $1/2(n - 1)(p + (1 - p)n)$. Given that a single walk takes $t$ steps, the expected number of cover checks in a single walk reads $1/2(n - 1)(p + (1 - p)n)t$. Then, if with $h(t)$ we denote the probability that during a single walk consisting of $t$ steps we hit the target value stated in Table 6.4 in the third column, $1/h(t)$ is the expected number of walks we must make in order to hit this value. Therefore, given that each walk takes $t$ steps, the total number $\gamma(t)$ of expected
<table>
<thead>
<tr>
<th>problem</th>
<th>variables/</th>
<th>cover</th>
<th>$t_{opt}$</th>
<th>$\gamma(t_{opt})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>constraints</td>
<td>sizes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{15}$</td>
<td>15/35</td>
<td>9</td>
<td>3</td>
<td>464</td>
</tr>
<tr>
<td>$A_{27}$</td>
<td>27/117</td>
<td>18</td>
<td>16</td>
<td>8,149</td>
</tr>
<tr>
<td>$A_{45}$</td>
<td>45/330</td>
<td>30</td>
<td>400 (±50)</td>
<td>9,924,444</td>
</tr>
<tr>
<td>$A_{81}$</td>
<td>81/1080</td>
<td>61</td>
<td>460 (±20)</td>
<td>5,317,284</td>
</tr>
<tr>
<td>$A_{135}$</td>
<td>135/3015</td>
<td>104</td>
<td>1,450 (±50)</td>
<td>193,660,000</td>
</tr>
<tr>
<td>$A_{243}$</td>
<td>243/9801</td>
<td>198</td>
<td>2,100 (±100)</td>
<td>252,120,000</td>
</tr>
</tbody>
</table>

Table 6.4: Expected (scaled) runtimes of our algorithm for the test problems.

cover checks required to hit the target value reads

$$\gamma(t) = \frac{t(n-1)(p + (1-p)n)}{2h(t)}.$$  \hspace{1cm} (6.7)

In Table 6.4 $\gamma$ is displayed for $t_{opt}$, where $t_{opt}$ is given by

$$t_{opt} = \text{argmin}\{ \gamma(t) \mid t = 1, 2, \ldots \} .$$  \hspace{1cm} (6.8)

The target values are shown in the third column under ‘cover sizes’. The $t_{opt}$ were found empirically by performing for a range of values for $t$ many random walks, thereby estimating $t/h(t)$ for those values of $t$. For problems $A_{15}$ and $A_{27}$, 10,000 walks were simulated for $t = 1, \ldots, 100$ and for $A_{45}, A_{81}, A_{135}$ and $A_{243}$, 1,000 walks were simulated for another range of values of $t$. Noise parameter $p$ was set to 0.1.

It strikes immediately that either the $A_{45}$ is exceptionally difficult or the $A_{81}$ is exceptionally easy. This monotonicity breaking experience is yet another clue for the intrinsic complexity of the class of test problems we consider here. That is, even though these problems are recursively generated, their respective complexities do not behave monotone, at least not with respect to our algorithm. Moreover, considering that 103 covers for the $A_{135}$ are difficult to find, it seems that the problems $A_n$ with $n = 15 \cdot 3^{k-1}$ are harder than problems with $n = 3^k, k = 1, 2, \ldots$.

Finally, we mention the phenomenon that for the $A_{243}$ problem none of the covers of size 199 or 200 that we encountered while simulating the algorithm were local optima. Thus it seems as if with a nearest neighbor search we either find a cover of size 198 or a cover of size 201 or higher. Moreover, the 198 cover in Table 6.1 possesses a remarkable property. That is, there are 990 different pairings of the 45 zero’s and by inspection it was found out that each of the 198 one’s appears with precisely 5 of those pairs in a triple.
Furthermore, we have that $198 \cdot 5 = 990$ and, again by inspection, it appears that the 198 ones can be taken as representatives of a partitioning of the 990 pairs of zero's in classes of size 5 each. The same observation was made for all other covers of size 198 that were found. Therefore, we conjecture that the 198 cover is optimal.
Bibliography


Railway Timetable Generation (summary)

Building or reconstructing railway infrastructure may take ten to fifteen years. Large amounts of money are usually involved and infrastructure, once built, must be useful for a long time. Therefore, it is crucial to define building projects such that the resulting infrastructure is robust with respect to future timetables. However, these timetables are influenced by many uncertain factors of political, economical, demographical, or geographical nature.

This thesis contributes to the development of a novel concept of specifying railway infrastructure at railway stations which is meant to deal with the uncertainties about future timetables. It is based on the generation of many timetables given some presuppositions regarding future production. Specifying infrastructure using these timetables, then, makes the final specification timetable-independent to a certain extent and, hence, robust with respect to future developments. More specifically, in this thesis we focus on the generation of timetables.

Roughly speaking, the last step of the timetabling process comprises the construction of a timetable from a so-called timetable structure. Mathematically, a timetable structure is a set of constraints on arrival and departure times modeling various requirements a timetable must meet, e.g. headway times and connections between trains. We discuss this model and algorithms to solve it.

Since the uncertainties about the future timetable grow with every step in the process of preparing timetables, the need to randomize the steps in this process grows with the planning horizon. Put differently, when planning 15 years ahead, aggregated information hidden in an origin-destination matrix is less uncertain than a detailed object like a timetable and accordingly a level of randomness in timetable generation is required to reflect this properly. In particular, it is desired to generate timetables from timetable structures at random. Therefore, in this thesis we focus on random generation of timetables.

Random timetable generation proceeds in two phases. First, a number of classes of timetables from a given timetable structure are randomly selected according to a probability distribution that favors classes of timetables that are likely to contain robust timetables. Thereafter, from each selected class a number of timetables are generated uniformly at random for the purpose of uncertainty and sensitivity analysis. How to achieve such analysis is not a topic in this thesis. The random generation problem is very hard as deciding about the existence of a feasible timetable is already NP-complete.

Apart from the above, we also discuss the Steiner Triple Set Covering Problem, which is a set covering problem arising from computing the 1-width of 0-1 matrices. Specific instances of this problem have been proposed and used as benchmark problems for solvers of 0-1 programming problems. For the two largest problems we present improved covers using a simple algorithm that we will describe as well.
Generatie van Treindienstregelingen (samenvatting)

Het bouwen of reconstrueren van spoorweg-infrastructuur kan 10 tot 15 jaar in beslag nemen. Grote geldbedragen zijn hiermee gemoeid en de infrastructuur moet, eenmaal aangelegd, lange tijd meegaan. Derhalve is het van groot belang om bouwprojecten zo te definiëren dat de resulterende infrastructuur robuust is en aan ziekte van toekomstige dienstregelingen. Echter, deze dienstregelingen worden beïnvloed door vele onzeker factoren van politieke, economische, demografische, of geografische aard.

Dit proefschrift draagt bij aan de ontwikkeling van een nieuwe methode om spoorweginfrastructuur te specificeren voor stationsomgevingen. Deze methode beoogt om te gaan met de onzekerheden ten aanzien van toekomstige dienstregelingen en is gebaseerd op het genereren van veel dienstregelingen, gegeven enkele uitgangspunten voor de toekomstige productie. Het specificeren van infrastructuur op basis van deze dienstregelingen maakt de uiteindelijke specificatie in zekere zin dienstregeling-onafhankelijk en dus robuust met betrekking tot toekomstige dienstregelingen. Concreet, in dit proefschrift gaan we in op het genereren van dienstregelingen.

Grofweg de laatste stap in het proces om dienstregelingen te maken is het construeren van een dienstregeling uit een zogenaamde dienstregelingstructuur. Mathematisch is een dienstregelingstructuur een verzameling beperkingen op aankomst- en vertrek tijdstippen waarmee uiteenlopende eisen aan beoogde dienstregelingen gesteld kunnen worden, zoals bijvoorbeeld voldoende afstand tussen treinen of aansluitingen op stations. We bespreken dit model alsmede algoritmes om het op te lossen.

De onzekerheden ten aanzien van toekomstige dienstregelingen is groter naarmate de tijd waarover gepland wordt langer is. Om om te kunnen gaan met die onzekerheden groeit derhalve de behoefte aan randomisatie van de stappen in het proces om dienstregelingen te genereren ook met de planningshorizon. Met andere woorden, als er 15 jaar vooruit gepland wordt zal geaggregeerde informatie in de herkomst-bestemmingsmatrix minder onzeker zijn dan de informatie in een gedetailleerde dienstregeling. Een gepaste mate van randomisatie is dan ook nodig om dit te weerspiegelen. In het bijzonder is er veel randomisatie gewenst bij het genereren van dienstregelingen uit een dienstregelingstructuur indien die dienstregelingen gebruikt worden voor het specificeren van infrastructuur. In dit proefschrift behandelen we dan ook het gerandomiseerd genereren van dienstregelingen.

Het gerandomiseerd genereren van dienstregelingen gebeurt in twee stappen. Ten eerste worden enkele klassen van dienstregelingen geloot uit een gegeven dienstregelingstructuur volgens een stochastische verdeling die klassen die robuuste dienstregelingen bevatten bevoordeelt. Daarna worden uit elke klasse een aantal dienstregelingen geloot volgens de uniforme verdeling. Deze worden vervolgens gebruikt om infrastructuur te specificeren en de robuustheid van de specificatie te bepalen. Hoe deze specificatie en gevoeligheidsanalyse gedaan moeten worden is niet aan de orde in dit proefschrift. Het gerandomiseerd genereren van dienstregelingen is een niet-triviaal probleem omdat enkel het bepalen van een dienstregeling al een NP-volledig probleem is.

Afgezien van het bovenstaande behandelen we ook het Steiner Triple Set Covering Problem. Dit is het probleem om gegeven een 0-1 matrix zijn 1-width te bepalen. In de literatuur
zijn specifieke probleeminstanties voorgesteld als testproblemen voor oplosmethoden voor 0-1 programmeringsproblemen. Voor de twee grootste van deze problemen presenteren we verbeterde covers. Om deze te vinden werd een eenvoudig algoritm gebruikt dat ook wordt beschreven.
Curriculum Vitae