A stylized graphic of a road intersection on a dark purple background. White lines represent the road lanes. Several cars are depicted: a yellow car on the left vertical road, a green car on the right vertical road, and a red car on the horizontal road. Some cars are enclosed in red or green rectangular boxes, possibly indicating specific traffic states or queues. Arrows indicate the direction of traffic flow.

Queueing models for urban traffic networks

Anna Oblakova

QUEUEING MODELS FOR URBAN TRAFFIC NETWORKS

Anna Oblakova

QUEUEING MODELS FOR URBAN TRAFFIC NETWORKS

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
Prof. dr. T.T.M. Palstra,
on account of the decision of the graduation committee,
to be publicly defended
on the 27th of September 2019 at 16:45 hours

by

Anna Igorevna Oblakova

born on the 21st of February 1992
in Moscow, Russia

This dissertation has been approved by:

Prof. dr. R.J. Boucherie
Prof. dr. W.H.M. Zijm
Dr. J.C.W. van Ommeren
Dr. ir. A. Al Hanbali

ISBN: 978-90-365-4847-2

DOI: 10.3990/1.9789036548472

Typeset in L^AT_EX. Printed by Gildeprint, Enschede, the Netherlands

Copyright © 2019, Anna Oblakova, Enschede, the Netherlands

All rights reserved. No parts of this thesis may be reproduced, stored in a retrieval system or transmitted in any form or by any means without permission of the author.

This thesis is part of the Dynafloat project, which was supported by Topconsortia voor Kennis en Innovatie Logistiek and Nederlandse Organisatie voor Wetenschappelijk Onderzoek, grant number 438-13-206.

This thesis is part of the PhD thesis series of the Beta Research School for Operations Management and Logistics (onderzoeksschool-beta.nl) in which the following universities cooperate: Eindhoven University of Technology, Maastricht University, University of Twente, VU Amsterdam, Wageningen University and Research, and KU Leuven.

Dissertation committee

Chairman & secretary:	Prof. dr. J.N. Kok <i>University of Twente</i>
Promotors:	Prof. dr. R.J. Boucherie <i>University of Twente</i> Prof. dr. W.H.M. Zijm <i>University of Twente</i>
Co-promotor:	Dr. J.C.W. van Ommeren <i>University of Twente</i>
Members:	Dr. ir. A. Al Hanbali <i>King Fahd University of Petroleum and Minerals</i> Prof. dr. O.J. Boxma <i>Eindhoven University of Technology</i> Prof. dr. R. Núñez Queija <i>University of Amsterdam</i> Prof. dr. ir. B. van Arem <i>Delft University of Technology</i> Prof. dr. ir. E.C. van Berkum <i>University of Twente</i> Prof. dr. S.A. van Gils <i>University of Twente</i>

*‘Still round the corner there may wait
A new road or a secret gate.’*

John Tolkien

Acknowledgements

Five years ago, I could not have imagined that I would be now finishing my PhD in the Netherlands. Moving from Russia and switching from pure to applied mathematics were huge and challenging steps in my life. Now, I would like to express my gratitude to those who contributed to the completion of my PhD and made me enjoy the time spent here.

First, I would like to thank my supervisors, Richard Boucherie, Henk Zijm, Jan-Kees van Ommeren and Ahmad Al Hanbali, for their guidance and support. Richard and Henk, thank you for keeping me on track, when I had too many ideas what to do next, and for all your useful remarks on research presentation, which helped me to improve my scientific writing. Jan-Kees and Ahmad, thank you for your patience when reading my too short and sometimes unintelligible first drafts, for putting all ‘the’s where they belong and for our weekly meetings, which often included interesting discussions on all kinds of topics.

I am grateful to my graduation committee, Onno Boxma, Sindo Núñez Queija, Bart van Arem, Eric van Berkum and Stephan van Gils, for the time spent on my thesis and defence.

Next, I would like to thank all members of the Dynafloat project, Richard, Jan-Kees, Ahmad, Onno, Ivo, Marko, Johan, Rick, Rik, Rob, Sindo, Elenna, Sara, Wim and Gerard, for our inspiring meetings. My special thanks are to my paronyms, Sara and Rik. Rik, thank you for your interest in my research and your encouragements that made me revisit old results and led to [Chapter 3](#) of this thesis. Sara, thank you for your enthusiasm and our nice talks.

I also wish to thank all my colleagues from MOR and IEBIS groups. Werner, Judith, Nelly, Jasper and Maarten, it was my pleasure to teach Probability theory and Stochastic processes with you. Nelly, thank you for the opportunity to talk in Russian from time to time and for your suggestion to participate in the Anna-Karenina readings, which was an unusual but exciting experience for me. Aleida, Mihaela, Kamiel, Shokoufeh, Jasper de Jong, Maartje, Michael, Pim, Tom, Nardo, Sem, Gréanne, Berksan, Xinwei, Ingeborg, Corine, Anne, Joost, Thomas, Stefan, Victor, Maarten, Jasper Bos, Eline, Marelise, Jasmijn, Nicky, Bodo, Marc, Johann, Clara, Lerna and those whom I forgot to mention, thank you for the enjoyable

time during coffee breaks, lunches and Lunteren conferences. Abhishta, Andrej, Wouter, Nils, Arturo, I wish I would have spent more time at IEBIS. Mihaela, Pim, Shokoufeh, Tom, Thomas, Maarten and Jasper, it was great to be officemates with you. Maarten and Jasper, thank you for tolerating my Dutch and teaching me some new words.

During most of my PhD, I was living in Groningen, where I got to know several PhD students from the University of Groningen. I would especially like to thank Anneroo, Bolor, Mónica, Matthijs, Réka, Yongjiao and, most of all, Nikolay for pleasant outings, game nights and films watched together.

I would like to thank all my relatives who supported and helped me through this time.

Nikolay, thank you for your support and patience especially during the last year of my PhD and for the wonderful time we spend together.

Anna Oblakova
Enschede,
August 2019

Contents

I	Introduction	1
1.	Motivation and literature review	3
1.1.	Randomness of urban traffic	4
1.2.	Congestion countermeasures	4
1.3.	Queueing approach	5
1.4.	General traffic models	6
1.5.	Queueing models in traffic	7
1.6.	Numerical approaches in queueing	14
1.7.	Contribution of this thesis	15
II	Queueing	23
2.	An integral approach for the fixed-cycle traffic-light model	25
2.1.	Our method	26
2.2.	Method comparison for the bulk-service queue	33
2.3.	FCTL-type queues	41
2.4.	Conclusions	58
2.A.	Proof of Lemma 2.15	58
2.B.	Computational remarks	60
3.	Roots, symmetry and contour integrals in queueing systems	63
3.1.	Preliminaries	65
3.2.	Pgf as a symmetric function of roots	72
3.3.	Factorisation of the pgf	74
3.4.	Conclusions	81
3.A.	Proofs of the auxiliary results	81

III Traffic networks	91
4. A tandem of intersections under semi-actuated and fixed control	93
4.1. Model	95
4.2. Numerical results	113
4.3. Conclusions	131
5. Green-wave efficiency for a tandem of traffic-light intersections	133
5.1. Green-wave efficiency and average delay per vehicle	134
5.2. Design of green waves	139
5.3. Optimal settings for an arterial road	144
5.4. Conclusions	148
5.A. Arrival process	149
6. An approximation model for large urban traffic networks	159
6.1. Model	161
6.2. Numerical results	174
6.3. Conclusions	179
7. Conclusions	181
Glossary	185
Bibliography	193
Summary	205
Samenvatting	207
About the author	209

Part I

Introduction

Chapter 1

Motivation and literature review

‘The ability of traffic engineers to build traffic signal regulators with a wide variety of controls far exceeds their ability to understand the effect of these controls upon the traffic.’

Darroch et al. (1964)

Current daily life is unimaginable without the ability to travel. Daily trips include, for example, commuting, shopping, and visiting friends and relatives. In the Netherlands, a person travels on average 32 kilometres a day which may take some 63 minutes, see [Berends-Ballast et al. \(2016\)](#). Besides personal trips, there exists a large goods transportation sector. Such an extensive use of limited road infrastructure often leads to *traffic jams*, which form a major problem for cities all over the world. For example, in 2018, drivers in Moscow, London and Bogota lost on average at least 210 hours in congestion, see [Reed and Kidd \(2019\)](#). Such excessive delays negatively affect not only individual travellers but also the economy and the environment. In particular, workers and goods may arrive late at their destinations, which decreases productivity of the companies. Long travel times increase fuel consumption and, therefore, air pollution. In 2011, congestion in urban areas of the United States caused 25 million tons of CO_2 emissions and cost 121 billion dollars, see [Schrank et al. \(2012\)](#). Additionally, traffic jams put public health and well-being at risk. During rush hours, the drivers may get stressed and show a more aggressive behaviour, see [Shinar and Compton \(2004\)](#), which may lead to accidents and further disruption of the traffic flow. Moreover, congestion may reduce the speed of emergency vehicles and, thus, prevent them from providing the required service.

1.1. Randomness of urban traffic

Traffic networks in urban areas are complex stochastic systems. The network behaviour depends on the decisions of thousands of people on aspects such as destination, timing and travel route. Moreover, even when these decisions are made, the actual trip may deviate from the planned one. For example, a person may leave earlier or later or will change the intended route due to an accident, maintenance works or a suggestion by his/her navigating system. Also, the travel time is considerably influenced by the reaction time of the drivers.

The random nature of traffic has a great impact on the travel times, see [Calvert et al. \(2012\)](#). For example, consider a road with a [traffic-light intersection](#). Due to randomness, there may be more arrivals at the road during a *cycle*, i.e., red time and consecutive green time, than can depart during one green time. The vehicles that could not depart in this green time join a queue, wait till the next green time and, hence, suffer from possibly long delays. When there are several cycles with excessive arrivals, the delayed vehicles may need to wait several cycles before leaving the queue. The effect is especially strong when the *demand*, i.e., average number of arrivals per cycle, is close to or even exceeds the *capacity*, i.e., maximum number of departures per cycle. The demand is usually time-dependent with its peak during morning or evening rush hours. Though, in general, the dynamics of demand during a day is known, the actual realisation of the arrivals over time may lead to ‘good’ or ‘bad’ days resulting in huge variation in travel times over days, see [Langer \(2005\)](#). The randomness of departure times also influences the delays. Even a few drivers that are departing slower from a [lane](#) than others decrease the capacity and, thus, affect the performance of the system. The departure times depend not only on the reaction time of the drivers but also on the vehicles’ characteristics and types. For example, a truck or a bus accelerates slower than a private car. Such diversity of vehicle types also affects the travel times between intersections leading to possible extra delays.

1.2. Congestion countermeasures

During the past decades, many approaches have been proposed for reducing traffic jams. Since the delays are longer for higher *load*, i.e., demand to capacity ratio, the congestion countermeasures usually focus on increasing the capacity or reducing the demand especially during rush hours. One way to increase the capacity is to expand the infrastructure. This method is very expensive and almost impossible in densely-populated metropolises. Moreover, provision of new roads may lead to *induced demand*, see [Goodwin \(1996\)](#), or to longer delays due to re-routing. The latter situation is known as Braess’s paradox, see [Braess \(1968\)](#). The use of *reversible lanes*, i.e., lanes that can be travelled in one or another direction depending on the conditions, provides a way to increase the capacity for the existing roads, see [Lu et al. \(2018\)](#).

Many approaches focus on reduction of demand, see [Gärling and Schuitema \(2007\)](#). This reduction may be achieved by promoting other modes of transportation such as public transport or cycling. For example, it was recently announced that Luxembourg will become the first country with free public transport, see [Boffey \(2018\)](#). In the Netherlands, cycling, though always being popular, was actively promoted starting from 1970s, [Stoffers \(2012\)](#), and is now so common that some cities encounter bicycle traffic jams, see [Wagenbuur \(2014\)](#). However, such solutions are not applicable to every city. For example, in Moscow, hills, city size and winter temperatures make daily cycling almost impossible. Another way of demand reduction is to impose direct or indirect traffic restrictions, which naturally contributes to the promotion of public transport or, e.g., carpooling. An example of an indirect restriction is limiting the parking space, see [Knoflachner \(2006\)](#). Direct restrictions are often applied to some groups of vehicles during certain time periods, such as delivery trucks during rush hours, see [Yannis et al. \(2006\)](#), and licence-plate rationing, e.g., only vehicles with even, resp. odd, licence-plate numbers are allowed in the city on alternate days, see [Rao et al. \(2017\)](#). Similarly, part of the infrastructure can be reserved for certain vehicle types. For example, in the United States, some lanes can only be used by high-occupancy vehicles, i.e., vehicles with two or more passengers, see [Javid et al. \(2017\)](#). Instead of full restrictions, it is possible to apply pricing schemes such as toll roads and properly-chosen parking costs, see [Nourinejad and Roorda \(2017\)](#), or, alternatively, rewarding schemes for commuters outside rush hours, see [Ben-Elia and Ettema \(2011\)](#).

To decrease congestion, it is also important to utilise the existing infrastructure more efficiently, for example, by choosing traffic-light control settings that minimise delays. This includes coordination of neighbouring intersections and better utilisation of the cycle time. To do so, one needs to understand how traffic-light settings influence the performance of the network. In this thesis, we analyse this influence using queueing theory.

1.3. Queueing approach

An urban traffic network can be naturally represented as a queueing network. The queues appear at intersections, which are the bottlenecks in the system. Vehicles travel between the queues along certain routes and leave the network when they reach their destination. At traffic-light intersections, the traffic-light control influences the moments when each vehicle is released from its current queue and can proceed to the next queue.

As will be demonstrated in this thesis, the queueing approach allows one to model traffic networks in an effective way. With a small number of parameters, an analytic queueing model can be as accurate as detailed traffic microsimulations that focus on the behaviour of individual vehicles, while having higher computation speed. Thus, this approach is very well suited for studying the advantages and

disadvantages of possible traffic policies or for the optimisation of the traffic-light settings.

In this thesis, we focus on modelling urban traffic networks using queueing theory and on the theoretical aspects concerning the computation of the queue length in such models. In the remainder of this chapter, we review general and queueing traffic models, see [Sections 1.4 and 1.5](#), and common techniques to obtain queue-length distributions, see [Section 1.6](#). We show that certain realistic elements of traffic behaviour are often overlooked in analytic traffic models and that classical queueing techniques for traffic models require some implicitly-defined variables. In [Section 1.7](#), we conclude the chapter with a summary of the contributions of this thesis.

1.4. General traffic models

The problem of traffic control exists for more than a century. Even before the invention of the car, traffic in big cities was a source of accidents. The first (gas-lit) traffic lights were installed in London in 1868 to protect the pedestrians, see [Mueller \(1970\)](#). Though the experiment resulted in a gas explosion, the traffic lights were considered successful in controlling the traffic and soon spread to other cities and countries.

At first, the traffic lights were manually operated and controlled. A police officer would sit in a special tower at an intersection and decide which direction should be served. This changed in the beginning of the 20th century when electric traffic lights were introduced. They were equipped with timers and operated under *fixed-cycle control* (or just *fixed control*), giving each direction a fixed length of green time, see [McShane \(1999\)](#). The cycle length and green times were based on engineering surveys, and sometimes the resulting cycles were adjusted using a trial-and-error process.

In the 1950s, the traffic control problem drew the attention of the scientific community, see [Gazis \(2002\)](#). Works of [Reuschel \(1950\)](#), [Pipes \(1953\)](#), [Lighthill and Whitham \(1955\)](#), [Richards \(1956\)](#) and [Webster \(1958\)](#) laid foundations for three types of traffic models, which are characterised by their level of detail: microscopic, macroscopic and mesoscopic models.

The **microscopic** models focus on individual vehicles. The traffic behaviour is described by the relation of positions, speeds and accelerations of the vehicles on the road. In particular, in *cellular automata* models, both time and space are discretised and the vehicles are ‘hopping’ between the ‘cells’ according to their speeds and positions of the other vehicles, see, e.g., [Nagel \(1996\)](#), [Maerivoet and De Moor \(2005\)](#). The *car-following* models characterise the movement of a vehicle in a (continuous) lane depending on the leading vehicle, see, e.g., [Reuschel \(1950\)](#), [Pipes \(1953\)](#), [Krauß \(1998\)](#), [Brackstone and McDonald \(1999\)](#). The *lane-changing* models complement the car-following models by providing decision-making and lane-changing algorithms, see, e.g., [Toledo et al. \(2005\)](#), [Kesting et al. \(2007\)](#). The

microscopic models often include randomness and are used in traffic microsimulations such as SUMO (Simulation of Urban Mobility, see Lopez et al. (2018)) and VISSIM (Verkehr In Städten - SIMulationsmodell, see Fellendorf and Vortisch (2010)). In such microsimulations, the behaviour of each vehicle is simulated according to the corresponding car-following and lane-changing models. This often results in realistic behaviour but also long computation times.

In the **macroscopic** models, the ideas of hydrodynamics are used to explain the traffic behaviour in terms of speed v , density k , and flow $q = kv$ on the road. Speed and density are considered to be functions of time and position at the road, and the main challenge is to find an equation that describes the evolution of the system. In *Lighthill-Whitham-Richards* models, see Lighthill and Whitham (1955) and Richards (1956), it is suggested that speed is a function of density. This assumption leads to non-continuous solutions with *shock waves*, see Newell (1993a), and poses a challenge of describing the speed-density relation, see, e.g., Greenshields et al. (1935), Daganzo (1994), Newell (1993b). In *Payne* models, see Payne (1971), a differential equation on speed derived from a car-following model is suggested. The *Helbing* models extend Payne models by including the variation of the speed, see Helbing (1996). Though originally the macroscopic models were applied only to highway flows, it is also possible to use them to predict blocking in traffic networks, see, e.g., Knoop et al. (2008).

The **mesoscopic** models provide more details than the macroscopic models but do not describe the behaviour of individual vehicles as in microscopic models. The quantities are stochastic and the traffic behaviour is described in terms of distributions. The *gas-kinetic* models, see Prigogine and Herman (1971), consider the speed distribution. The extensions include the desired speed of vehicles, see Paveri-Fontana (1975), multiple classes of vehicles, see Hoogendoorn and Bovy (1998), and multiple lanes, see Helbing (1997). Other mesoscopic models consider the distribution of the *time-headway*, i.e., distance in time, between vehicles (Zhang et al. (2007)) or of the size of a vehicle cluster (*platoon*) (Mahnke and Kühne (2007)). The traffic queueing models, see the following Section 1.5, belong to the class of mesoscopic models but sometimes are described as microscopic models, see Abhishek (2019), and are often overlooked in the model classifications. For a more substantial review on non-queueing traffic models, we refer to Hoogendoorn and Bovy (2001) and van Wageningen-Kessels et al. (2015).

1.5. Queueing models in traffic

The queues in traffic networks naturally appear at intersections, and the random nature of traffic behaviour plays an important role in this queueing process. Even a small fluctuation in the arrival rates may influence the performance of the traffic system especially for a high traffic load, see Calvert et al. (2012). Therefore, it is essential to take the randomness of the arrival process into account, which is within the realm of queueing theory. For the analysis of traffic networks, it is important

to understand what happens at one intersection, see [Subsection 1.5.1](#), and how the queues at closely located intersections influence each other, see [Subsection 1.5.2](#). We note that application of queueing theory to traffic is not limited to networks of intersections. It is interesting to see that it was used to explain the *fundamental diagram*, i.e., the relation between density k and flow q , see [Baer et al. \(2019\)](#) and references therein.

1.5.1. An isolated intersection

The basic traffic model for an isolated intersection is the *fixed-cycle traffic-light (FCTL)* model, which we describe in detail to show the usual assumptions of the queueing traffic models, see [Subsection 1.5.1.1](#). In [Subsection 1.5.1.2](#), we present an overview of models for various control policies. Then, we discuss the results on transient behaviour in [Subsection 1.5.1.3](#). Finally, we relate the considered models to polling systems in [Subsection 1.5.1.4](#).

1.5.1.1. FCTL model

The FCTL model is a discrete-time model that describes the evolution of the queue length at a lane operated under fixed control. The cycle is split in green and red time periods, where the yellow time is disregarded for the following reason. Many traffic models instead of the real green time consider the *effective green time*, during which the queued vehicles depart at equal time intervals of τ seconds, see [Branston and van Zuylen \(1978\)](#). The effective green time of a lane is given by the following formula

$$g_{\text{eff}} = g_{\text{real}} - s + e,$$

where g_{eff} (resp., g_{real}) is the effective (resp., real) green time, s is the *start-up lag*, i.e., the time lost due to acceleration of the queued vehicles, and e is the *green-end lag*, i.e., the part of the yellow (amber) time during which the vehicles still depart, see [Figure 1.1](#). The rest of the cycle is the *effective red time*, during which there are no departures from the lane.

In the FCTL model, a time interval is equal to τ seconds. A green period consists of g time intervals and a red period of r time intervals giving in total $c = g + r$ time intervals. The time intervals during a green (resp., red) period are called green (resp., red) time intervals. During each time interval, Y vehicles arrive at the lane independently of the previous arrivals.

The queue at the stop line is assumed to be *vertical*, which means that the arriving vehicles move at the *free-flow speed* till the stop line, where they stop instantaneously and wait their turn ‘on top of each other’, see [Motie and Savla \(2015\)](#). At the moment of departure the vehicles immediately accelerate to the free-flow speed and leave the system with one departure per time interval, see [Figure 1.1](#). In vertical-queue models, the real positions of the vehicles in the lane are not important. In contrast, *horizontal-queue* models focus on the actual position of the last vehicle in the queue, see [Motie and Savla \(2015\)](#). The relation between

Figure 1.1: The real and effective green times. The real and idealised trajectories of the vehicles are given by the solid black lines and dashed grey lines, respectively, and numbered for referencing in the text. In a vertical-queue model with effective green time, the vehicles are assumed to depart with the free-flow speed at equal departure intervals during the effective green time, see dashed lines. The red, green, yellow and black bold horizontal lines represent real red, green, yellow times and effective red times, respectively. The vertical axis shows the distance between the vehicles and the intersection.

the vertical and horizontal queues is not straightforward and depends on the green time and wave propagation in the queue, see [Quinn \(1992\)](#). However, the *(total) delay* (or *lost time*) of a vehicle, i.e., the difference between actual travel time and travel time under the free-flow speed, is the same in both vertical- and horizontal-queue models as long as the departure moments from the system (several meters after the intersection) are the same for these models, see [Figure 1.1](#). Thus, the use of complicated horizontal-queue models is mostly relevant for estimation of the spill-back and the blocking probabilities in traffic networks, see [Subsection 1.5.2](#).

The departures in the FCTL model are only possible during the green time intervals. In particular, as long as the queue is not empty, one vehicle departs each time interval. However, if the queue is empty, all arriving vehicles proceed without delay, and the queue remains empty till the next red time.

In the last 60 years, the FCTL model was extensively studied. In [Beckmann et al. \(1956\)](#), a relation between the *overflow queue length*, i.e., queue length at the beginning of the red time, and the average delay per vehicle was obtained for

the case of Bernoulli arrivals, i.e., when Y is a Bernoulli random variable. Webster (1958) used a simulation to find a semi-empirical formula for the average delay in the case of Poisson arrivals. This formula has three terms, where the first term is the deterministic delay, i.e., the delay for the case of deterministic inter-arrival times, the second term is the delay at a ‘bottleneck’ with a deterministic service time, i.e., $M/D/1$ queue, and the third term is an empirical compensation term. Newell (1960) noted that for the case of Bernoulli arrivals the overflow queue behaves as the bulk-service queue, see Bailey (1954):

$$X_{n+1} = [X_n + A_n - g]^+,$$

where X_n is the overflow queue length in cycle n , A_n is the number of arrivals during cycle n , and $x^+ = \max\{x, 0\}$. This remark allowed Newell to represent the probability-generating function (pgf) $X(z)$ of the overflow queue length in terms of the roots of the characteristic equation **outside** the unit disk, similar to the later result of Chaudhry and Kim (2003) for the bulk-service queue. Later, Darroch (1964) proved that $X(z)$ for the case of general arrivals has the following form:

$$X(z) = \frac{\sum_{k=0}^{g-1} q_k z^k (Y(z))^{g-k-1}}{z^g - (Y(z))^c} (z - Y(z)), \quad (1.1)$$

where $Y(z)$ is the pgf corresponding to Y , and q_k is the probability that the queue is empty after k green time intervals. Darroch (1964), similar to Bailey (1954), suggested to determine the unknown probabilities using the roots of the equation

$$z^g = (Y(z))^c \quad (1.2)$$

inside the closed unit disk. Due to the analyticity of the pgf $X(z)$ in the unit disk, these roots should be zeroes of the numerator, which gives a system of linear equations on the unknown probabilities, see for more details Section 1.6 below. This solution is numerically challenging due to the problem of finding the roots. Therefore, many authors suggested approximations or bounds for the average overflow queue length, see, e.g., McNeil (1968), Newell (1965), van den Broek et al. (2006), or algorithms and formulas to find the roots of equation (1.2), see, e.g., Janssen and van Leeuwen (2008). Using the roots, it is possible to find the pgf of the delay, see van Leeuwen (2006).

1.5.1.2. Beyond fixed control

The analysis of an isolated intersection is not only limited to fixed control. Authors also consider different types of vehicle-actuated control and study intersections without traffic lights.

When traffic detectors are available for all lanes, one may use *fully-actuated control*. The detectors measure the presence of a queue and/or the *time-headway* between arriving vehicles. Depending on this data a decision is made either to

continue serving the corresponding direction or to switch to the following direction. In the literature, several variants of fully-actuated control are analysed. They differ in the number of (simultaneously) served lanes, the decision policy and assumptions on the arrival and departure processes. For example, [Darroch et al. \(1964\)](#), [Lehoczký \(1972\)](#) and [Litvak and Fedotkin \(2000\)](#) consider an intersection of two one-way roads. In [Darroch et al. \(1964\)](#), each road is served until there are no vehicles in the queue and the time-headway is at least $\beta \geq 0$. This model is analysed for Poisson arrivals and random departure and switch times. In [Lehoczký \(1972\)](#), each road is served until the queue empties (independently of the time-headway). The arrival process is given by a discrete-time Markov chain, and the departure times are deterministic as in the FCTL model. The analysis in these models is relatively simple since the overflow queues are empty. In [Litvak and Fedotkin \(2000\)](#), after a fixed start-up time, the green time for the road is extended by a given value if the queue has at least K vehicles, or there was an arrival. If the number of extensions exceeds a certain value n , the green time ends independently of the queue length. The values of K and n may depend on the road. Even for Poisson arrivals and deterministic service times, this model is quite complicated, and the analytical results are limited to the necessary and sufficient conditions for system stability. In reality, there are usually more than two lanes at an intersection. Moreover, several lanes have green time simultaneously, in which case they are said to form a *phase*, which further complicates the analysis, see [Boon et al. \(2012\)](#). In the latter paper, the inter-arrival and inter-departure times are assumed to be general, and a phase is served as long as there is a queue on at least one lane. Similarly to the FCTL model, the vehicles that arrive during the green time and find no queue proceed without any delay. For this model, the authors obtain asymptotic results for heavy and light traffic, which they use to determine an approximation for all loads.

Sometimes the detectors are only installed on part of the lanes, which can be significantly cheaper than installing detectors on all lanes. For example, detectors can be placed on minor roads that intersect a major multi-lane road, see [Tarnoff and Parsonson \(1981\)](#). Such an intersection can operate under *semi-actuated control*. The green time for the main direction can be fixed, while the green time of a minor road depends on the queue. In [Macedo et al. \(2017\)](#), the queue length at a minor lane is modelled as an $M/D/1$ queue with deterministic vacations, during which the major road is served. The green time of the minor lane has a fixed minimum and maximum length. If the cycle length is fixed, this type of control can be used for a tandem of intersections to provide coordination between intersections, see [Lee \(2016\)](#).

Intersections without traffic lights can be divided in three types: priority intersections, uncontrolled intersections and roundabouts. At *priority intersections*, the vehicles at the low-priority road need to yield to the vehicles on the high-priority road. A typical example would be a highway crossing. The queue at the low-priority road is analysed using gap-acceptance models, see [Heidemann and Wegmann \(1997\)](#), [Weiss and Maradudin \(1962\)](#). For a substantial review and the

most recent results, we refer to [Abhishek \(2019\)](#). *Uncontrolled intersections* are intersections for which there are no priority signs and the vehicles yield to the vehicles on their right (or on their left for left-hand traffic as in the United Kingdom or Japan). The traffic rules do not specify what should be done if there is traffic from all directions. This type of intersections is used for rural and residential areas, where the traffic densities are very low, and, to the best of our knowledge, was not analysed using queueing theory. A *roundabout* is a circular intersection, which is used as an alternative for uncontrolled intersections or *traffic-light intersections* when two or more roads (with similar traffic densities) intersect, see [Pochowski et al. \(2016\)](#). According to [Storm et al. \(2018\)](#), the analysis of a roundabout is difficult due to correlation between the queues, and only partial results can be obtained.

1.5.1.3. Transient analysis

Queueing systems are often analysed in steady state. A necessary condition to do so is the stability of the system. For the case of the FCTL queue, it means that the expected number of arrivals per cycle is less than the green time, i.e., $cY'(1) < g$, see the notation in [Subsection 1.5.1.1](#). However, during rush hours, the stability condition is not guaranteed. Moreover, the arrival rates are time-dependent, and even when they change slowly, the system does not necessarily reach the steady state, see [Son et al. \(1995\)](#).

For these reasons, some authors study the behaviour of the (average) delay depending on the time of observation. Based on the approximation results for the FCTL queue, [Akçelik \(1980\)](#) proposes an approximation of the average delay of the vehicles during a time period T . Similar to [Webster \(1958\)](#), the delay is composed of several quantities, which correspond to deterministic and random sources of delay. In [Fu and Hellenga \(2000\)](#), an approximation of the delay variance is suggested. These types of results are given in the Highway capacity manual, see [Transportation Research Board \(2000\)](#).

However, it is also possible to obtain analytical results using queueing models. In [Olszewski \(1994\)](#), the average delay of the vehicles arriving at one cycle is derived in terms of the overflow queue length, the distribution of which is calculated from the previous overflow queue length. A similar approach is used in [van Zuylen and Viti \(2007\)](#) and [Viti et al. \(2007\)](#) to find the delay distribution for the case of fixed and fully-actuated control, respectively.

1.5.1.4. Comparison to polling systems

A traffic intersection closely resembles a *polling system*, where a number of queues are served by a single server according to a certain service policy, see [Takagi \(2000\)](#). In the case of *exhaustive* service, a queue is served until it empties, while for the *gated* service, the server serves all the customers that were present in the queue upon arrival of the server, see [Boxma and Groenendijk \(1987\)](#). For

the (*exhaustive*) *time-limited* service, the server visits the queue for a given time period (or until the queue empties), see [Al Hanbali et al. \(2012\)](#), [Frigui and Alfa \(1998\)](#). The service times of customers in a queue are assumed to be independent and identically distributed (i.i.d.) random variables.

Fixed and actuated controls can be modelled as non-exhaustive time-limited and exhaustive services, respectively. However, there are some differences between traffic intersections and the classical polling systems. For example, there are usually multiple lanes that have green time simultaneously, which means that the server visits several queues at the same time. Moreover, the assumption on the i.i.d. service times may be debatable since vehicles that arrive during green time and find no queue proceed faster than the delayed vehicles, see also [Section 1.7](#) where we discuss problems concerning the usage of effective green time. Despite these differences, the methods for analysing polling systems are widely used for traffic queueing models. For example, authors often relate the pgfs of the queue length in two consecutive cycles to find the steady-state distribution, see also [Boon \(2011\)](#).

1.5.2. Network models

Analysis of traffic networks poses several challenges. First, the arrivals from an upstream intersection form *platoons*, which means that the arrivals are correlated and time dependent. Second, the queues at different intersections depend on each other, which makes the system multi-dimensional. Third, due to the finite length of the roads between the intersections, blocking may occur, which affects the traffic flow through the system.

The literature on queueing network models is limited. Some authors apply results for an isolated intersection to analyse the behaviour of the network. In [Newell \(1989\)](#), an arterial road that operates under a common cycle length is described in terms of the busiest intersection. The delay is split in two parts: the deterministic part, which is due to *offsets* between intersections, and the stochastic part due to queueing. An offset is defined as the time between the beginning of a cycle at one intersection and the beginning of a cycle at the other intersection. The deterministic and stochastic parts of delay are affected by the choices of offsets and green splits, respectively. The author provides lower and upper bounds on the stochastic delay and gives some qualitative remarks concerning control parameters. A similar approach of considering stochastic queueing at one intersection and deterministic flow between intersections is proposed in [Zheng and van Zuylen \(2014\)](#). In this paper, two intersections are considered, and the authors derive the *waiting time* in terms of an overflow queue length at the first intersection, the distribution of which is suggested to be estimated using empirical data.

In [Tarko \(2000\)](#), the results of [Miller \(1963\)](#) are used to approximate the overflow queue lengths at the intersections. The main idea is to analyse the variance of the number of arrivals by considering filtering (due to traffic lights), merging and splitting of the streams. The effect of the offsets is ignored. In [Akçelik and](#)

Rouphail (1994), the approximation formula for the transient delay at an isolated intersection is generalised to the case of platooned arrivals.

In Boon and van Leeuwen (2018), the FCTL model is generalised to a tandem of intersections. The authors decompose the network into individual lanes and include the dependency between lanes in the arrival process. For the sake of tractability, the authors consider the joint distribution of the arrivals at a lane in one cycle and assume independence of the arrivals between cycles.

Another way of modelling traffic networks is to represent each lane as an $M/M/1/l$ queue, see, e.g., Osorio and Chong (2015), Osorio and Wang (2017), Osorio and Yamani (2017). The service rate of the lane depends on the green time of the lane and on the eventual blocking downstream. In such an approach, the effect of offsets is not included, and a detailed microsimulation can be used to complement the model, see Osorio and Chong (2015). In these models, the blocking occurs due to finite capacity of the vertical queues. In Lu and Osorio (2018), a link model is proposed that incorporates the shock-wave theory results. The state of the queue at the upstream end of the lane, i.e., of the horizontal queue, is described in terms of the downstream (vertical) queue and the lagged inflow and outflow queues. The horizontal queues are also considered in Viti et al. (2009), where two intersections in tandem are analysed. In these models, when the downstream lane is full, the possibility of entering this lane is blocked, and the vehicles wait at the upstream queues. In this case, the queue at the downstream lane spills back but does not block the vehicles at the crossing lanes. The analysis of the case when blocking of crossing lanes occurs is particularly difficult since the vehicles that violate traffic regulations and stop at the intersection may lead to a gridlock, when no vehicles can move to the desired lane as their way is blocked.

Finally, we mention simulations which are based on queueing models. In Gawron (1999) and Grether et al. (2012), the vehicles travel between vertical queues in the traffic networks without and with traffic lights, respectively, see also www.matsim.org for MATSim (Multi-agent Transport Simulation). In Marinica and Boel (2012), the simulation is based on the behaviour of the platoons.

1.6. Numerical approaches in queueing

Systems like the FCTL model or the bulk-service queue are often analysed in terms of roots of the characteristic equation, see Darroch (1964) and Bailey (1954). Here, we briefly explain the idea for the FCTL model, see Subsection 1.5.1.1. Note that the pgf of the overflow queue length $X(z)$, see (1.1), should be continuous in the closed unit disk $\bar{D}_1 = \{z: |z| \leq 1\}$. Thus, any zero of the denominator in $\bar{D}_1$ should be also a zero of the numerator. In the case of a stable system, there are exactly g such zeroes, which are denoted by $z_0 = 1, z_1, \dots, z_{g-1}$, see Adan et al. (2006). Substituting $z = z_k$, $k = 1, \dots, g - 1$ in the numerator of $X(z)$ gives 0, which yields $g - 1$ linear equations on the unknown probabilities q_k , $k = 0, \dots, g - 1$. An additional equation is given by $X(1) = 1$. If the zeroes are distinct, there is

a unique solution of the obtained system of linear equations. In this way, it is possible to find the unknowns using the roots of equation (1.2).

This *root-finding approach* is used for many discrete- and continuous-time queueing systems. Examples are a discrete-time $G_D/G_D/1$ queue with exceptional first service, see [van Ommeren \(1991\)](#), a $G/D/1$ queue with occasional extra server, see [Bruneel and Wittevrongel \(2017\)](#), an $M/G_r^Y/1$ queue with bulk-size-dependent service, see [Pradhan et al. \(2016\)](#), a continuous-time queue with either Markov arrival process or Markov service process, see [Chaudhry et al. \(2013\)](#) and [Chaudhry et al. \(2012\)](#).

The main drawback of this method is finding the roots since there is no explicit formula in general. Moreover, the derived solution can be very sensitive to the precision of the roots, which, in turn, can be poor even due to a small mistake in the coefficients, see, e.g., the study of the so-called Wilkinson's polynomial [Wilkinson \(1984\)](#).

Another approach is to represent the queue as an $M/G/1$ or $G/M/1$ -type queue, which means that the transition matrix is given by a block matrix

$$P_{M/G/1} = \begin{pmatrix} B_0 & B_1 & B_2 & \dots \\ C_0 & A_1 & A_2 & \dots \\ 0 & A_0 & A_1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad \text{or} \quad P_{G/M/1} = \begin{pmatrix} B_0 & A_0 & 0 & \dots \\ B_1 & A_1 & A_0 & \dots \\ B_2 & A_2 & A_1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Then, it is possible to apply the *matrix-analytic approach*, see [Neuts \(1984\)](#). The distribution is computed using the minimal non-negative solution of a certain matrix equation and Ramaswami's formula, see [He \(2014\)](#). To find the moments of distribution it may be better to use the aggregated distribution, see [Riska and Smirni \(2002\)](#). Both the FCTL queue, see [Pacheco et al. \(2017\)](#), and the bulk-service queue can be represented as $M/G/1$ -type queues. Other examples of applications are the $G/PH/1$ queue, see [Neuts \(1981\)](#), and a queue with working vacations, see [Tian et al. \(2009\)](#).

The speed of the matrix-analytic method depends on the size of a block in the transition matrix. As with the root-finding method, there are no general formulas for the solution of the matrix equation. Thus, the accuracy of the found solution may influence the overall accuracy.

1.7. Contribution of this thesis

In this thesis, we present a new numerical approach for computing the queue-length distribution for certain queueing models in [Part II](#) and propose models for urban traffic networks in [Part III](#). In [Subsections 1.7.1](#) and [1.7.2](#) below, we discuss the contributions of these parts. We conclude the thesis in [Chapter 7](#) with an overview of the results and future research directions. Afterwards, for convenience of the reader, we provide a glossary with the main terms used in the thesis.

1.7.1. Queueing models

As we saw in [Section 1.6](#), the classical numerical approaches to find the distribution of the queue length often include implicitly-defined variables, which prevents closed-form results. For a class of discrete-time queueing systems that have a rational form of the pgf, we present a new exact method of computing both the expectation and the distribution of the queue length using contour integrals. In [Chapter 2](#), we consider systems like the bulk-service queue and the FCTL queue, i.e., systems with pgf

$$X(z) = \frac{\sum_{k=0}^{g-1} x_k z^k (B(z))^{g-k-1}}{D(z)} f(z), \quad (1.3)$$

where x_k , $k = 0, \dots, g-1$, are unknown coefficients and $B(z)$, $D(z)$ and $f(z)$ are known (analytic) functions. For these systems, we represent the queue-length expectation in an exact closed-form expression using one contour integral. Compared to the root-finding and matrix-analytic approaches, our method turns out to be faster and more reliable.

In [Chapter 2](#), we propose several generalisations of the FCTL model, which include turning flows, drivers' distraction and actuated control at other lanes. For these generalisations, the pgf of the overflow queue length has form (1.3). The numerical results show the extent to which these additional features change the average delay per vehicle. Comparison of the classical FCTL model with its extension for the turning flow yields a decomposition result, which gives a bound on the difference between the FCTL and bulk-service queue models.

The intermediate result of [Chapter 2](#) is the representation of the numerator of (1.3) in a special product form, which we use to derive the main results. Such products often occur in discrete-time systems, see, e.g., [Bruneel and Wittevrongel \(2017\)](#), [Servi \(1986\)](#), [van Ommeren \(1991\)](#). Thus, our results can also be used for these systems. [Chapter 2](#) is based on the following paper:

Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., An exact root-free method for the expected queue length for a class of discrete-time queueing systems. *Queueing Systems*, 92 (3–4), 257–292, 2019

The queueing models proposed in [Part III](#) also lead to rational pgfs. However, the numerator is of different form than in (1.3). For this reason, we consider a more general pgf in [Chapter 3](#):

$$X(z) = \frac{\sum_{k=0}^{g-1} x_k f_k(z)}{D(z)},$$

where $f_k(z)$, $k = 0, \dots, g-1$, and $D(z)$ are known analytic functions. As in [Chapter 2](#), we show that the unknowns x_k can be computed using g contour integrals instead of finding the roots of the characteristic equation. Compared to

Chapter 2, the expectation $X'(1)$ may depend on more than one contour integral due to the general form of the numerator. We give a necessary and sufficient condition for the expectation to have a form similar to that in Chapter 2 and for the product form of the numerator mentioned above. In particular, we find a special case when the expectation does not depend on the roots. Chapter 3 is based on

Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Roots, symmetry and contour integrals in queueing systems. *Memorandum Faculty of Mathematical Sciences University of Twente*, 2067, 2019

The use of contour integrals in queueing theory dates back to works of Pollaczek, see, e.g., Pollaczek (1961). Though his results “did not receive the recognition they deserved” (Kingman (2009)), contour integrals are still used for some queueing systems, see, e.g., Al Hanbali (2011) for the analysis of the $PH/PH/1/K$ queue. Using today’s computers, it is much easier and straightforward to compute an integral than to apply the methods described in Section 1.6. Moreover, closed-form solutions give better insights in the behaviour of the system and can also be exploited to obtain structural results.

1.7.2. Urban traffic networks

In Part III, we propose analytical queueing models for urban traffic networks. To make the models more accurate than the existing ones, we consider a realistic departure process, see Subsection 1.7.2.1, and take into account the offsets between intersections and the correlation between arrivals, see Subsections 1.7.2.2 and 1.7.2.3.

1.7.2.1. Real and effective green times

Many traffic models use the effective green time instead of the real green time and assume instant acceleration, see Subsection 1.5.1.1. In these models, the inter-departure times are assumed to be constant and the vehicles leave the queue with the free-flow speed. In Figure 1.1, from observing the moments when the vehicles leave the lane, we see that the real inter-departure times are not the same. However, the vehicles pass a point 40 meters downstream the intersection at constant time intervals. Therefore, in this case, the delay is the same for the real behaviour and its approximation using instant acceleration and the effective green time.

The differences appear when the queue is or becomes empty during the green time. For example, suppose that only one vehicle (vehicle 3) arrives during a cycle in Figure 1.1. This vehicle arrives during the start-up lag and finds no queue. In reality, it would proceed without stopping. In the idealised model with effective green time, this vehicle arrives during the end of the red time, joins the queue and

Figure 1.2: The difference between real and effective green times for traffic-light networks. The real trajectories of the vehicles are given by the solid black lines. Dashed grey lines show the trajectories of vehicles when the effective green time is used instead of the real green time. The **effective red times** are represented by bold black horizontal lines. The vertical axis shows positions of the vehicles on the arterial road. When the effective green time is used, vehicle 1 is delayed and prevents vehicle 2 from proceeding without delay at the downstream intersection.

departs a bit later at the moment when vehicle 1 departs in [Figure 1.1](#). Also note that when the green time is short, the last departing vehicle during a green time does not reach the free-flow speed. Therefore, its real departure moment is earlier than the idealised one, which means that the green-end lag should be assumed longer than the part of the yellow time that is actually used by the vehicles to depart. This may lead to departures in the idealised model which are not possible in reality.

For coordinated intersections, the difference between reality and the idealised model is crucial. First, the arrivals are time-dependent and are more likely at certain periods during a cycle. Second, the delayed vehicle arrive much later than those that passed an upstream intersection without stopping, which means that, in reality, the arrivals may be spread over a longer period of time than the length of real or effective green time. Finally, a wrongly delayed vehicle in the model with effective green times arrives later at the downstream intersection, which can lead to further delay for this and other vehicles, see [Figure 1.2](#). A similar situation may happen when a vehicle is delayed in reality but is not delayed in the idealised model.

In the network models of Part III, we use the real green time instead of the effective green time. In particular, we consider longer inter-departure times at the beginning of a green time, and split the delay of a vehicle into the *waiting time*, i.e., time lost before passing the stop line, and the *acceleration delay* due to acceleration after passing the stop line. Thus, we avoid the problems mentioned above. We also take the randomness of the departures into account, which, as we show, results in a realistic departure process and accurate delay predictions.

Note that the effects described in this subsection are not directly related to the differences between *horizontal* and *vertical* queueing. For example, the description of a *traffic-light intersection* using the *shock-wave* theory, which focuses on spatial aspects of the traffic flow, also exploits the idea of the effective red and green times, see, e.g., Wu and Liu (2011). In our models, we use the vertical-queue principle, but consider a realistic departure process and include the acceleration of the vehicles that leave the intersection.

1.7.2.2. Coordination of traffic-light intersections

In traffic networks, the arrival rate at a downstream lane is time-dependent, which means that there is a higher chance of arrivals during certain time periods. If neighbouring intersections are not coordinated, it may happen that during these time periods the traffic light for the downstream lane is red, and arriving vehicles always need to stop, which causes long delays. Independent actuated control at adjacent intersections may even lead to an unstable behaviour, see Lämmer and Helbing (2008).

For *fixed* and *semi-actuated* control, the traffic-light intersections are usually coordinated via the same cycle length and properly chosen offsets. In Part III, we propose discrete-time models which include the effect of offsets. In particular, in Chapter 4, we study a tandem of two traffic-light intersections under fixed and semi-actuated control. For these controls, we analyse the effect of the traffic-light settings, i.e., (maximum) green times and offsets, on delays and emphasise the differences in the choice of optimal settings. One of the main contributions of Chapter 4 is the modelling of semi-actuated control in the case of coordinated intersections. Chapter 4 is based on

Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Comparing semi-actuated and fixed control for a tandem of two intersections. *Memorandum Faculty of Mathematical Sciences University of Twente*, 2061, 2017

In Chapter 5, we focus on coordination of the intersections under fixed control using the concept of a *green wave*, which means that the vehicles travelling along an arterial road arrive at the intersections during green time. Green waves on arterial roads increase the coordination between intersections aiming to reduce delay and the number of stops per vehicle. Coordination algorithms in the literature usually assume a deterministic flow of vehicles, whereas randomness due to, e.g., reaction

time of the drivers and their route choice may seriously affect the efficiency of green waves and significantly influence the delays. In [Chapter 5](#), we propose a stochastic [green-wave efficiency](#) measure that reflects the drivers' point of view and study the simultaneous behaviour of the green-wave efficiency and the average delay per vehicle. [Chapter 5](#) is based on

Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Green wave analysis in a tandem of traffic-light intersections. *Memorandum Faculty of Mathematical Sciences University of Twente*, 2062, 2017

In [Chapter 6](#), we consider the transient behaviour of a traffic-light network and we do not require a cyclic structure of the traffic control. In this case, the offset between the upstream and downstream lane is time-dependent. Based on the green-time schedules at every intersection, we compute the queue length distribution at each lane and at each second, providing a prediction of system response to varying arrival rates. In [Chapter 6](#), we do not limit ourselves to only traffic-light intersections but also analyse [priority intersections](#). In this way, we are able to consider realistic networks. In our numerical results, we study the behaviour of a traffic network during a rush hour using as an example a part of the city of Enschede in the Netherlands. [Chapter 6](#) is based on

Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Stochastic approximation model for urban traffic networks, 2019 (in preparation)

1.7.2.3. Arrivals correlation

To avoid the [curse of dimensionality](#), we decompose the network into individual lanes. In this decomposed network, the arrival process forms the only dependency between upstream and downstream queues. Therefore, it is important to accurately describe the arrivals at the downstream lane. Due to traffic-light intersections in the network, the arrivals at the lane during different seconds are correlated. This correlation is especially strong during an arrival of a [platoon](#), i.e., a group of vehicles which departed from an upstream lane during one green time.

In [Chapters 4](#) and [5](#), we take into account the correlation of arrivals in one cycle but assume independence of arrivals in one cycle from arrivals in the previous cycles. This approach gives accurate results except for very high loads. It is similar to the approach of [Boon and van Leeuwen \(2018\)](#), who characterise the arrival process to a lane via the joint distribution of arrivals for one cycle. We use a discrete-time [Markovian arrival process \(MAP\)](#), where arrivals to a lane are generated using an underlying Markov chain. The MAP for a lane is obtained from the departure processes of upstream lanes, taking into account acceleration time of vehicles, offsets and the vehicle routing probabilities.

The MAP approach of [Chapters 4 and 5](#) is less computationally demanding than the consideration of the joint distribution of arrivals especially for the case of small networks. For large networks, the size of the underlying Markov chain may become very big, which significantly slows down the computation speed. For this reason, we propose an approximation model in [Chapter 6](#). There, we simplify the arrival process, which makes the runtime linear in the number of lanes. In this approximation, we use dependency between consecutive arrivals to describe the correlation inside a platoon. Our model accurately predicts the evolution of the number of vehicles in the system and can be used for model-based online optimisations, as in [Hoogendoorn et al. \(2007\)](#) and [Kutadinata et al. \(2016\)](#).

Part II

Queueing

Chapter 2

An integral approach for the fixed-cycle traffic-light model

'Discontent is the first necessity of progress.'

Thomas Edison

In this chapter, we present a new exact method of computing both the expectation and the distribution of the queue length for a class of discrete-time queueing systems. This class includes among others the [FCTL](#) queue, described in [Subsection 1.5.1.1](#). For such systems, the [pgf](#) of the queue length is represented as a rational function with several unknown coefficients. The classical way of finding the unknowns is to use the roots of a certain characteristic equation, see [Section 1.6](#). The main drawback of this approach is that, in general, there is no explicit expression for the roots. Moreover, the imprecision of the found zeroes also influences the results. Another approach is to use the [matrix-analytic](#) method, see [Section 1.6](#). However, this method also requires solving a (matrix) equation and, consequently, may also suffer from imprecisions.

The key point of our method is that the unknown probabilities depend on the zeroes of the denominator in a symmetric way, and that, as we show, the actual values of the zeroes are not important. We apply the residue theorem and find these unknown probabilities in terms of contour integrals. The mean queue length is, in fact, a function of only one such contour integral. This makes our method computationally efficient in comparison with the standard [root-finding](#) method. Moreover, our method gives a closed-form solution without implicitly-defined variables such as the roots of the characteristic equation in the root-finding method or the rate matrix in the matrix-analytic approach. Thus, our solution can be used, for example, to find asymptotic results for heavy load.

The considered class of queueing systems includes the [bulk-service queue](#), see

Bailey (1954). It appears as an embedded Markov chain for the continuous-time multi-server $M/D/s$ queue, see Janssen and van Leeuwen (2008), or $M^X/G^B/1$ queue with batch arrivals and batch service. Our method is also applicable for queueing systems, for which the pgf of the analysed random variable depends on the *minimal polynomial* of the unknown zeroes, i.e., the polynomial that has the same zeroes (with the same multiplicity) and has the smallest power. An example of such a system is the discrete-time $G_D/G_D/1$ queue, see van Ommeren (1991).

In this chapter, we also consider a modification of the FCTL model for a lane with turning flow. For this new model, we prove a decomposition result similar to Boxma and Groenendijk (1987). Moreover, we prove that given the same arrival process and server capacity, the expected bulk-service queue is bounded from below by the expected *overflow queue* length for the FCTL model and from above by the expected overflow queue length for the FCTL model for turning flow, where the overflow queue is the queue just after service. These results give us a bound on the difference between these models. We propose several other extensions of the FCTL queue, which include the effects of traffic interruption, for example, due to actuated control for pedestrians or uncertain departure times.

In the numerical results, we compare our method with the root-finding and matrix-analytic approaches. We observe that the root-finding approach can be numerically quite unreliable, while the matrix-analytic approach is much slower than our method. We also apply our method to the FCTL queue and its extensions. In particular, we investigate the impact of the variability of the arrival process on the expected queue length and consider the difference between straight-going and turning flows. In addition, we study effects of traffic disruption.

This chapter is structured as follows. The method in its general form is presented in Section 2.1. In Section 2.2, we apply our method to the bulk-service queue and compare it with the root-finding and matrix-analytic approaches in terms of speed and reliability. In Section 2.3, we present several realistic extensions of the FCTL model. In Section 2.4, we conclude the chapter.

2.1. Our method

In this section, we propose a generic method to analyse various discrete-time queueing systems, such as the FCTL queue and the bulk-service queue. For these systems, the pgf of the queue length is a rational function with unknown probabilities in the numerator. The classical way of determining the unknowns requires finding the roots of the characteristic equation. In our method, we use the special structure of the numerator for these systems to find an alternative root-free solution. Let us consider the function $X(z)$ of the following form:

$$X(z) = \frac{\sum_{k=0}^{g-1} x_k z^k (B(z))^{g-1-k}}{D(z)} f(z), \quad (2.1)$$

where x_k are unknown coefficients that we want to determine; $D(z)$, $B(z)$ and $f(z)$ are known functions such that $B(1) = 1$, $f(1) = 0$, and, for each zero $\bar{z}_l \in \{z: |z| \leq 1, z \neq 1\}$ of the denominator, $f(\bar{z}_l) \neq 0$; and $X(1) = 1$. For the analysis in this section, we do not need $X(z)$ to be a pgf. We use the following assumptions on the functions $B(z)$, $D(z)$, $f(z)$ and $X(z)$:

Assumption 2.1 (Analyticity assumption). For some $\epsilon > 0$, the functions $B(z)$ and $D(z)$ are analytic and the function $f(z)$ is two times differentiable in the disk $D_{1+\epsilon} = \{z: |z| < 1 + \epsilon\}$. The function $X(z)$ is continuous in the closed unit disk $\bar{D}_1 = \{z: |z| \leq 1\}$.

Assumption 2.2 (Roots assumption). The equation

$$D(z) = 0 \tag{2.2}$$

has exactly g roots $\bar{z}_0 = 1, \bar{z}_1, \dots, \bar{z}_{g-1}$ in the disk $\bar{D}_1$. For any $z \neq w$, $z, w \in \bar{D}_1$, the function $B(z)$ satisfies

$$B(z)w \neq B(w)z.$$

Note that from [Assumption 2.2](#), the following statement is deduced immediately.

Corollary 2.3. Equation $z = B(z)$ has only one root in $\bar{D}_1$, namely 1.

Remark 2.4 (On the value of $D(0)$). Without loss of generality, we can assume that $D(0) \neq 0$. Otherwise, both denominator and numerator can be divided by z^n for some $n > 0$ such that the new $D(z)$ satisfies the condition $D(0) \neq 0$.

2.1.1. Coefficients expressed in terms of roots

In this subsection, we discuss how coefficients x_k , $k = 0, \dots, g-1$, depend on the roots of equation (2.2).

Let $y_k = B(\bar{z}_k)/\bar{z}_k$ for $k = 0, \dots, g-1$. Since we know that $D(0) \neq 0$, then $\bar{z}_k \neq 0$ for each $k = 0, \dots, g-1$. Rewrite $X(z)$ in the following way:

$$X(z) = \frac{\sum_{k=0}^{g-1} x_k \left(\frac{B(z)}{z} \right)^{g-1-k}}{D(z)} f(z) z^{g-1}.$$

Since $\bar{z}_k \neq 0$ and $f(\bar{z}_k) \neq 0$, $k = 1, \dots, g-1$, the numerator is equal to 0 for $z = \bar{z}_k$ and

$$\sum_{l=0}^{g-1} x_l y_k^{g-1-l} = 0.$$

Consider the following polynomial:

$$h(y) = \sum_{l=0}^{g-1} x_l y^{g-1-l} = \sum_{l=0}^{g-1} x_{g-1-l} y^l.$$

The function $h(y)$ is a polynomial of degree $g - 1$, and y_k for $k = 1, \dots, g - 1$ are zeroes of this polynomial. Note that whenever $\bar{z}_k$ is a multiple root of the equation $D(z) = 0$, y_k is a multiple root of the polynomial $h(y)$ with the same multiplicity. Moreover, according to [Assumption 2.2](#), $y_k \neq y_l$ if $\bar{z}_k \neq \bar{z}_l$. Thus, $h(y)$ can be written as

$$h(y) = x_0 \prod_{k=1}^{g-1} (y - y_k). \quad (2.3)$$

By applying Vieta's formulas (see [Vinberg \(2003\)](#)) to the polynomial $h(y)$, we get for $k = 1, \dots, g - 1$ that

$$\frac{x_k}{x_0} = (-1)^k \sigma_k(y_1, \dots, y_{g-1}), \quad (2.4)$$

where

$$\sigma_k(y_1, \dots, y_{g-1}) = \sigma_k = \sum_{1 \leq i_1 < \dots < i_k \leq g-1} y_{i_1} \dots y_{i_k}$$

are [elementary symmetric polynomials](#). Here for notational convenience we assume

$$\sigma_0(y_1, \dots, y_{g-1}) = \sigma_0 = 1.$$

Equation (2.4) gives us x_k up to a normalization constant, which we derive from the equation $X(1) = 1$ by applying L'Hôpital's rule:

$$X(1) = \frac{h(1)}{D'(1)} f'(1) = \frac{\sum_{k=0}^{g-1} x_k}{D'(1)} f'(1) = 1. \quad (2.5)$$

2.1.2. Coefficients expressed in terms of contour integrals

Using (2.4), we can find the coefficients x_k if we know the values of the elementary symmetric polynomials σ_k . We will represent these symmetric polynomials as functions of the [power-sum symmetric polynomials](#):

$$\eta_k = \eta_k(y_1, \dots, y_{g-1}) = \sum_{l=1}^{g-1} y_l^k$$

for $k = 1, \dots, g - 1$. Below, in [Theorem 2.6](#), we provide a way to find η_k without computing the roots of (2.2). The proof of the theorem requires the following auxiliary result, which can be obtained using L'Hôpital's rule.

Lemma 2.5. *Consider a zero $\bar{z}$ of the function $D(z)$ and some function $F(z)$ that is analytic in a neighbourhood of $\bar{z}$. Then the residue of the function $\frac{D'(z)}{D(z)} F(z)$ at $\bar{z}$ is equal to*

$$\lim_{z \rightarrow \bar{z}} \frac{D'(z) F(z) (z - \bar{z})}{D(z)} = m_{\bar{z}} F(\bar{z}),$$

where $m_{\bar{z}}$ is the multiplicity of the zero $\bar{z}$.

Figure 2.1: Circle $S_{1+\epsilon}$ such that there are no roots of equation (2.2) in $\bar{D}_{1+\epsilon} \setminus \bar{D}_1$. The roots of the equation $D(z) = z^5 - (z/2 + 1/2)^9 = 0$ are represented as bold black points, and 0 is shown as the bold grey point.

Theorem 2.6. Consider $\epsilon > 0$ and $\delta > 0$ such that $\epsilon < \epsilon$, and there are no roots of equation (2.2) in ring $\bar{D}_{1+\epsilon} \setminus \bar{D}_1$ and disk $\bar{D}_\delta$. Then,

$$\eta_k = \frac{1}{2\pi i} \oint_{S_{1+\epsilon}} \frac{D'(z)}{D(z)} \left(\frac{B(z)}{z} \right)^k dz - \frac{1}{2\pi i} \oint_{S_\delta} \frac{D'(z)}{D(z)} \left(\frac{B(z)}{z} \right)^k dz - 1, \quad (2.6)$$

where $S_r = \{z: |z| = r\}$.

Proof. Note that the only singularities of the function $\frac{D'(z)}{D(z)} \left(\frac{B(z)}{z} \right)^k$ in $D_{1+\epsilon}$ are $0, 1, \bar{z}_1, \dots, \bar{z}_{g-1}$, see Figure 2.1. Thus, by the residue theorem and Lemma 2.5, the difference of the integrals on the right-hand side of (2.6) is equal to

$$\sum_{l=0}^{g-1} y_l^k + r_0 - r_0 = y_0^k + \eta_k = \eta_k + 1,$$

where r_0 is the residue of the function $\frac{D'(z)}{D(z)} \left(\frac{B(z)}{z} \right)^k$ at 0. □

Remark 2.7 (Values of δ , ϵ and ϵ). In many applications, ϵ is very big or infinite, and, therefore, we omit requirement $\epsilon < \epsilon$ hereafter. In Appendix 2.B, we elaborate on how to choose ϵ and δ .

The following lemma provides a way of finding σ_k , $k = 1, \dots, g-1$, using η_k , $k = 1, \dots, g-1$. We omit the proof of this lemma since it requires only careful computation of the monomials' coefficients in the right side of equation (2.7), see Mead (1992).

Lemma 2.8 (Newton's formula). *For each $k = 1, \dots, g-1$, the following recurrence equation holds:*

$$\sigma_k = \frac{1}{k} \sum_{l=1}^k (-1)^{l+1} \sigma_{k-l} \eta_l. \quad (2.7)$$

To find the coefficients x_k , $k = 0, \dots, g-1$, in a root-free way, we use equations (2.4), (2.5), (2.6) and (2.7).

2.1.3. Expectation expressed in terms of roots

Now consider $X'(1)$. For queueing systems, it represents the expected queue length.

$$\begin{aligned} X'(1) &= \left(\frac{\sum_{k=0}^{g-1} x_k z^k (B(z))^{g-1-k}}{D(z)} f(z) \right)' \Big|_{z=1} = \\ &= \frac{f'(1)}{D'(1)} \cdot \left(h \left(\frac{B(z)}{z} \right) z^{g-1} \right)' \Big|_{z=1} + h(1) \cdot \left(\frac{f(z)}{D(z)} \right)' \Big|_{z=1} = \\ &= \frac{f'(1)}{D'(1)} (h'(1)(B'(1) - 1) + (g-1)h(1)) + \\ &\quad + h(1) \frac{f''(1)D'(1) - f'(1)D''(1)}{2(D'(1))^2}. \end{aligned}$$

From (2.5), we observe that $f'(1)/D'(1) = 1/h(1)$. Thus,

$$X'(1) = (B'(1) - 1) \frac{h'(1)}{h(1)} + g - 1 + \frac{f''(1)D'(1) - f'(1)D''(1)}{2D'(1)f'(1)}.$$

The remaining term we need to find here is $h'(1)/h(1)$. By using representation (2.3), we get that

$$\frac{h'(1)}{h(1)} = \frac{\left(\prod_{k=1}^{g-1} (y - y_k) \right)' \Big|_{y=1}}{\prod_{k=1}^{g-1} (y - y_k)} = \sum_{k=1}^{g-1} \frac{1}{1 - y_k}. \quad (2.8)$$

Thus, in terms of the roots, $X'(1)$ can be found using

$$X'(1) = (B'(1) - 1) \sum_{k=1}^{g-1} \frac{\bar{z}_k}{\bar{z}_k - B(\bar{z}_k)} + g - 1 + \frac{f''(1)}{2f'(1)} - \frac{D''(1)}{2D'(1)}. \quad (2.9)$$

2.1.4. Expectation expressed in terms of contour integrals

We can use a contour integral to find $\sum_{k=1}^{g-1} \bar{z}_k / (\bar{z}_k - B(\bar{z}_k))$. Let ε be as above, then

$$\sum_{k=1}^{g-1} \frac{\bar{z}_k}{\bar{z}_k - B(\bar{z}_k)} = \frac{1}{2\pi i} \oint_{S_{1+\varepsilon}} \frac{D'(z)}{D(z)} \frac{z}{z - B(z)} dz - r_1, \quad (2.10)$$

where

$$r_1 = \frac{1}{1 - B'(1)} \left(1 + \frac{B''(1)}{2(1 - B'(1))} + \frac{D''(1)}{2D'(1)} \right)$$

is the residue of the function $\frac{D'(z)}{D(z)} \frac{z}{z - B(z)}$ at 1. By [Corollary 2.3](#), we know that there are no roots of the equation $z = B(z)$ inside $\bar{D}_1$ except 1. Thus, we do not need to subtract any other residues of the function inside the integral.

Remark 2.9 (Beyond the expectation). When $X(z)$ is a [pgf](#) of the queue length, it is also possible to find the variance of the queue length, i.e., $X''(1) + X'(1) - (X'(1))^2$, in the same way, but since it is a lengthy expression, we omit it here. The application of contour integrals goes further. Under certain conditions, the function $X(z)$ can be written as an exponent of a contour integral yielding a Pollaczek integral, see [Boon et al. \(2019\)](#), where authors use our factorisation result [\(2.3\)](#).

Note that the expectation $X'(1)$ represents the average queue length and is a real number. Thus, for $B'(1) \in \mathbb{R}$, only the real part of the integral [\(2.10\)](#) should be computed.

2.1.5. Algorithms

In this subsection, we summarise our contour-integral algorithms to compute $X'(1)$ and the coefficients x_k using the results of [Subsections 2.1.1 - 2.1.4](#). For computational questions such as how to choose parameters ε and δ or how to compute a complex integral we refer to [Appendix 2.B](#).

Algorithm 2.10 (Computation of $X'(1)$).

1. Find $\varepsilon > 0$ such that there are only g roots of equation [\(2.2\)](#) in the disk $\bar{D}_{1+\varepsilon}$, using one of two ways described in [Subsection 2.B.3](#).
2. Compute (the real part of) the integral

$$I = \int_{-\pi}^{\pi} z(\varphi) \frac{D'(z(\varphi))}{D(z(\varphi))} \frac{z(\varphi)}{z(\varphi) - B(z(\varphi))} d\varphi,$$

where $z(\varphi) = (1 + \varepsilon)e^{i\varphi}$.

3. Compute $X'(1)$ using

$$X'(1) = \frac{(B'(1) - 1)}{2\pi} I + \frac{B''(1)}{2(1 - B'(1))} + g + \frac{f''(1)}{2f'(1)}. \quad (2.11)$$

Remark 2.11 (Parametrisation of $S_{1+\varepsilon}$). Note that we parametrised $S_{1+\varepsilon}$ by $z(\varphi) = (1 + \varepsilon)e^{i\varphi}$. Therefore, $dz = izd\varphi$ and

$$\oint_{S_{1+\varepsilon}} F(z) dz = i \int_{-\pi}^{\pi} F(z(\varphi)) z(\varphi) d\varphi,$$

where $F(z)$ represents the integration function.

Algorithm 2.12 (Computation of x_k , $k = 0, \dots, g-1$).

1. Find $\varepsilon > 0$ such that there are only g roots of equation (2.2) in the disk $\bar{D}_{1+\varepsilon}$, using one of two ways described in Subsection 2.B.3.
2. Find $\delta > 0$ such that there are no roots of equation (2.2) in the disk $\bar{D}_\delta$, using one of two ways described in Subsection 2.B.2.
3. Compute η_k , using

$$\eta_k = -1 + \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[z \frac{D'(z)}{D(z)} \left(\frac{B(z)}{z} \right)^k - w \frac{D'(w)}{D(w)} \left(\frac{B(w)}{w} \right)^k \right] d\varphi,$$

where for convenience $z = z(\varphi) = (1 + \varepsilon)e^{i\varphi}$, $w = w(\varphi) = \delta e^{i\varphi}$.

4. Compute σ_k from η_k using (2.7).
5. Compute x_k , $k = 0, \dots, g-1$, using (2.4) and (2.5).

In Appendix 2.B.1, we consider a simplified algorithm for the case when $B(z)$ has no roots inside $\bar{D}_1$, which is equivalent to $B(-1) > 0$ if $B(z)$ is a pgf. Fortunately, in many applications, the function $B(z)$ has this property. Therefore, it is often better to use the simplified algorithm as described in Appendix 2.B.1.

2.1.6. Method application in queueing theory

In this subsection, we give some remarks concerning the application of our method for queueing models. First, we consider a special subclass of functions $X(z)$ of form (2.1) for which it is easy to check Assumption 2.2. This subclass generalises the form of the queue-length pgf for the bulk-service and FCTL-type queues considered in Sections 2.2 and 2.3. Then, we describe the application to queueing systems that do not have the pgf of form (2.1).

Suppose that

$$D(z) = z^g - A(z) \tag{2.12}$$

and the functions $A(z)$, $B(z)$ are pgfs. Consider the following assumption.

Assumption 2.13 (Stability assumption). The functions $A(z)$ and $B(z)$ are pgfs and satisfy $A'(1) < g$ and $B'(1) < 1$.

The following two lemmas show that Assumptions 2.1 and 2.13 imply Assumption 2.2. Therefore, our method can be applied in the case that the denominator of $X(z)$ has form (2.12), and Assumptions 2.1 and 2.13 hold. Note that under Assumption 2.13, $D'(1) = g - A'(1) > 0$, which means that the root $\bar{z}_0 = 1$ has multiplicity 1.

Lemma 2.14. *Consider a pgf $A(z)$ that is analytic in the disk $D_{1+\epsilon}$ for some $\epsilon > 0$. Suppose that $A'(1) < g$. Then, the equation $z^g - A(z) = 0$ has exactly g roots $\bar{z}_0 = 1, \bar{z}_1, \dots, \bar{z}_{g-1}$ in $\bar{D}_1$.*

Lemma 2.15. *Consider a pgf $B(z)$ that is analytic in the disk $D_{1+\epsilon}$ for some $\epsilon > 0$. Suppose that $B'(1) < 1$. If $z \neq w$ and $z, w \in \bar{D}_1$, then*

$$zB(w) \neq wB(z).$$

For the proof of Lemma 2.14, we refer to [Adan et al. \(2006\)](#). The proof of Lemma 2.15 is given in [Appendix 2.A](#).

Remark 2.16 (On assumption $B'(1) < 1$). In our examples, the second part, $B'(1) < 1$, of the stability assumption will follow from the first part, $A'(1) < g$. However, for the general case, we have to assume that $B'(1) < 1$ to prove the second part of [Assumption 2.2](#), see the proof of [Lemma 2.15](#).

Note that for many queueing systems, it was proven that the unknown part of the pgf can be expressed in terms of the function

$$\bar{h}(z) = \prod_{k=1}^{g-1} \frac{z - \bar{z}_k}{1 - \bar{z}_k},$$

where $\bar{z}_k, k = 1, \dots, g-1$ are the only zeroes in $\bar{D}_1 \setminus \{1\}$ of a certain analytic function. For such systems, our approach can be used to find the pgf, see equations (2.4), (2.6) and (2.7), or the expectation, see equations (2.8) and (2.10). One of the examples is a discrete-time $G_D/G_D/1$ queue, see [van Ommeren \(1991\)](#). For the $G_D/G_D/1$ queue, the generating function of the limiting waiting-time distribution depends either on $\bar{h}(z)$ or on $1/(z^g \bar{h}(1/z))$, depending on whether the pgf of the inter-arrival or service time is a rational function, i.e., is equal to $A_1(z)/A_2(z)$, where $A_1(z)$ and $A_2(z)$ are polynomials of (at most) power g . Similar results hold for the queue-length pgf in case of a bulk-service queue with a random bulk size. The reason for this is that these two systems are governed by the same Lindley-type equation

$$X_{n+1} = [X_n + A_n - B_n]^+,$$

where $x^+ = \max\{x, 0\}$. For the bulk-service queue, X_n is the queue length at time n , A_n and B_n are the numbers of arrivals and departures at time n , respectively. For the $G_D/G_D/1$ queue, X_n and A_n are the [waiting time](#) and the service time of the n^{th} customer, respectively, and B_n is the inter-arrival time between the n^{th} and $(n+1)^{\text{st}}$ customers.

2.2. Method comparison for the bulk-service queue

In this section, we consider a discrete-time [bulk-service queue](#), see [Bailey \(1954\)](#). For this example, we compare our contour-integral approach in terms of speed

and reliability with the root-finding and matrix-analytic methods. In [Subsection 2.2.1](#), we describe the model. Then, we apply the considered methods in [Subsections 2.2.2-2.2.4](#). Finally, we present the numerical results in [Subsection 2.2.5](#) and discuss the differences between the approaches in [Subsection 2.2.6](#).

2.2.1. Discrete bulk-service queue

Consider a discrete-time queue with bulk service. In this model, the time is split in time units. During each time unit the server serves g customers in the queue and A_b new customers arrive with a pgf $A_b(z)$. Only those customers are served that are present in the queue at the beginning of the time unit. If there are less than g customers in the queue, all of them are served.

Denote by $X_b(z)$ the pgf of the queue length at the beginning of a time unit in the stationary state. Let q_k be the stationary probability of having k customers in the queue at the beginning of a time unit. Then, (see e.g., [Bailey \(1954\)](#)) we obtain:

$$X_b(z) = \frac{\sum_{k=0}^{g-1} q_k (z^g - z^k)}{z^g - A_b(z)} A_b(z). \quad (2.13)$$

The stability condition in this model is $A'_b(1) < g$. If we consider the pgf $X_b^-(z)$ of the queue length after service but before arrivals, we get

$$X_b^-(z) = \frac{\sum_{k=0}^{g-1} q_k (z^g - z^k)}{z^g - A_b(z)}.$$

In the following subsections, we apply our method, the root-finding and matrix-analytic approaches to find the expectation $(X_b^-)'(1) = X'_b(1) - A'_b(1)$.

2.2.2. Our method

At first glance, the pgf of the queue length, see [\(2.13\)](#), does not satisfy the form [\(2.1\)](#), but we can rearrange it as follows:

$$X_b(z) = \frac{\sum_{k=0}^{g-1} q_k (z^g - z^k)}{z^g - A_b(z)} A_b(z) = \frac{\sum_{l=0}^{g-1} (\sum_{k=0}^l q_k) z^l}{z^g - A_b(z)} (z-1) A_b(z).$$

Hence, we take $A(z) = A_b(z)$, $B(z) = 1$, $f(z) = (z-1)A_b(z)$ and $x_k = \sum_{l=0}^k q_l$. Note that $q_k = x_k - x_{k-1}$, $k = 2, \dots, g-1$, and $q_1 = x_1$. In case $A_b(0) = 0$, we can reduce the complexity as before, since q_l , as the probability of having l customers in the queue, is zero for $l = 0, \dots, n-1$, where n is the least possible number of customers that arrive in one time slot. Thus, we suppose that $A_b(0) \neq 0$. Hence, $\bar{z}_k \neq 0$ for every $k = 1, \dots, g-1$ and

$$f(\bar{z}_k) = A_b(\bar{z}_k)(\bar{z}_k - 1) = \bar{z}_k^g(\bar{z}_k - 1) \neq 0.$$

Here we used the fact that the root $\bar{z}_0 = 1$ has multiplicity 1, and, therefore, $\bar{z}_k \neq 1$ for $k = 1, \dots, g-1$. If the system is stable, then $A'(1) = A'_b(1) < g$. Since $B'(1) = 0 < 1$, the stability assumption holds. Therefore, we can use our method if $A_b(z)$ is analytic in the disk $D_{1+\epsilon}$ for some $\epsilon > 0$. For the case of $X_b^-(z)$, the only change would be that $f(z) = z - 1$.

From (2.11), we get that the expected queue length is given by the following formula:

$$X'_b(1) = -\frac{I}{2\pi} + g + A'_b(1), \quad (2.14)$$

where I is given by

$$I = \int_{-\pi}^{\pi} \frac{gz^g - zA'_b(z)}{z^g - A_b(z)} \frac{z}{z-1} d\varphi, \quad (2.15)$$

$z = z(\varphi) = (1 + \varepsilon)e^{i\varphi}$, and ε is chosen as in Theorem 2.6. Note that (2.14) and (2.15) are equivalent to the following formula:

$$X'_b(1) = A'_b(1) - \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{gz^g - zA'_b(z)}{z^g - A_b(z)} \frac{1}{z-1} d\varphi.$$

Thus, for the case of $X_b^-(z)$, we get

$$(X_b^-)'(1) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{gz^g - zA'_b(z)}{z^g - A_b(z)} \frac{1}{1-z} d\varphi. \quad (2.16)$$

2.2.3. Root-finding approach

The root-finding approach requires a numerical algorithm to find the roots of (2.2). Then, one needs to pick the roots that are inside the unit circle, namely, roots $\bar{z}_1, \dots, \bar{z}_{g-1}$. The coefficients x_k in (2.1) then follow from a linear system of equations. For the case of bulk-service queue, we get

$$\begin{pmatrix} 0 & 1 & \dots & g-1 \\ 1 & \bar{z}_1 & \dots & \bar{z}_1^{g-1} \\ \dots & \dots & \dots & \dots \\ 1 & \bar{z}_{g-1} & \dots & \bar{z}_{g-1}^{g-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \dots \\ x_{g-1} \end{pmatrix} = \begin{pmatrix} g - A'_b(1) \\ 0 \\ \dots \\ 0 \end{pmatrix}. \quad (2.17)$$

Note that the matrix is singular if two roots coincide. If the roots are close, the resulting values of x_l depend heavily on the precision of the roots. Given values of x_l , we compute the probabilities $q_k = x_k - x_{k-1}$, where $x_{-1} = 0$, and find the expected queue length as follows:

$$(X_b^-)'(1) = \frac{\sum_{k=0}^{g-1} q_k(g(g-1) - k(k-1))}{2(g - A'_b(1))} - \frac{g(g-1) - A''_b(1)}{2(g - A'_b(1))}.$$

Alternatively to solving (2.17), we can directly use roots, see equation (2.9). For the bulk-service, we get

$$(X_b^-)'(1) = - \sum_{k=1}^{g-1} \frac{\bar{z}_k}{\bar{z}_k - 1} + g - 1 - \frac{g(g-1) - A_b''(1)}{2(g - A_b'(1))}. \quad (2.18)$$

The possible drawbacks of using the roots in (2.17) or (2.18) are that the root-finding algorithm can fail to find exactly $g-1$ roots inside the unit circle, and that the precision of the roots can be unsatisfactory.

2.2.4. Matrix-analytic approach

It is possible to represent the bulk-service queue as an $M/G/1$ -type queue. Consider the Markov chain of the queue length at the service completion, i.e., before the arrivals. Then, the transition matrix is a block matrix

$$P = \begin{pmatrix} B_0 & A_2 & A_3 & \dots \\ A_0 & A_1 & A_2 & \dots \\ 0 & A_0 & A_1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where

$$B_0 = \begin{pmatrix} \sum_{i=0}^g a_i & a_{g+1} & \dots & a_{2g-1} \\ \sum_{i=0}^{g-1} a_i & a_g & \dots & a_{2g-2} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^2 a_i & a_2 & \dots & a_g \end{pmatrix}, \quad A_j = \begin{pmatrix} a_{jg} & a_{jg+1} & \dots & a_{jg+g-1} \\ a_{jg-1} & a_{jg} & \dots & a_{jg+g-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{jg-g+1} & a_{jg-g+2} & \dots & a_{jg} \end{pmatrix},$$

and a_k is the probability of having k arrivals with $a_k = 0$ for $k < 0$. Using the notation for $M/G/1$ -type queues, instead of queue length l , we consider a two-dimensional process (n, m) , where the *level state* n is equal to an integer part of quotient l/g , i.e., $n = \lfloor l/g \rfloor$, and the *background state* m is the remainder of such a division, i.e., $m = l - \lfloor l/g \rfloor g$.

The steady-state distribution is found using the matrix G defined as a minimal non-negative solution of the following equation:

$$G = \sum_{j=0}^{\infty} A_j G^j.$$

For numerical computation, this infinite summation should be truncated. Consider the stationary probabilities $\pi_n = (\pi_{ng}, \dots, \pi_{ng+g-1})$ for level n , where π_k is the probability of having k customers in the queue. These probabilities are computed recursively using Ramaswami's formula, see He (2014):

$$\pi_n = \sum_{k=0}^{n-1} \pi_k \bar{A}_{n+1-k} (I - \bar{A}_1)^{-1}, \quad n = 1, 2, \dots,$$

where

$$\bar{A}_n = \sum_{k=n}^{\infty} A_{n+k} G^k,$$

and $\boldsymbol{\pi}_0$ follows from the following system of linear equations:

$$\begin{aligned} \boldsymbol{\pi}_0 \bar{B}_0 &= \boldsymbol{\pi}_0, \\ \boldsymbol{\pi}_0 \left(\mathbf{e} + \mathbf{e} \left(I - \sum_{i=1}^{\infty} \bar{A}_i \right)^{-1} \sum_{i=1}^{\infty} \bar{A}_i \right)^T &= 1, \end{aligned}$$

with $\bar{B}_0 = B_0 + \bar{A}_2 G$ and $\mathbf{e} = (1, \dots, 1)$. The average queue length is approximated by

$$(X_b^-)'(1) \approx \sum_{k=1}^{gN} k \pi_k, \quad (2.19)$$

where the value for N can be chosen so that the contribution of vector $\boldsymbol{\pi}_N$ is small, i.e.,

$$\sum_{k=0}^{g-1} (gN + k) \pi_{gN+k} < \varepsilon_M.$$

In Riska and Smirni (2002), an alternative way was proposed to find the average queue length for $M/G/1$ -type queues. For the bulk-service queue, we can use a simplified version of this approach. The average queue length is computed using $\boldsymbol{\pi}_0$ and $\boldsymbol{\pi}_* = \sum_{i=1}^{\infty} \boldsymbol{\pi}_i$, which can be found from the following system of linear equations:

$$\begin{aligned} \boldsymbol{\pi}_0 \mathbf{e}^T + \boldsymbol{\pi}_* \mathbf{e}^T &= 1, \\ \boldsymbol{\pi}_0 \left(B_0 - \sum_{n=3}^{\infty} \bar{A}_n G \right) + \boldsymbol{\pi}_* \left(A_0 - \sum_{n=2}^{\infty} \bar{A}_n G \right) &= \boldsymbol{\pi}_0, \\ \boldsymbol{\pi}_0 \sum_{n=2}^{\infty} \bar{A}_n + \boldsymbol{\pi}_* \sum_{n=1}^{\infty} \bar{A}_n &= \boldsymbol{\pi}_*. \end{aligned}$$

Note that the average queue length is the sum of the average level and the average background state, i.e.,

$$(X_b^-)'(1) = \sum_{k=1}^{\infty} g k \pi_k \mathbf{e}^T + \boldsymbol{\pi}_0 \mathbf{b}^T + \boldsymbol{\pi}_* \mathbf{b}^T, \quad (2.20)$$

where $\mathbf{b} = (0, 1, \dots, g-1)$. Thus, it is only left to find $\mathbf{r} = \sum_{k=1}^{\infty} k \pi_k$ using

$$\mathbf{r} \left(I - \sum_{k=0}^{\infty} A_k \right) = \boldsymbol{\pi}_0 \left(\sum_{k=2}^{\infty} (k-1) A_k \right) + \boldsymbol{\pi}_* \left(\sum_{k=2}^{\infty} (k-1) A_k - A_0 \right),$$

Methods	Root- -finding (2.17)	Root- -finding (2.18)	Matrix- -analytic (2.19)	Matrix- -analytic (2.20)	Our approach (2.16)
Speed (ms. per run)	7.15	3.37	17.73	17.69	0.99
Absolute difference:					
> 0.0001	937	95	4222	0	0
> 0.5	305	0	1018	0	0
Total failures:	1179	389	0	0	0
wrong number of roots	329	329	N.A.	N.A.	N.A.
$\Re(\mathbb{E}L) < -0.0001$	241	60	0	0	0
$\Re(\mathbb{E}L) < -0.5$	112	0	0	0	0
$ \Im(\mathbb{E}L) > 0.0001$	788	0	0	0	0
$ \Im(\mathbb{E}L) > 0.5$	258	0	0	0	0

Table 2.1: Comparison of speed and reliability for the expected queue length. There are 10000 random combinations of g , c and λ . For the speed reference of our method, only the real part of the integral was calculated. The real and imaginary parts of the expected queue length are denoted by $\Re(\mathbb{E}L)$ and $\Im(\mathbb{E}L)$, respectively. The absolute difference is calculated between our and other approaches.

$$\mathbf{r} \left(A_0 - \sum_{k=2}^{\infty} (k-1) A_k \right) \mathbf{e}^T = (\boldsymbol{\pi}_0 + \boldsymbol{\pi}_*) \sum_{k=2}^{\infty} \frac{k(k-1)}{2} A_k \mathbf{e}^T.$$

In the following subsection, we numerically compare the considered approaches in terms of speed and reliability.

2.2.5. Numerical results

In this subsection, we compare our approach with the [root-finding](#) and [matrix-analytic](#) methods for the bulk-service queue. For these methods, we focus on their speed and reliability, i.e., absence of failures such as wrong number of roots or unrealistic result (negative, imaginary or not accurate queue length), see [Table 2.1](#). We consider Binomial arrivals with $A_b(z) = (\rho g z / c + 1 - \rho g / c)^c$, where parameter $\rho = A'_b(1)/g$ represents the [load](#). We generate 10000 triples (g, c, ρ) of random parameters $g \in \{2, \dots, 30\}$, $c \in \{g+1, \dots, 70\}$ and $\rho \in [0, 0.99]$. We used standard *python* methods for computing integrals, finding polynomial roots and solving linear systems of equations.

For the root-finding technique, the use of the linear system [\(2.17\)](#) leads to poor reliability. In almost 12% of the cases, the approach failed to give a realistic result, where in 3% of the cases the number of roots in the unit circle was wrong, and there was no result at all. The root-finding algorithm together with [\(2.18\)](#) is two times faster and fails in only 4% of the cases. Note that the failures occur more

Figure 2.2: The percentage of failed runs for root-finding algorithm depending on the batch size g .

often for large batch sizes, see Figure 2.2. For a batch size of 30, the root-finding method with (2.17) fails in more than half of the cases. Our approach has no fails for all values of batch sizes and produces realistic results.

The matrix-analytic approach always produces non-negative queue length. However, the use of (2.19) leads to a serious underestimation of the queue length in 10% of the cases. This happens due to slow convergence of the distribution tail for high loads ($\rho > 0.9$). The calculation of the aggregated distribution (given by π_0, π_*) leads to results that (almost) coincide with the results of our approach, suggesting that both approaches produce accurate results. The main drawback is the slow computation speed, which is mainly due to the computation of the matrix G , which we find by recursively applying

$$G_{n+1} = \sum_{k=0}^{\infty} A_k G_n^k, \quad n = 0, 1, \dots$$

with $G_0 = 0$, until absolute values of entries in the matrix $G_{n+1} - G_n$ are less than $\varepsilon_M = 0.1^{10}$. The runtime is longer for bigger sizes of the matrices, i.e., for bigger g , see Figure 2.3.

We conclude that our contour-integral method outperforms the considered approaches both in speed and reliability. In the following subsection, we summarise the advantages of our method compared to the existing approaches.

Figure 2.3: The average runtime as a function of g for the matrix-analytic approach using (2.19), T_D , or (2.20), T_A , and for our integral approach (2.16), T_I .

2.2.6. Discussion

The main contribution of the chapter is our *explicit* method of finding the queue-length distribution. Both the root-finding approach and the matrix-analytic approach require the computation of *implicitly* defined variables, i.e., the roots $\bar{z}_1, \dots, \bar{z}_{g-1}$ or the matrix G . For the root-finding approach, this step is error-prone resulting in poor reliability of the method, while for the matrix-analytic approach the main issue is the speed. Note that explicit closed-form results are much easier to manipulate to show a number of structural results such as convexity and sensitivity to input parameters or asymptotic behaviour.

In Subsection 2.2.5, we used a simple example of a bulk-service queue with Binomial arrivals. For such arrivals, the pgf $A_b(z)$ is a polynomial, and $A_n = 0$ for $n \geq c/g + 1$. This allowed us to use standard *python* methods to find the roots of the polynomials and avoid truncating the arrival distribution in the definition of the matrix G , which could have resulted in an underestimation of the queue length. For our approach, the required pgf $A_b(z)$ is usually easy to compute even if it is not a polynomial, see Subsection 2.3.5 below. Therefore, we do not have to choose a truncation bound, as in the existing methods.

We note that for certain arrival distributions, there are formulas to compute the roots, see, e.g., Janssen and van Leeuwen (2008). These formulas contain infinite summations, which should be truncated for numerical results. As we saw in Subsection 2.2.5, a poor precision of the roots may lead to unrealistic results, which can also happen in case of a badly-chosen truncation bound. For

the case of the bulk-service queue, root-free results were obtained in [Janssen and van Leeuwen \(2005\)](#). The average queue length is represented as a (double) infinite summation, which also leads to a truncation problem.

Our results are applicable to a quite general class of queueing systems. It includes several $M/G/1$ -type queues, such as bulk-service queue and FCTL-type queues described in [Section 2.3](#), but also discrete $G_D/G_D/1$ queues, see [Subsection 2.1.6](#), for which the matrix-analytic approach is not applicable.

2.3. FCTL-type queues

In this section, we consider several traffic-light models and apply our method to them. First, we explain the basic FCTL model in detail and use our approach to find the average queue length, see [Subsection 2.3.1](#). Then, we propose several modifications of this model to include the following realistic situations:

1. the lane serves a turning flow, so vehicles slow down before departing;
2. pedestrian and/or bicycle traffic lights have an actuated control;
3. just after the intersection there is a bridge or a railway and a part of the green time may be lost;
4. another lane at the intersection does not have a fixed length of the green time due to an actuated control;
5. times between departures differ due to driver distraction and/or vehicle's condition and length.

In [Subsection 2.3.2](#), we consider the turning flow and prove a decomposition result. Even though situations 2, 3 and 4 differ a lot, they can be modelled in the same way by assuming that the green and red times are random, see [Subsection 2.3.3](#). In [Subsection 2.3.4](#), we suggest a way to model the distraction of the drivers. We conclude this section by presenting numerical examples in [Subsection 2.3.5](#).

2.3.1. FCTL model

Consider an isolated intersection under fixed traffic-light control, i.e., the green and red times of each lane are fixed. This system is modelled using the FCTL model, which is a basic model in traffic theory and has been extensively studied; see, for example, [Darroch \(1964\)](#), [van Leeuwen \(2006\)](#).

Due to the fact that the green and red times are fixed, the queues at different lanes are independent and can be considered separately. Thus, we focus on one lane and consider the queue length for this lane. The analysis for the other lanes is the same. Suppose that each delayed vehicle needs the same time τ to depart from the lane. Thus, if there is a long queue at the beginning of the green time, we see a departure every τ seconds. In what follows, we suppose that time is split

in time intervals, each of τ seconds. The **effective green time** consists of $g \in \mathbb{N}$ time intervals and the **effective red time** of $r \in \mathbb{N}$ time intervals. Thus, no more than g delayed vehicles can depart during the green time. We suppose that each **cycle** starts with g green time intervals and then switches to r red time intervals, where by green and red time intervals we mean time intervals during green and red time, respectively. Together this gives $c = g + r$ time intervals in a cycle.

Let $Y_{n,m}$ be the number of arrivals during the n^{th} time interval of the m^{th} cycle with pgf $Y_{n,m}(z)$. We make the following two assumptions:

Assumption 2.17 (Independent arrivals assumption). The numbers of arrivals $Y_{n,m}$, for $m = 1, 2, \dots, n = 0, \dots, c - 1$, are identically distributed and independent of each other.

Thus, we can denote $Y_{n,m}$ simply as Y and $Y_{n,m}(z)$ as $Y(z)$. This assumption is realistic for an intersection that lies far enough from other **traffic-light intersections**; for example, if the intersection is isolated. However, if the distance between intersections is short, then the vehicles arrive in **platoons** and this assumption does not hold. This case we consider in **Chapters 4 - 6**. Following **van Leeuwen (2006)**, we add the so called FCTL assumption:

Assumption 2.18 (FCTL assumption). If a vehicle arrives during the green time and finds an empty queue, then it proceeds without delay.

Assumption 2.18 means that if the queue is cleared before the end of the green time, the vehicles that arrive during the remaining green time will immediately depart. Therefore, the queue will remain empty till the end of the green time. This assumption is quite realistic for a straight-going flow since vehicles that find no queue will proceed without stopping and, thus, are able to depart at the **free-flow speed**. However, for turning flows, this assumption is not realistic, especially, for right turn (or left turn in the case of left-hand traffic), because the turning vehicles need to slow down. Thus, in the next subsection, we also consider the FCTL model with the one-vehicle assumption (see **Assumption 2.20** below) instead of the FCTL assumption.

Denote by $X_n(z)$ the pgf of the queue length in the stationary state at the beginning of the n^{th} time interval during an arbitrary cycle. The pgf exists in the case of a stable system, i.e., when the possible number of departures is greater than the expected number of arrivals:

$$cY'(1) < g.$$

Let q_n be the stationary probability of having an empty queue at the beginning of the n^{th} time interval during a cycle, i.e., $q_n = X_n(0)$. Then, under **Assump-**

tions 2.17 and 2.18, we get:

$$\begin{aligned} X_{n+1}(z) &= \frac{X_n(z) - q_n}{z} Y(z) + q_n, & \text{for } n = 0, \dots, g-1, \\ X_{n+1}(z) &= X_n(z)Y(z), & \text{for } n = g, \dots, c-2, \\ X_0(z) &= X_{c-1}(z)Y(z). \end{aligned} \quad (2.21)$$

The first equation in (2.21) is based on the fact that if there is at least one vehicle in the queue at the beginning of the green time interval, then one vehicle departs and Y vehicles arrive. In terms of pgfs, this means multiplying by $Y(z)/z$. However, if the queue is empty, it remains empty throughout the time interval. Thus, q_n should not be multiplied by any additional function. During the red time, Y vehicles arrive each time interval and none depart. Therefore, we only multiply by $Y(z)$ in the final two equations of (2.21).

This gives us, after some rearrangement, the pgf of the *overflow queue* length, defined as the queue length at the beginning of the red time:

$$X_g(z) = \frac{\sum_{k=0}^{g-1} q_k z^k (Y(z))^{g-1-k}}{z^g - (Y(z))^c} (z - Y(z)). \quad (2.22)$$

The expression has form (2.1), where $D(z) = z^g - A(z)$, $A(z) = (Y(z))^c$, $B(z) = Y(z)$, $f(z) = z - Y(z)$ and $x_k = q_k$. Note that by Corollary 2.3, $f(z) = 0$ inside and on the unit circle only for $z = 1$. Thus, $f(\bar{z}_k) \neq 0$ for each root $\bar{z}_k$ of (2.2), $k = 1, \dots, g-1$. We need to assume that $Y(z)$ is an analytic function in the disk $D_{1+\epsilon}$ for some $\epsilon > 0$. Then Assumption 2.1 holds. Since we consider only stable systems, we assume that $A'(1) = cY'(1) < g$, which means that $B'(1) = Y'(1) < 1$, and, therefore, Assumption 2.13 holds.

One can prove, see, e.g., Darroch (1964), that the expected overflow queue length $\mathbb{E}X_g = X'_g(1)$ and the expected queue length $\mathbb{E}L_{\text{FCTL}}$ at an arbitrary point during the cycle are connected in the following way:

$$\begin{aligned} \mathbb{E}L_{\text{FCTL}} &= \frac{r}{c(1 - Y'(1))} X'_g(1) + \frac{r^2 Y'(1)}{2c(1 - Y'(1))} + \\ &\quad + \frac{r(Y''(1) + Y'(1) - (Y'(1))^2)}{2c(1 - Y'(1))^2}. \end{aligned} \quad (2.23)$$

Now consider $X'_g(1)$. From (2.11), we get the following theorem.

Theorem 2.19. *For the FCTL model, the overflow queue length is given by*

$$X'_g(1) = \frac{(Y'(1) - 1)}{2\pi} I + g, \quad (2.24)$$

where

$$I = \int_{-\pi}^{\pi} \frac{gz^g - zc(Y(z))^{c-1}Y'(z)}{z^g - (Y(z))^c} \frac{z}{z - Y(z)} d\varphi,$$

$z = z(\varphi) = (1 + \varepsilon)e^{i\varphi}$, and ε is chosen as described in Theorem 2.6.

Plugging (2.24) in (2.23) gives after rearrangement the following formula:

$$\mathbb{E}L_{\text{FCTL}} = -\frac{r}{2\pi c}I + \frac{gr}{c} + \frac{r(c+g+1)Y'(1)}{2c(1-Y'(1))} + \frac{rY''(1)}{2c(1-Y'(1))^2}. \quad (2.25)$$

2.3.2. FCTL model for the turning flow

For some cases, it may turn out that the FCTL assumption does not hold. For example, in the case of a right turn at an intersection (or a left turn in the case of left-hand traffic), vehicles that find the queue empty need to decelerate almost to the speed of a delayed vehicle. Thus, rather than the FCTL assumption it is logical to make the following assumption.

Assumption 2.20 (One-vehicle assumption). If a number of vehicles arrives during the green time interval and finds the queue empty, then only one of the vehicles proceeds without delay, and the others form a queue.

This assumption means that even if the queue was cleared during the previous green time intervals, the overflow queue may be non-empty. The stability condition in this model coincides with the stability condition of the FCTL model under the FCTL assumption. Equations (2.21) will change to:

$$\begin{aligned} X_{n+1}(z) &= X_n(z) \frac{Y(z)}{z} + q_n \left(1 - \frac{1}{z}\right) Y(0), & \text{for } n = 0, \dots, g-1, \\ X_{n+1}(z) &= X_n(z) Y(z), & \text{for } n = g, \dots, c-2, \\ X_0(z) &= X_{c-1}(z) Y(z). \end{aligned} \quad (2.26)$$

Indeed, if there are no vehicles in the queue during the green time, then $[Y-1]^+$ vehicles will queue, with pgf $(Y(z)-Y(0))/z + Y(0)$. Therefore, we obtain for $n < g$ that

$$X_{n+1}(z) = (X_n(z) - q_n) \frac{Y(z)}{z} + q_n \left(\frac{Y(z) - Y(0)}{z} + Y(0) \right).$$

This gives us the above result after a small rearrangement. Hence, the pgf of the overflow queue length is expressed as:

$$X_g(z) = \frac{\sum_{k=0}^{g-1} q_k z^k (Y(z))^{g-1-k}}{z^g - (Y(z))^c} (z-1) Y(0). \quad (2.27)$$

As in the case of the FCTL assumption, we can use our method for the FCTL model with the one-vehicle assumption if the system is stable, i.e., $cY'(1) < g$, and the function $Y(z)$ is analytic in some disk $D_{1+\epsilon}$.

Note that in case of Bernoulli arrivals, Assumptions 2.18 and 2.20 coincide. In this case, $Y(z) = (1 - Y(0))z + Y(0)$, and, thus, $z - Y(z) = Y(0)(z - 1)$. In general, coefficients q_k may differ under these two assumptions even though the equation for roots is the same. This happens because one of the roots is 1 and

plugging it in the numerator gives 0 without resulting in an additional equation to find q_k . The required g^{th} equation comes from the fact that $X_g(1) = 1$. This normalization equation has different forms for [Assumptions 2.18](#) and [2.20](#). Under [Assumption 2.18](#), we get that:

$$\sum_{k=0}^{g-1} q_k(1 - Y'(1)) = g - cY'(1). \quad (2.28)$$

However, under [Assumption 2.20](#), the normalization equation is of the following form:

$$\sum_{k=0}^{g-1} q_k Y(0) = g - cY'(1). \quad (2.29)$$

We see that these models coincide only if $1 - Y(0) = Y'(1)$, i.e., this happens only in the case of Bernoulli arrivals.

From equations (2.28) and (2.29) and the fact that the roots are the same in both models, it follows that

$$q_{k,\text{FCTL}}(1 - Y'(1)) = q_{k,1v}Y(0),$$

where $q_{k,\text{FCTL}}$ (resp., $q_{k,1v}$) is the stationary probability of having an empty queue at the beginning of the k^{th} time interval during a cycle in case of [Assumption 2.18](#) (resp., [2.20](#)). Moreover, by comparing equations (2.22), (2.27) and (2.21), (2.26), we get that in the case of a stable system, for each $k = 0, \dots, c - 1$:

$$X_{k,1v}(z) = X_{k,\text{FCTL}}(z)X_{\text{dif}}(z),$$

where

$$X_{\text{dif}}(z) = \frac{(1 - Y'(1))(z - 1)}{z - Y(z)},$$

$X_{k,\text{FCTL}}(z)$ and $X_{k,1v}(z)$ are pgfs at the beginning of the k^{th} time interval for the FCTL model and the one-vehicle model, respectively. Note that $X_{\text{dif}}(z)$ is independent of g and c with $X_{\text{dif}}(1) = 1$ and $X'_{\text{dif}}(1) = \frac{Y''(1)}{2(1-Y'(1))}$.

Consider the case $g = c = 1$. In the case of the FCTL assumption, the queue is always empty once it becomes empty since there is no red time, during which vehicles can form a queue. Therefore, in case of stability, i.e., $Y'(1) < 1$, we get $X_{0,\text{FCTL}}(z) = 1$. Thus, we get $X_{0,1v} = X_{\text{dif}}(z)$, i.e., $X_{\text{dif}}(z)$ is the pgf of the queue with $g = c = 1$ and arrivals $Y(z)$. Hence, we get the following decomposition result (compare with the decomposition results in [polling systems](#), [Boxma and Groenendijk \(1987\)](#)):

Theorem 2.21. *For arbitrary $g \leq c$, in the case of a stable system, the queue with the one-vehicle assumption can be considered as a sum of two independent queues: the FCTL queue with the same g, c and the one-vehicle queue with $g = c = 1$.*

Note that $X_{\text{dif}}(z)$ can be also viewed as the pgf of the queue length at a bottleneck, e.g., when a part of the road has low speed limit and all the vehicles decelerate before this place.

We can as well change the model to allow more than one vehicle, but not all, to pass the junction if the queue is empty. Then the changes will be only in the function $f(z)$, and we can easily use our method if [Assumptions 2.1](#) and [2.13](#) hold. For the same reason, there would be a similar decomposition result. The expected queue length in this new case will be less than the expected queue length in the case of the one-vehicle assumption but more than the expected queue length in the case of the FCTL assumption.

Remark 2.22 (The relation to the [bulk-service queue](#)). The bulk-service model can be seen as an intermediate step between the considered traffic-light models. More precisely, for $A_b(z) = Y(z)^c$, it can be shown using a sample path argument that the average queue length satisfies the following inequalities:

$$X'_{g,\text{FCTL}}(1) \leq (X_b^-)'(1) \leq X'_{g,1v}(1). \quad (2.30)$$

In traffic terms, the bulk service model can be seen as a queue with service at the end of the green time that serves g (or less) vehicles and then waits one cycle for new arrivals.

From [\(2.30\)](#), the difference between the two FCTL models, described by [Theorem 2.21](#), gives an upper bound on the difference between the FCTL and bulk-service models. In [Subsection 2.3.5.2](#), we show that this difference is usually small.

2.3.3. Random green and red times

Suppose we know that with probability $\theta_{r,g}$ the red time consists of r time intervals and the following green time consists of g time intervals, $r, g \in \mathbb{N} \cup \{0\}$, $r + g \neq 0$. Thus, cycles start with a red time, followed by a green time. The lengths of the red time and the green time are independent of the lengths of red and green times before this cycle. We assume that the length of the green time is at most N time intervals, i.e., $\sum_{k=0}^{\infty} \sum_{l=N+1}^{\infty} \theta_{k,l} = 0$. For simplicity, we consider the smallest such N , so $\sum_{k=0}^{\infty} \theta_{k,N} > 0$. As before, in each time interval, Y vehicles arrive with pgf $Y(z)$. We also use the FCTL assumption. One can easily alter the formulas in case of the one-vehicle assumption.

Let $X(z)$ be the pgf of the overflow queue length and $X_k^{r,g}(z)$ be the (conditional) pgf of the queue length at the beginning of the k^{th} green time interval given that there are r red time intervals and g green time intervals in this cycle. Note that when the choices of r and g are known, the pgf of the queue length changes as in [\(2.21\)](#). Thus,

$$\begin{aligned} X_0^{r,g}(z) &= X(z)Y(z)^r, \\ X_{k+1}^{r,g}(z) &= X_k^{r,g}(z) \frac{Y(z)}{z} + p_k^{r,g} \left(1 - \frac{Y(z)}{z} \right), \end{aligned}$$

where $p_k^{r,g}$ is the probability of having an empty queue at the beginning of the k^{th} green time interval in case of a cycle with r red and g green time intervals. Hence, with probability $\theta_{r,g}$ the pgf of the queue-length at the beginning of the next red time will be

$$X_g^{r,g}(z) = X(z) \frac{(Y(z))^{g+r}}{z^g} + \left(1 - \frac{Y(z)}{z}\right) \sum_{k=0}^{g-1} p_k^{r,g} \left(\frac{Y(z)}{z}\right)^{g-1-k}.$$

Therefore, the pgf of the overflow queue length satisfies the following equation:

$$X(z) = \sum_{r,g} \theta_{r,g} \left(X(z) \frac{(Y(z))^{g+r}}{z^g} + \left(1 - \frac{Y(z)}{z}\right) \sum_{k=0}^{g-1} p_k^{r,g} \left(\frac{Y(z)}{z}\right)^{g-1-k} \right).$$

Isolating the function $X(z)$ gives us

$$X(z) = \frac{\sum_{r,g} \theta_{r,g} \left(1 - \frac{Y(z)}{z}\right) \sum_{k=0}^{g-1} p_k^{r,g} \left(\frac{Y(z)}{z}\right)^{g-1-k}}{1 - \sum_{r,g} \theta_{r,g} \frac{(Y(z))^{g+r}}{z^g}}.$$

After multiplying both numerator and denominator by z^N and making some rearrangements, we get

$$X(z) = \frac{\sum_{k=0}^{N-1} q_k (Y(z))^k z^{N-1-k}}{z^N - A(z)} (z - Y(z)),$$

where $q_k = \sum_{r,g} \theta_{r,g} p_{g-1-k}^{r,g}$ and $A(z) = \sum_{r,g} \theta_{r,g} (Y(z))^{g+r} z^{N-g}$. As we see, we can apply our method if $A(z)$ and $Y(z)$ are analytic in some disk $D_{1+\epsilon}$, $A'(1) < N$ and $Y'(1) < 1$.

For applications, $A(z)$ is always a finite sum, so it is analytic in some $D_{1+\epsilon}$ if $Y(z)$ is analytic in $D_{1+\epsilon}$. Note that

$$A'(1) = \sum_{r,g} \theta_{r,g} ((g+r)Y'(1) + (N-g)) = \mathbb{E}(G+R)Y'(1) + N - \mathbb{E}G,$$

where G and R are random variables that denote the length of the green and red times during a cycle. Thus, if $A'(1) < N$, we get $\mathbb{E}(G+R)Y'(1) < \mathbb{E}G$. It immediately follows that $Y'(1) < 1$. We see once again that the stability assumption holds in the case of a stable system.

Now we give three examples of possible applications of this model.

2.3.3.1. Interruption by pedestrians and/or cyclists

Let us focus on one lane and, for simplicity, on one source causing interruptions. Suppose that with probability p there is an arrival of a cyclist during a cycle. Suppose that we need t_c green time intervals for switching on and off the cyclists green time. The green time for pedestrians and cyclists can be provided using the green time g of this lane or can be added as extra time to the cycle time $c = g + r$. In the first case, we have $\theta_{r,g} = 1 - p$, $\theta_{r+t_c, g-t_c} = p$, and in the second case, we have $\theta_{r,g} = 1 - p$, $\theta_{r+t_c, g} = p$.

2.3.3.2. Interruption by trains and boats

This case is similar to the previous one. The only difference is in the length of interruption. We assume that interruption happens during a cycle with probability p . We also assume that if it happens, then all green time is effectively red. Therefore, $\theta_{r,g} = 1 - p$, $\theta_{c,0} = p$.

2.3.3.3. Vehicle-actuated control on another lane

Suppose that another lane has a vehicle-actuated control, and, thus, the length of its green time is not deterministic. In this case, to find an approximation for the queue length on the considered lane we assume that the lengths of red times during different cycles are independent of each other. Then, if we know the distribution of the green time for the other lane, we get a distribution of the red time for the considered lane. Let $\theta_{r,g} = p_r$, where p_r is the probability of the red time length r , and g is fixed. In the case of a fixed cycle, we take $\theta_{r,c-r} = p_r$.

Remark 2.23 (Actuated control). The analysis of the actuated-controlled lane is complicated due to the fact that the arrival and service processes are connected. We return to this problem in [Chapter 4](#).

Remark 2.24 (Random green and red times). It is interesting to note that if the length of the green time depends on the length of the following red time but is independent of the lengths of the preceding green and red times, the resulting pgf of the queue length at the beginning of a green time does not necessarily have form (2.1). For examples, consider a case when with probability p the cycle consists of 3 green and 1 red time interval and with probability $1 - p$ the cycle consists of 1 green and 3 red time intervals. Then, the pgf of the queue length at the beginning of a cycle can be represented as follows

$$X(z) = \frac{q_0(p + (1 - p)z^2)(Y(z))^2 + q_1zY(z) + q_2z^2}{z^3 - (Y(z))^4(p + (1 - p)z^2)}(z - Y(z))Y(z),$$

where q_k is the probability of having an empty queue at the beginning of the k^{th} green time interval, see (2.21). As we see, the functions in the numerator do not form a geometric sequence, so the results of this chapter are not applicable. In this case, one needs to use the results for a general pgf, see [Chapter 3](#).

2.3.4. Uncertain departure time

In this subsection, we consider the following extension of the FCTL model. Suppose that during a green time interval, the driver of the departing vehicle may be distracted with a fixed probability $p \in [0, 1]$. Therefore, during the green time, even in case of a non-empty queue, vehicles do not depart each time interval. Then, each driver has a geometrically distributed number of tries to depart from the queue. This can be modelled using the FCTL model in the following way.

Type of arrivals	Pgf $Y(z)$	Variance
Bernoulli	$\lambda z + (1 - \lambda)$	$\lambda - \lambda^2$
Binomial	$\left(\frac{\lambda}{n}z + 1 - \frac{\lambda}{n}\right)^n$	$\lambda - \frac{\lambda^2}{n}$
Poisson	$e^{\lambda(z-1)}$	λ
Negative binomial	$\left(\frac{n}{n+\lambda-\lambda z}\right)^n$	$\lambda + \frac{\lambda^2}{n}$

Table 2.2: The considered types of the arrival process.

Suppose that during the green time there is an extra arrival with probability p . In this way, we compensate for the uncertainty in the departures. The distribution of the queue length in this model will be the same as the distribution in our new model. So we consider the case that the number of the arrivals during the green time interval has pgf $Y(z)(pz + (1 - p))$. However, during the red time we have still Y arrivals per time interval with pgf $Y(z)$. Hence, the pgf of the overflow queue length is given by the following formula:

$$X_g(z) = \frac{\sum_{k=0}^{g-1} q_k z^k (Y(z)(pz + 1 - p))^{g-1-k}}{z^g - (Y(z))^c (pz + 1 - p)^g} (z - Y(z)(pz + 1 - p)).$$

Using the same argumentation as in [Subsection 2.3.1](#), one can see that the pgf is of the general form, described in [Section 2.1](#). Thus, it is possible to use the same techniques in case when $Y(z)$ is analytic in some $D_{1+\epsilon}$, and the system is stable, i.e., $cY'(1) + gp < g$.

2.3.5. Numerical results

In this subsection, we present the numerical results for FCTL-type models. First, we consider the impact of the variability of the arrival process for the FCTL queue in [Subsection 2.3.5.1](#). Second, we discuss the difference between the FCTL assumption and the one-vehicle assumption in [Subsection 2.3.5.2](#). Then, we study the impact of the traffic interruptions caused by cyclists and uncertain departure times in [Subsections 2.3.5.3](#) and [2.3.5.4](#). Finally, we consider the green-time allocation problem for an isolated [traffic-light intersection](#) in [Subsection 2.3.5.5](#). Further numerical results can be found in the technical report [Oblakova et al. \(2016\)](#).

2.3.5.1. Variability of the arrival process

In this subsection, we show the impact of the variability of the arrival process on the queue length for the FCTL model. In what follows, λ denotes the arrival rate per time interval, i.e., $\lambda = Y'(1)$. In [Table 2.2](#), we summarise pgfs and variances of the considered arrival processes.

Figure 2.4: The expected delay as a function of load for Bernoulli, binomial, Poisson and negative binomial arrivals with $c = 60$, $n = 2$ and a) $g = 5$, b) $g = 40$.

Difference	$g = 5$	$g = 15$	$g = 30$	$g = 40$
$\mathbb{E}D_{negbin} - \mathbb{E}D_{pois}$	29.1472	28.6778	28.1833	27.7916
$\mathbb{E}D_{pois} - \mathbb{E}D_{binom}$	29.1369	28.6156	28.0097	27.5466
$\mathbb{E}D_{binom} - \mathbb{E}D_{bern}$	29.1258	28.5392	27.7332	27.0498

Table 2.3: The differences (in seconds) in the expected delay for different types of arrival process, $c = 60$ and load $\rho = 0.9833$.

Note that Bernoulli arrivals have the least possible variance for a fixed λ . Indeed, the variance of an arbitrary arrival process with rate λ is equal to $Y''(1) + Y'(1) - (Y'(1))^2 \geq \lambda - \lambda^2$. In what follows, the parameter n of both the binomial and the negative binomial distribution is set to be equal to 2.

First, we focus on the expected delay per vehicle for various types of arrivals. The expected delay is computed using Little's law $\mathbb{E}D = \mathbb{E}L/\lambda$, see [Little \(1961\)](#), where $\mathbb{E}L = \mathbb{E}L_{\text{FCTL}}$ is computed using (2.25). In [Figure 2.4](#), the expected delay is plotted as a function of the load $\rho = c\lambda/g$ for $c = 60$ and $g = 5$ or 40 . The expected delay is given in seconds rather than time intervals, and one time interval is set to be equal to 2 seconds.

Our first observation is that the higher the variability of the arrival process the longer the expected delay. However, for each green time g the difference is significant only for a load higher than 0.8. As we can see for high loads, the relative difference between expected delays for various types of arrivals is increasing as the green time increases. However, the absolute difference is decreasing. For load $\rho = 0.9833$ (the highest load in the figures) and $g = 5, 15, 30, 40$, the absolute differences are given in [Table 2.3](#). In the table, we use $\mathbb{E}D_{bern}$, $\mathbb{E}D_{binom}$, $\mathbb{E}D_{pois}$,

Figure 2.5: The probabilities q_k of a queue being empty, $k = 0, \dots, g-1$, for $g = 10$, $c = 20$, $n = 2$ and a) $\lambda = 0.2$, b) $\lambda = 0.45$.

ED_{negbin} to express the expected delay in the cases of Bernoulli, binomial, Poisson and negative binomial arrivals, respectively.

From the table, we see that the absolute difference is decreasing as the green time increases. In many approximations (see [Miller \(1963\)](#) and [van den Broek et al. \(2006\)](#)), the dependence on the variance is set to be linear for the fixed arrival rate, green times and cycle times. However, we see that this difference increases for higher variance.

Next we consider the probabilities q_k of a queue being empty for $k = 0, \dots, g-1$. In [Figure 2.5](#), for $g = 10$, $c = 20$ and $\lambda = 0.2, 0.45$ these probabilities are plotted as functions of k . For fixed g , c and λ , the sum of probabilities is the same for various types of arrival, but the distributions of this sum are different. As we see, for all arrival rates, at the beginning of the green time the probability of a queue being empty is lower for the arrival processes that have lower variability, whereas at the end of green time the situation is reversed. The graphs look very different depending on the arrival rates. For low arrival rates, the graphs are concave and the difference between graphs for different arrival types is small. For high arrival rates, the graphs are convex and the relative difference between arrival types is larger.

2.3.5.2. Comparing the FCTL and one-vehicle assumptions

Let us first consider $X'_{\text{dif}}(1)/\lambda$, i.e., the expected difference in the delay between FCTL and one-vehicle models. We plot it as a function of the arrival rate $\lambda = Y'(1)$ in [Figure 2.6](#).

Note that if at an intersection there are at least two conflicting “main” phases, then both of them use less than half of the cycle time. Thus, in the case of a stable system, the arrival rate is lower than 0.5. If we consider, for example,

Figure 2.6: The expected extra delay due to the one-vehicle assumption as compared to the FCTL assumption.

Figure 2.7: The relative difference (in %) in the expected delay between the FCTL and one-vehicle models as a function of load for binomial, Poisson and negative binomial arrivals for $c = 60$, $n = 2$ and a) $g = 5$, b) $g = 40$.

Poisson arrivals, the expected extra delay is $\frac{1}{\lambda} \cdot \frac{\lambda^2}{2(1-\lambda)} \leq \frac{1}{2(1-\lambda)} \leq 1$ time interval, i.e., not more than 2 seconds. Therefore, for most of the cases the extra delay will be very short. Similarly, in these cases, the extra queue length is less than one vehicle.

As we have shown, the absolute difference between FCTL and one-vehicle mod-

els is quite small. However, it is interesting to consider the relative difference. For the same settings as in Figure 2.4, we plot the relative difference in the expected delay in Figure 2.7. For a shorter green time, the delay observed in the FCTL model is already very long and the expected difference $X'_{\text{dif}}(1)/\lambda$ is small. Thus, the relative difference is very small. For a longer green time, the delay is shorter and the arrival rate is higher, so the relative difference is higher. As we see, for $g = 40$, this relative difference reaches 10% for negative binomial arrivals with $n = 2$.

2.3.5.3. Disruption of the traffic

In this subsection, we consider disruption of traffic due to cyclists (or pedestrians). Suppose that cyclists need 5 time intervals, i.e., 10 seconds, to cross the road. There are two ways to provide the required green time. We can either shorten the green time of one lane (thus making room for cyclists to cross), or, alternatively, extend the total cycle time and, hence, add extra red time to all lanes. Let p be the probability of cyclists arrival during a cycle. In Figure 2.8, we plot the **overflow queue** length as a function of the rate for different p and g and fixed $c = 60$. Each graph we plot only up to load $\rho = 0.975$. The arrival process is assumed to be Poisson. As we see, the first way to deal with cyclists is highly disadvantageous for the affected lane. It significantly decreases the **capacity** of the lane and increases the overflow queue length. The effect is stronger for a shorter green time. The conclusion is that it is better, if possible, to increase the cycle time.

2.3.5.4. Uncertain departure times

Consider the FCTL model with uncertain departure times that was presented in Subsection 2.3.4. In Figure 2.9, we plot the expected overflow queue length as a function of the arrival rate for different probabilities of departure. We suppose that, on average, a vehicle departs in 2 seconds. The length of a time interval is set to be equal to $\tau = 2p$ seconds, where p is the probability of departure. The arrival process is assumed to be Poisson. We fix the cycle time (in seconds) and consider two different green times. Depending on $p = 1, 5/6, 5/7$, there are on average 5 departures in 5, 6 or 7 time intervals.

The uncertainty in departure times does not influence the capacity of the system but only increases the overflow queue length and, consequently, the delay. The overflow queue length is a bit longer for a green time of 10 seconds (about 2 vehicles for the highest considered load 0.9833). The effect of the uncertainty becomes noticeable for loads over 0.8. Note that the average overflow queue length is almost 0 for loads up to 0.7-0.8 (depending on the green time), which means that most of the vehicles are served during one cycle time after the arrival.

Figure 2.8: The overflow queue length in case of cyclists as a function of arrival rate. The cycle time is $c = 60$. The green time in figures a) and b) is $g = 15$ and on c) and d) $g = 30$. In figures a) and c), the green time for cyclists (5 time intervals) is taken from the green time of the lane. In figures b) and d), the extra time is added to the cycle.

2.3.5.5. Green-time allocation problem

In this section, we consider the green-time allocation problem. We compare three allocation policies. The first one is due to Webster, see [Webster \(1958\)](#). Webster proposed to allocate the green time proportionally to the arrival rate, i.e., such that the load is the same for each lane. As we see in [Figure 2.4](#), the lane with a short green time faces longer delays than the lane with a long green time for the same load. To deal with such unfairness, we consider the second allocation policy which is based on the minimization of the maximum expected delay per lane. In this case, we expect that all the vehicles will encounter a similar average delay (independently of the lane). Note that even when the delay for the lane with the

Figure 2.9: The overflow queue length for FCTL model with uncertain departure times depending on the probability p of a departure. The cycle time is 120 seconds, the green times are a) 10 and b) 60 seconds.

lowest arrival rate is long, it is experienced by a small fraction of vehicles. Therefore, this lane has a small contribution to the average delay of a vehicle arriving at the intersection. For this reason, our third allocation policy is to minimise the expected total queue length or, alternatively, to minimise the expected delay of an arbitrary vehicle.

We consider the following example. Suppose we have three lanes with arrival rates λ_1 , $\lambda_2 = 3\lambda_1$ and $\lambda_3 = 6\lambda_1$. Suppose also that we assign to them, in total, 50 green time intervals out of 60 time intervals in a cycle. Due to the computational efficiency of our method we can minimise the expected queue length or delay using a simple exhaustive search.

Let ρ be the total load, i.e., $\rho = \sum_i \lambda_i 60/50 = 12\lambda_1$. Consider the case $\rho = 0.9$. The results for Bernoulli and Poisson cases are given in Table 2.4. In the table, the total delay is a sum of average delays per lane. Even though it has no physical meaning, it measures the change in the expected delays. Note that for both types of arrivals all three ways of the green time allocation give similar results. However, the difference in the delay between different policies is significant, especially for the first lane. So, as we see, the proportional green time is the most beneficial for the busiest lane but very unfair for the lane with the lowest rate. Using either minimal total delay or minimal delay per lane policy improves significantly (2 times) the situation for the lane with the lowest rate but increases the delay for the busiest lane.

For lower loads these three ways of the green time allocation work in completely different way. For the same parameters, we consider our allocation policies for total loads between 0.0166 and 0.9833. The arrival process is assumed to be Poisson. The results can be found in Figure 2.10, where we plot the average delay per lane

	Bernoulli arrivals				Poisson arrivals			
	Lane 1	Lane 2	Lane 3	Total	Lane 1	Lane 2	Lane 3	Total
Arrival rate λ	0.075	0.225	0.450	0.750	0.075	0.225	0.450	0.750
Proportional green time								
Green time	5	15	30		5	15	30	
Delay	139.63	61.73	31.75	233.11	147.91	68.99	37.91	254.81
Queue length	5.24	6.95	7.14	19.33	5.55	7.76	8.53	21.84
Minimal total queue length								
Green time	6	15	29		6	15	29	
Delay	68.88	61.73	38.10	168.71	71.10	68.99	48.67	188.76
Queue length	2.58	6.95	8.57	18.10	2.67	7.76	10.95	21.38
Minimal delay per lane								
Green time	7	15	28		6	15	29	
Delay	56.27	61.73	55.36	173.35	71.10	68.99	48.67	188.76
Queue length	2.11	6.95	12.46	21.51	2.67	7.76	10.95	21.38

Table 2.4: The comparison of the expected delay and the queue length for various green time allocation policies and total load $\rho = 0.9$. The delay is given in seconds and the green time in time intervals. A time interval is set to be equal to 2 seconds.

and the average total queue length for each allocation policy as functions of the total load and the allocated green time for each lane.

As we see, the minimum delay per lane policy suggests almost equal green time allocation for low total load, while the minimum total queue length policy suggests to give the largest part (more than proportional 30 time intervals) of the green time to the third lane with the highest arrival rate. The difference in the average delay for different policies is more significant for the lanes with low and high arrival rates than for the lane with a medium arrival rate. We also see that the total queue length for proportional policy and minimum total length are almost the same. However, we already saw from Table 2.4 how significant is the difference in the average delay (especially for the first lane).

For lanes 1 and 2 (low and medium rate), the minimum delay per lane policy is the most beneficial, while for the third lane it is the most disadvantageous. Conversely, the minimum total length policy is the most beneficial for the busiest lane and the most disadvantageous for the first two lanes. However, for a higher load it results in a shorter delay for the lane with the lowest arrival rate than the proportional policy.

Figure 2.10: The effect of various green-time allocation policies on a), b), c) the average delay per lane, d) the average total queue length and e), f) the green time allocated to each lane. Lanes 1, 2, 3 correspond to the lanes with low, medium and high arrival rate.

2.4. Conclusions

We have presented a new method to calculate the expectation and distribution of the queue length for several discrete-time queueing systems. The solution is given in terms of contour integrals without any implicitly-defined variables. We successfully applied this method to the [bulk-service queue](#) and several traffic-light models providing insights in traffic behaviour. For the FCTL model with one-vehicle assumption, we proved a decomposition result, which allowed us to give a bound on the difference between the FCTL and bulk-service queue-length expectations, which turns out to be small for many relevant applications.

We have shown that our method outperforms the classic [root-finding](#) and [matrix-analytic](#) methods in terms of speed and reliability. Using our method, the queue-length expectation can be found more than 7 and 17 times faster compared to the root-finding and matrix-analytic methods, respectively. Moreover, we found that the root-finding approach can fail to give a realistic result. This happens more often for the case of many roots, where up to 10% of the considered cases result in no answer and 40% in the negative or non-real (complex) expected queue length. The speed of the matrix-analytic approach depends on the size of the block matrices, which coincides with the number of roots, and can be up to 54 times slower than our approach. In the next [Chapter 3](#), we extend our integral method to a wider class of systems.

2.A. Proof of [Lemma 2.15](#)

Recall that we need to prove that $zB(w) \neq wB(z)$ for $z \neq w$, $z, w \in \bar{D}_1$ if $B(z)$ is a pgf with $B'(1) < 1$. Consider Taylor expansion $\sum_{j=0}^{\infty} b_j z^j$ of $B(z)$ around zero. Let $\tilde{B}(z) = \sum_{j=2}^{\infty} b_j z^j$. Then, it is equivalent to prove that $z(b_0 + \tilde{B}(w)) \neq w(b_0 + \tilde{B}(z))$ for $z \neq w$, $z, w \in \bar{D}_1$. Now, we can reformulate the lemma in the following way: there is not more than one root of the equation $z = a(b_0 + \tilde{B}(z))$ in $\bar{D}_1$ for each $a \in \mathbb{C}$. Here a represents the value $w/(b_0 + \tilde{B}(w))$ for some $w \in \bar{D}_1$. To make such a reformulation we need to show that $b_0 + \tilde{B}(w) \neq 0$ for all $w \in \bar{D}_1$.

First, we show that $b_0 > \tilde{B}(1)$. Consider the derivative of $\tilde{B}(z)$ at 1. On the one hand, according to the stability assumption, we get that $\tilde{B}'(1) = B'(1) - b_1 < 1 - b_1 = \tilde{B}(1) + b_0$. On the other hand, $b_k \geq 0$ for $k \in \mathbb{N}$, and, thus,

$$\tilde{B}'(1) = \sum_{j=2}^{\infty} j b_j \geq \sum_{j=2}^{\infty} 2 b_j = 2\tilde{B}(1).$$

Finally, we get that $2\tilde{B}(1) \leq \tilde{B}'(1) < \tilde{B}(1) + b_0$. Therefore, $b_0 > \tilde{B}(1)$.

Now, we need to use the following small lemma:

Lemma 2.25. *Consider an analytic function $C(z) = \sum_{j=0}^{\infty} c_j z^j$ in $D_{r+\epsilon}$, where $r \in \mathbb{R}$, $\epsilon > 0$. Suppose that $c_j \geq 0$ for all $j \in \mathbb{N} \cup \{0\}$, then the absolute value of $C(z)$ reaches the maximum value in r , i.e., $|C(z)| \leq C(r)$ for each $z \in \bar{D}_r$.*

Proof. Since the function $C(z)$ is analytic in disk $D_{r+\epsilon}$, Taylor series converges absolutely. Therefore, $|C(z)| \leq \sum_{j=0}^{\infty} c_j |z|^j \leq \sum_{j=0}^{\infty} c_j r^j = C(r)$. $\square$

As a corollary we get that $|\tilde{B}(z)| < b_0$ for each $z \in \bar{D}_1$, and, consequently, there are no solutions of the equation $b_0 + \tilde{B}(z) = 0$ in $\bar{D}_1$. Hence, it is sufficient to prove that the equation

$$z = a(b_0 + \tilde{B}(z)) \quad (2.31)$$

has not more than one root in $\bar{D}_1$ for each $a \in \mathbb{C}$.

If $a = 0$, we have the simple equation $z = 0$ that has not more than one solution. If $a \neq 0$, we can uniquely represent it as $te^{i\varphi}$, where $0 \leq \varphi < 2\pi$, $t > 0$. We want to prove that the number of roots inside the unit disk does not increase when t increases and φ is fixed. To do so, we consider our roots as functions of t and prove that the absolute value of a root increases as t increases. Suppose $z(t)$ is a root of equation (2.31) inside the unit disk. Consider its derivative:

$$\frac{dz}{dt} = \frac{d}{dt}(te^{i\varphi}(b_0 + \tilde{B}(z))) = e^{i\varphi}(b_0 + \tilde{B}(z)) + te^{i\varphi}\tilde{B}'(z)\frac{dz}{dt}.$$

Rearranging and plugging $z/(t(b_0 + \tilde{B}(z)))$ instead of $e^{i\varphi}$ give us

$$t \frac{dz}{dt} = \frac{z}{1 - \frac{z\tilde{B}'(z)}{b_0 + \tilde{B}(z)}}.$$

Thus, the derivative of the $|z(t)|^2$ is equal to

$$\frac{d(z\bar{z})}{dt} = \frac{dz}{dt}\bar{z} + \frac{d\bar{z}}{dt}z = 2\operatorname{Re}\left(\frac{dz}{dt}\bar{z}\right) = \frac{2|z|^2 \operatorname{Re}\left(1 - \frac{z\tilde{B}'(z)}{b_0 + \tilde{B}(z)}\right)}{t \left|1 - \frac{z\tilde{B}'(z)}{b_0 + \tilde{B}(z)}\right|^2},$$

where $\bar{z}$ is the complex conjugate of z . To prove that this derivative is positive we only need to prove the following lemma.

Lemma 2.26. *There is some $\epsilon > 0$ such that for each $z \in D_{1+\epsilon}$*

$$\operatorname{Re}\left(1 - \frac{z\tilde{B}'(z)}{b_0 + \tilde{B}(z)}\right) > 0. \quad (2.32)$$

Proof. First of all, it is sufficient to prove that (2.32) holds for each $z \in \bar{D}_1$. Indeed, if for some point z inequality (2.32) holds, then for some neighbourhood of z it also holds. Since $\bar{D}_1$ is compact, if (2.32) holds in $\bar{D}_1$, it also holds for a small neighbourhood $D_{1+\epsilon}$, $\epsilon > 0$.

Now note that inequality (2.32) is equivalent to the following inequality:

$$\operatorname{Re}\left((b_0 + \tilde{B}(z))(b_0 + \tilde{B}(\bar{z})) - z\tilde{B}'(z)(b_0 + \tilde{B}(\bar{z}))\right) > 0.$$

It can be rewritten as

$$b_0^2 > \operatorname{Re} \left(b_0(z\tilde{B}'(z) - \tilde{B}(z) - \tilde{B}(\bar{z})) + (z\tilde{B}'(z) - \tilde{B}(z))\tilde{B}(\bar{z}) \right).$$

Since $\operatorname{Re}(\tilde{B}(\bar{z})) = \operatorname{Re}(\tilde{B}(z))$, we finally get that (2.32) is equivalent to

$$b_0^2 > \operatorname{Re} \left(b_0(z\tilde{B}'(z) - 2\tilde{B}(z)) + (z\tilde{B}'(z) - \tilde{B}(z))\tilde{B}(\bar{z}) \right). \quad (2.33)$$

Note that $\operatorname{Re}(z) \leq |z|$ for each $z \in \mathbb{C}$, and, thus,

$$\begin{aligned} \operatorname{Re} \left(b_0(z\tilde{B}'(z) - 2\tilde{B}(z)) + (z\tilde{B}'(z) - \tilde{B}(z))\tilde{B}(\bar{z}) \right) \\ \leq b_0|z\tilde{B}'(z) - 2\tilde{B}(z)| + |z\tilde{B}'(z) - \tilde{B}(z)| \cdot |\tilde{B}(z)|. \end{aligned}$$

Since the functions $z\tilde{B}'(z) - 2\tilde{B}(z)$, $z\tilde{B}'(z) - \tilde{B}(z)$ and $\tilde{B}(z)$ are analytic and have non-negative coefficients in their Taylor expansion at 0, we can use Lemma 2.25. This gives us that

$$\begin{aligned} \operatorname{Re} \left(b_0(z\tilde{B}'(z) - 2\tilde{B}(z)) + (z\tilde{B}'(z) - \tilde{B}(z))\tilde{B}(\bar{z}) \right) \\ \leq b_0(\tilde{B}'(1) - 2\tilde{B}(1)) + (\tilde{B}'(1) - \tilde{B}(1)) \cdot (\tilde{B}(1)) < b_0(b_0 - \tilde{B}(1)) + b_0\tilde{B}(1) = b_0^2. \end{aligned}$$

Here we used that $\tilde{B}'(1) < \tilde{B}(1) + b_0$. Thus, we proved for each $z \in \bar{D}_1$ inequality (2.33), which is equivalent to (2.32). $\square$

We proved that $\frac{d|z\bar{z}|}{dt} > 0$ for each $z = z(t) \in D_{1+\epsilon} \setminus \{0\}$. Since $b_0 + \tilde{B}(0) \neq 0$, we get that $z(t) \neq 0$ for every t . Therefore, any root of equation (2.31) for $t > 0$ goes outside $\bar{D}_{1+\epsilon}$. Hence, the number of roots inside D_1 decreases when t increases. But for small $t > 0$ we can use Rouché's theorem to show that there is only one root of equation (2.31). Thus, we have proved Lemma 2.15. $\square$

2.B. Computational remarks

In this appendix, we give several computational remarks.

2.B.1. Zero-free case

From Assumption 2.2, it follows that the function $B(z)$ has at most one zero in D_1 . It is possible that the function $B(z)$ has no zeroes in D_1 . For example, if $B(z)$ is the pgf of a Poisson random variable. We will show here how to simplify the computation in this case. Then, we will give a simplified version of Algorithm 2.12 and an easy test how to check if $B(z)$ is zero-free in D_1 if $B(z)$ is a pgf.

Consider $\tilde{y}_k = \bar{z}_k/B(\bar{z}_k)$. In the same way as before, we can get formulas for $\tilde{\eta}_k = \sum_{j=1}^{g-1} \tilde{y}_j^k$ that do not contain any integrals along S_δ . More specifically, we consider

$$\tilde{h}(y) = \sum_{k=0}^{g-1} x_k y^k.$$

Thus, $\tilde{h}\left(\frac{z}{B(z)}\right)(B(z))^{g-1} = h\left(\frac{B(z)}{z}\right)z^{g-1}$. We conclude that $\tilde{y}_k$ are zeroes of $\tilde{h}(y)$, and, thus,

$$\tilde{h}(y) = x_{g-1} \prod_{k=1}^{g-1} (y - \tilde{y}_k).$$

Using the same argument as in [Subsection 2.1.1](#), we get

$$\frac{x_{g-1-k}}{x_{g-1}} = (-1)^k \sigma_k(\tilde{y}_1, \dots, \tilde{y}_{g-1}) = (-1)^k \tilde{\sigma}_k. \quad (2.34)$$

Note that, if $B(z)$ does not have zeroes in $\bar{D}_1$, then

$$\tilde{\eta}_k = \frac{1}{2\pi i} \oint_{S_{1+\varepsilon}} \frac{D'(z)}{D(z)} \left(\frac{z}{B(z)}\right)^k dz - 1.$$

Hence, we do not need to compute two integrals as in (2.6). As before, we need to subtract the residue at 1, which is equal to 1. This leads to the following simplified algorithm to find unknown coefficients x_k :

Algorithm 2.27 (Computation of x_k , $k = 0, \dots, g-1$ in the case of zero-free $B(z)$).

1. Find $\varepsilon > 0$ such that there are only g roots of equation (2.2) in the disk $\bar{D}_{1+\varepsilon}$, using one of two ways described in [Subsection 2.B.3](#).
2. Compute $\tilde{\eta}_k$, using

$$\tilde{\eta}_k = -1 + \frac{1}{2\pi} \int_{-\pi}^{\pi} z \frac{D'(z)}{D(z)} \left(\frac{z}{B(z)}\right)^k d\varphi,$$

where for convenience $z = z(\varphi) = (1 + \varepsilon)e^{i\varphi}$.

3. Compute $\tilde{\sigma}_k$ from $\tilde{\eta}_k$ using (2.7).
4. Compute x_k , $k = 0, \dots, g-1$, using (2.34) and (2.5).

The necessary and sufficient condition for a pgf $B(z)$ to be zero-free in D_1 is given by the following lemma.

Lemma 2.28. *The pgf $B(z)$ is zero-free in D_1 if and only if $B(-1) > 0$.*

Proof. Suppose that $B(-1) > 0$ and there is a zero t of $B(z)$ in D_1 . Since $B(z)$ has at most one zero and $B(\bar{t}) = \overline{B(t)}$, this zero is on the real line. Consider some positive $\Delta < \min\{B(-1), 1 - B'(1)\}$ and the function $C(z) = \frac{B(z) - \Delta}{1 - \Delta}$. Since $B(0) \geq 1 - B'(1) > \Delta$, the function $C(z)$ is a pgf and $C'(1) = \frac{B'(1)}{1 - \Delta} < 1$. Therefore, according to [Lemma 2.15](#), $C(z)$ has not more than one zero in the unit disk. However, $C(t) < 0 < C(-1), C(1)$. Thus, by the intermediate value theorem it has at least two zeroes. We got a contradiction.

The rest of the proof follows from the fact that if $B(-1) \leq 0 < B(1)$, then by intermediate value theorem there is such $t \in [-1, 1]$ that $B(t) = 0$. $\square$

2.B.2. Choice of δ

In (2.6), we use the parameter δ such that there are no roots of equation (2.2) in $\bar{D}_\delta = \{z: |z| \leq \delta\}$. For a general function $D(z)$, we use the following method to find an appropriate δ . Consider the integral

$$\frac{1}{2\pi i} \oint_{S_\delta} \frac{D'(z)}{D(z)} dz = \frac{1}{2\pi} \int_{-\pi}^{\pi} z \frac{D'(z)}{D(z)} d\varphi,$$

where z in the last integral stands for $z(\varphi) = \delta e^{i\varphi}$. By Lemma 2.5 and the residue theorem, the value of this integral is equal to the number of roots of equation (2.2) inside the disk $\bar{D}_\delta$. Therefore, if this integral is equal to zero, then δ is correct. Hence, we suggest to start with $\delta = 1/2$. Then, check the value of the integral (2.2). If it is not zero (in this case it should be at least 1), then set δ to $1/4$ and check the integral again. Continuing like this, we will find an appropriate δ after a finite number of steps since $D(z)$ is an analytic function with $D(0) \neq 0$.

In case $D(z) = z^g - A(z)$, we prove that any $\delta < \delta_0 = A(0)/(2 - A(0))$ satisfies our condition. Consider $A(z) = \sum_{j=0}^{\infty} a_j z^j$. Since $a_j \geq 0$ and $\sum_{j=0}^{\infty} a_j = 1$,

$$|A(z) - A(0)| = \left| \sum_{j=1}^{\infty} a_j z^j \right| \leq \sum_{j=1}^{\infty} a_j |z|^j \leq (1 - A(0))|z|$$

for any z in the unit disk. Therefore, for $z \in \bar{D}_\delta$, where $\delta < \delta_0$, we get

$$|A(z)| \geq A(0) - |A(z) - A(0)| \geq A(0) - (1 - A(0))|z| > |z| \geq |z|^g.$$

Thus, there are no roots of equation (2.2) in $\bar{D}_\delta$ for $\delta < \delta_0$.

Note that if $g > 1$ and $A(0) \neq 1$, $A(0)/(2 - A(0)) < 1$, and, thus, for each $z \in \bar{D}_{\delta_0}$ we have $|z| > |z|^g$. Therefore, $|A(z)| > |z|^g$ even if $A(0) - (1 - A(0))|z| = |z|$, which is equivalent to $|z| = \delta_0$. Hence, in almost all cases $\delta = \delta_0$ is an appropriate value. Note that $\delta_0 > 0$ since $A(0) > 0$. However, δ_0 can be so small that computational errors will arise. In this case, we can use the general way to find δ .

2.B.3. Choice of ε

The choice of ε may be done in the same way as the choice of δ by computing contour integrals. For $D(z) = z^g - A(z)$, one can prove using Rouché's theorem that there are no roots of equation (2.2) in $\bar{D}_{1+\varepsilon} \setminus \bar{D}_1$ for $1 + \varepsilon < \bar{z}_{-1}$, where $\bar{z}_{-1}$ is the (only) root of equation (2.2) on the open ray $(1, \infty)$. Thus, if a lower bound on $\bar{z}_{-1}$ is known or it is easy to compute numerically, one can use any point on $(0, \bar{z}_{-1} - 1)$ as ε . However, the smaller the ε the greater the computational error for the integral, so we suggest to take $\varepsilon = \operatorname{argmax}_{z \in [1, \bar{z}_{-1}]} |z^g - A(z)| - 1$ or, in the case when the errors are unlikely to arise, just $\varepsilon = (\bar{z}_{-1} - 1)/2$. In this way, we stay far enough from the roots of equation (2.2).

Chapter 3

Roots, symmetry and contour integrals in queueing systems

‘It’s useful to go out of this world and see it from the perspective of another one.’

Terry Pratchett

In [Chapter 2](#), we considered a special class of rational [pgfs](#). This type of pgf appears in several discrete-time queueing systems such as the [bulk-service queue](#) and the [FCTL queue](#). The numerator of such a pgf consists of a linear combination of functions that form a geometric sequence. In this chapter, we extend the results of [Chapter 2](#) for a pgf with general functions in the numerator:

$$X(z) = \frac{\sum_{j=0}^n x_j f_j(z)}{D(z)}(z-1), \quad (3.1)$$

where $f_j(z)$ for $j = 0, \dots, n$ and $D(z)$ are known analytic functions, $f_j(1) \neq 0$ for some j , $D(z)$ has $n+1$ zeroes inside the closed unit disk denoted as $\bar{z}_0 = 1, \bar{z}_1, \dots, \bar{z}_n$, and x_j are the unknown coefficients. The case of [Chapter 2](#) is given by the following functions $f_j(z)$:

$$f_j(z) = z^j (B(z))^{n-j} f(z) / (z-1), \quad (3.2)$$

where the functions $B(z)$ and $f(z)$ are known, and $f(1) = 0$. These functions form a geometric sequence with [common ratio](#) $z/B(z)$.

When relaxing the assumption of the geometric form (3.2), we cannot apply the results of [Chapter 2](#). Therefore, we find an alternative solution. As before, we can apply the classical [root-finding](#) approach to find the unknowns in the numerator using the zeroes of the denominator $\bar{z}_0 = 1, \bar{z}_1, \dots, \bar{z}_n$ in the unit disk. Due to the analyticity of the pgf, the zeroes of the denominator are also zeroes of the

numerator. This yields n linear equations for the unknowns:

$$\sum_{j=0}^n x_j f_j(\bar{z}_i) = 0, \quad i = 1, \dots, n. \quad (3.3)$$

One more equation follows from the normalisation equation $X(1) = 1$:

$$\sum_{j=0}^n x_j f_j(1) = D'(1). \quad (3.4)$$

The system of equation (3.3) and (3.4) can be rewritten in matrix form:

$$M(1, \bar{z}_1, \dots, \bar{z}_n)(x_0, \dots, x_n)^T = (D'(1), 0, \dots, 0)^T, \quad (3.5)$$

where

$$M(z, z_1, \dots, z_n) = \begin{pmatrix} f_0(z) & f_1(z) & \dots & f_n(z) \\ f_0(z_1) & f_1(z_1) & \dots & f_n(z_1) \\ \vdots & \vdots & \ddots & \vdots \\ f_0(z_n) & f_1(z_n) & \dots & f_n(z_n) \end{pmatrix}. \quad (3.6)$$

Here and later, we use $\bar{z}_1, \dots, \bar{z}_n$ for the zeroes of the denominator and $z_1, \dots, z_n$ for complex variables. In this chapter, we use the properties both of the matrix $M(z, z_1, \dots, z_n)$ and of symmetric polynomials to find the pgf without computing the roots. We represent the pgf using a determinant of a certain matrix, where each entry is a symmetric function of the roots, which can be computed using contour integrals.

Note that the pgf of the type considered in [Chapter 2](#) has two properties. First, the pgf can be represented in such product form that each term of the product depends on not more than one root. Second, the mean of the corresponding random variable, e.g., the queue length, is an additive function of the roots, which allows us to find the mean using one contour integral. In this chapter, we give a necessary and sufficient condition in terms of the functions $f_j(z)$ for these properties to hold. The special product form of the pgf exists if and only if the functions $f_j(z)$ form, after a linear transformation, a geometric sequence, where the common ratio of the sequence may be more general than in [Chapter 2](#). For the additive-mean property, we distinguish two cases. In the first case, which we call *degenerate* case, the mean is independent of the roots and, therefore, is always an additive function of the roots. We give a simple necessary and sufficient condition for the degenerate case in terms of values $f_j(1)$ and $f'_j(1)$ for $j = 0, \dots, n$. For the non-degenerate case, the additive-mean property is equivalent to the existence of the special product form of the pgf. The systems with these properties include the bulk-service queue, see [Bailey \(1954\)](#), the multi-server $M/D/s$ and $Geo/D/s$ queues, which are, in some sense, equivalent to the bulk-service queue with Poisson and Binomial arrivals, see [Janssen and van Leeuwen \(2008\)](#) and [Kim and Chaudhry \(2006\)](#), and the

FCTL queue, see [Section 2.3](#). The [traffic-light](#) queues described in [Part III](#) are examples of queueing systems with pgf of form [\(3.1\)](#) without these properties. For such queues, we can use the results for the systems with a general matrix $M(z, z_1, \dots, z_n)$, see [Section 3.2](#).

This chapter is structured as follows. In [Section 3.1](#), we give the definitions and the properties of symmetric polynomials and symmetric functions that we use later. In [Section 3.2](#), we obtain the pgf in terms of symmetric functions for a general numerator. Then, we analyse a special sub-class of pgfs in [Section 3.3](#). Finally, we conclude the chapter in [Section 3.4](#). The proofs of the intermediate results are given in [Appendix 3.A](#) with the exception of the short proofs that are included in the text.

3.1. Preliminaries

In this section, we give the required definitions and the preliminary results. First, in [Subsection 3.1.1](#), we define symmetric, skew-symmetric and additive functions and alternant matrices. Then, we relate linearly dependent functions and singular matrices, see [Subsection 3.1.2](#). In [Subsection 3.1.3](#), we describe two types of symmetric polynomials and their properties. Afterwards, in [Subsection 3.1.4](#), we analyse the determinant of an alternant matrix, which will be used later in [Section 3.2](#) to obtain the pgf as a symmetric function of the roots. Finally, in [Subsection 3.1.5](#), we obtain the values of symmetric functions at certain points in terms of contour integrals.

3.1.1. Definitions

Consider a function $f(z_1, \dots, z_n)$ of n complex variables. We focus on two types of functions: symmetric and skew-symmetric.

Definition 3.1. A function $f(z_1, \dots, z_n)$ is called *symmetric* if

$$f(z_1, \dots, z_n) = f(z_{s(1)}, \dots, z_{s(n)})$$

for any permutation $s \in \mathbf{S}_n$, where $\mathbf{S}_n$ is the set of all permutations of the set $\{1, \dots, n\}$.

Definition 3.2. A function $f(z_1, \dots, z_n)$ is called *skew-symmetric* if

$$f(z_1, \dots, z_n) = \text{sgn}(s)f(z_{s(1)}, \dots, z_{s(n)})$$

for any permutation $s \in \mathbf{S}_n$. Here, $\text{sgn}(s)$ is the sign of the permutation s and is equal to $(-1)^{m_s}$, where m_s denotes the number of [transpositions](#), i.e., permutations that interchange two elements, needed to construct s . The sign is independent of the representation of s as a product of transpositions.

In the analysis in the following section, we also use a subtype of symmetric functions, namely additive functions.

Definition 3.3. A function $f(z_1, \dots, z_n)$ is called *additive* if

$$f(z_1, \dots, z_n) = \sum_{k=1}^n g(z_k),$$

for some function $g(z)$.

In the analysis in [Sections 3.2](#) and [3.3](#), we mainly work with alternant matrices.

Definition 3.4. Consider the functions $f_1(z), \dots, f_n(z)$ and points $z_1, \dots, z_n$. The matrix

$$\Lambda(z_1, \dots, z_n) = \begin{pmatrix} f_1(z_1) & \dots & f_n(z_1) \\ \vdots & \ddots & \vdots \\ f_1(z_n) & \dots & f_n(z_n) \end{pmatrix}$$

is called an *alternant* matrix.

An example of an alternant matrix is a [Vandermonde matrix](#), where $f_k(z) = z^{k-1}$. The determinant of the Vandermonde matrix is denoted by

$$V(z_1, \dots, z_n) = \prod_{1 \leq i < j \leq n} (z_j - z_i).$$

Note that the determinant of an alternant matrix is a skew-symmetric function of $z_1, \dots, z_n$. This follows immediately from the fact that if one interchanges two rows (or columns) in a square matrix, such an operation changes the sign of the determinant, see [Vinberg \(2003\)](#).

Remark 3.5 (Equality of rational functions). In what follows, we will work with rational functions of several variables, i.e., $f(z_1, \dots, z_n)/g(z_1, \dots, z_n)$. Suppose that both the numerator and the denominator are analytic functions and $g(z_1, \dots, z_n)$ is not identically equal to 0. For the case of one variable, i.e., $n = 1$, there are not more than a finite number of points where this rational function is not defined, namely where $g(z_1) = 0$. However, for $n > 1$ this is not true. For example, the function $1/(z_1 + z_2)$ is not defined on the plane $z_1 = -z_2$. Suppose that the functions $f(z_1, \dots, z_n)$ and $g(z_1, \dots, z_n)$ are defined on the set $D_1^n = \{(z_1, \dots, z_n) \in \mathbb{C}^n : |z_i| < 1, i = 1, \dots, n\}$. Then, one can prove that the function $f(z_1, \dots, z_n)/g(z_1, \dots, z_n)$ is defined on a dense open subset of D_1^n , because the set of the zeroes of the function $g(z_1, \dots, z_n)$ is closed and has an empty interior, see [Gunning and Rossi \(2009\)](#). In what follows, when we say that two rational functions are equal, we mean that they are equal on an open dense subset of D_1^n , where both of them are defined.

3.1.2. Singular matrices and linear independence

In this subsection, we relate linear dependency between functions and singular matrices, see the following properties. For the proofs of [Properties 3.6](#) and [3.7](#), we refer to [Bôcher \(1900\)](#) and [Appendix 3.A.1](#), respectively.

Property 3.6. *The analytic functions $f_1(z), \dots, f_n(z)$ are linearly dependent if and only if the [Wronskian](#)*

$$\det W(z) = \det \begin{pmatrix} f_1(z) & \dots & f_n(z) \\ \vdots & \ddots & \vdots \\ f_1^{(n-1)}(z) & \dots & f_n^{(n-1)}(z) \end{pmatrix} \quad (3.7)$$

is identically equal to 0.

Property 3.7. *Consider numbers $a_i \in \mathbb{C}$, $i = 0, \dots, n$. Suppose the matrix*

$$\Lambda_a = \begin{pmatrix} a_0 & \dots & a_n \\ f_0(z_1) & \dots & f_n(z_1) \\ \vdots & \ddots & \vdots \\ f_0(z_n) & \dots & f_n(z_n) \end{pmatrix}$$

is singular for all $z_1, \dots, z_n$. If $a_i \neq 0$ for some i , then the functions $f_0(z), \dots, f_n(z)$ are linearly dependent.

From [Property 3.7](#), it follows that if the matrix Λ_a is singular for all $z_1, \dots, z_n$ and the functions $f_0(z), \dots, f_n(z)$ are linearly independent, then $a_i = 0$ for $i = 0, \dots, n$.

3.1.3. Symmetric polynomials and their properties

In this subsection, we define two types of symmetric polynomials and introduce their properties. Recall that the [elementary symmetric polynomials](#) are given by

$$\sigma_m = \sigma_m(z_1, \dots, z_n) = \sum_{1 \leq i_1 < \dots < i_m \leq n} z_{i_1} \cdots z_{i_m}.$$

The above formula is used for $m = 1, \dots, n$. For convenience, $\sigma_0 = 1$, and $\sigma_m = 0$ if either $m > n$ or $m < 0$. As mentioned in [Chapter 2](#), elementary symmetric polynomials naturally arise in Vieta's formulas that relate the coefficients of a polynomial with its roots. Consider a polynomial $\sum_{j=0}^n a_j z^j$ with the roots $\tilde{z}_1, \dots, \tilde{z}_n$, then it can be written as

$$a_n \prod_{i=1}^n (z - \tilde{z}_i) = a_n \sum_{j=0}^n (-1)^j \sigma_j(\tilde{z}_1, \dots, \tilde{z}_n) z^{n-j}. \quad (3.8)$$

The proof of (3.8) requires the expansion of the left-hand side of (3.8), see [Vinberg \(2003\)](#).

In the analysis below, we mainly use the *complete homogeneous symmetric polynomials* defined as

$$\zeta_m = \zeta_m(z_1, \dots, z_n) = \sum_{1 \leq i_1 \leq \dots \leq i_m \leq n} z_{i_1} \cdots z_{i_m}.$$

Note the difference in the definitions of the elementary and complete homogeneous symmetric polynomials: the indices i_k and i_{k+1} for the latter case may coincide for some or all k . This allows us to use the above formula for $m > n$. For $m = 0$, we define $\zeta_0 = 1$, and, for $m < 0$, $\zeta_m = 0$.

The following property can be used to recursively find all complete homogeneous polynomials from the elementary symmetric polynomials. The proof is given in [Appendix 3.A.2](#).

Property 3.8. *For $m > 0$, the following equality holds*

$$\zeta_m = \sum_{j=1}^n (-1)^{j-1} \sigma_j \zeta_{m-j}. \quad (3.9)$$

The upper limit of summation in (3.9) can be changed to any number that is at least $\min\{n, m\}$ since $\zeta_{m-j} = 0$ for $j > m$ and $\sigma_j = 0$ for $j > n$.

When we are interested in the values of the complete homogeneous and elementary symmetric polynomials at point $(\bar{z}_1, \dots, \bar{z}_n)$, where $\bar{z}_1, \dots, \bar{z}_n$ are the roots of a certain equation, these values can be found without actually knowing the roots, see [Subsection 3.1.5](#). In the following [Subsection 3.1.4](#), for an analytic function of one variable, we construct a *symmetric* function of n variables using the complete homogeneous symmetric polynomials. Such functions on D_1^n are later used to rewrite the determinant of an alternant matrix as a product of a skew-symmetric Vandermonde determinant and a symmetric function.

3.1.4. Symmetric functions and alternant matrices

In this subsection, we represent the determinant of an alternant matrix in terms of the [Vandermonde determinant](#) and symmetric functions. We use such determinants in [Section 3.2](#) to give an alternative representation of the considered type of pgf. First, we define a transformation rule of an analytic function in the unit disk $D_1 = \{z \in \mathbb{C} : |z| < 1\}$ to a symmetric function defined in $D_1^m \subset \mathbb{C}^m$. Then, we give several properties of this transformation. The main result of this subsection is presented in [Lemma 3.11](#).

Consider an analytic function $f(z)$, with the Taylor expansion at 0 given by $f(z) = \sum_{i=0}^{\infty} \alpha_i z^i$. Let

$$F_k^m = F_k^m(z_1, \dots, z_m) = \sum_{i=k}^{\infty} \alpha_i \zeta_{i-k}(z_1, \dots, z_m),$$

where m corresponds to the number of variables. We call F_k^m the (k, m) -*transformation* of the function $f(z)$. The (k, m) -transformation of the function $f_j(z)$ is denoted by $F_{j,k}^m$. Note that the function $F_k^m(z_1, \dots, z_m)$ is a symmetric function of $z_1, \dots, z_m$.

In the analysis in [Section 3.3](#), we will use the following properties.

Property 3.9. *Consider $m \leq n$. Then,*

$$F_k^n(z_1, \dots, z_m, 0, \dots, 0) = F_k^m(z_1, \dots, z_m).$$

Property 3.10. *For $k \geq 0$, consider the function $f(z) = \sum_{i=0}^{\infty} \alpha_i z^i$ and its (l, n) -transformations F_l^n , for $l = k, \dots, k+n$. Then,*

$$F_k^n + \sum_{j=1}^n (-1)^j \sigma_j F_{k+j}^n = \alpha_k.$$

[Property 3.9](#) follows from the definition of complete homogeneous symmetric polynomials and only requires the observation that $\zeta_j(z_1, \dots, z_m, 0, \dots, 0) = \zeta_j(z_1, \dots, z_m)$. [Property 3.10](#) follows from [Property 3.8](#) and equality $\zeta_0 = 1$. In the following lemma, we show that the determinant of an alternant matrix can be written as the product of a Vandermonde determinant and a matrix composed from (k, n) -transformations of the functions $f_1(z), \dots, f_n(z)$ that are used in the alternant matrix. The proof is given in [Appendix 3.A.3](#).

Lemma 3.11. *Suppose that the functions $f_1(z), \dots, f_n(z)$ are analytic in a neighbourhood of 0. Then,*

$$\det \begin{pmatrix} f_1(z_1) & f_2(z_1) & \dots & f_n(z_1) \\ f_1(z_2) & f_2(z_2) & \dots & f_n(z_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(z_n) & f_2(z_n) & \dots & f_n(z_n) \end{pmatrix} = V(z_1, \dots, z_n) \det \Phi(z_1, \dots, z_n), \quad (3.10)$$

where $V(z_1, \dots, z_n)$ is the *Vandermonde determinant* and

$$\Phi(z_1, \dots, z_n) = \begin{pmatrix} F_{1,0}^n & F_{2,0}^n & \dots & F_{n,0}^n \\ F_{1,1}^n & F_{2,1}^n & \dots & F_{n,1}^n \\ \vdots & \vdots & \ddots & \vdots \\ F_{1,n-1}^n & F_{2,n-1}^n & \dots & F_{n,n-1}^n \end{pmatrix}. \quad (3.11)$$

Remark 3.12 (Symmetry and singularity). [Lemma 3.11](#) is an important result for our analysis in [Sections 3.2](#) and [3.3](#). Note that the matrix $\Phi(z_1, \dots, z_n)$ in [\(3.11\)](#) consists of symmetric functions of $z_1, \dots, z_n$. Therefore, it is, in general, non-singular for $z_i = z_j$, which will allow us in [Section 3.3](#) to give proofs using induction in the number of variables, n . Moreover, $\det \Phi(z_1, \dots, z_n)$ is not identically

equal to 0 even when $z_1 = \dots = z_n$ provided that the functions $f_1(z), \dots, f_n(z)$ are analytic and linearly independent. The reason for this is that the function $\det \Phi(z, \dots, z)$ is up to a constant equal to the [Wronskian \(3.7\)](#), which is not identically equal to 0, see [Property 3.6](#). To relate $\det \Phi(z, \dots, z)$ with the Wronskian, it is sufficient to note that

$$F_{j,k}^n(z, \dots, z) = \sum_{l=k}^{n-1} \frac{(n-1-k)!}{l!(l-k)!(n-1-l)!} z^{l-k} f_j^{(l)}(z)$$

for $k = 0, \dots, n-1$. The last equation follows from

$$\begin{aligned} \zeta_m(\underbrace{z, \dots, z}_n) &= \binom{n+m-1}{m} z^m = \frac{1}{(n-1)!} \frac{d^{n-1}}{dz^{n-1}} z^{m+n-1} \\ &= \frac{1}{(n-1)!} \frac{d^{n-1}}{dz^{n-1}} (z^{n-1-k} z^{m+k}) \\ &= \frac{1}{(n-1)!} \sum_{l=k}^{n-1} \binom{n-1}{l} \frac{d^{n-1-l}}{dz^{n-1-l}} z^{n-1-k} \frac{d^l}{dz^l} z^{m+k} \\ &= \sum_{l=k}^{n-1} \frac{(n-1-k)!}{l!(l-k)!(n-1-l)!} z^{l-k} \frac{d^l}{dz^l} z^{m+k}, \end{aligned}$$

for $k = 0, \dots, n-1$.

3.1.5. Roots and contour integrals

In this subsection, we provide a way of computing the values of symmetric polynomials at special points. Consider the analytic function $D(z)$. Suppose $1, \bar{z}_1, \dots, \bar{z}_n$ are the only roots of the equation

$$D(z) = 0 \tag{3.12}$$

in the closed unit disk $\bar{D}_1 = \{z \in \mathbb{C}: |z| \leq 1\}$. Then, it is possible to compute $\zeta_k(\bar{z}_1, \dots, \bar{z}_n)$ without finding the roots. For this we use the approach of [Chapter 2](#). The first step is to represent the complete homogeneous symmetric polynomials in terms of the elementary symmetric polynomials, see [Property 3.8](#). Then, we recursively use Newton's formula, see [Mead \(1992\)](#),

$$k\sigma_k = \sum_{j=1}^k (-1)^{j-1} \sigma_{k-j} \eta_j, \quad k = 1, \dots, n, \tag{3.13}$$

to find the elementary symmetric polynomials in terms of the [power-sum symmetric polynomials](#) $\eta_j = \eta_j(z_1, \dots, z_n) = \sum_{l=1}^n z_l^j$, $j = 1, \dots, n$. The power sums, in turn, are found using Cauchy's residue theorem, see [Knopp \(1996\)](#). Namely,

$$\eta_j(\bar{z}_1, \dots, \bar{z}_n) + 1 = \frac{1}{2\pi i} \oint_{S_{1+\varepsilon}} \frac{D'(z)}{D(z)} z^j dz, \tag{3.14}$$

where $\varepsilon > 0$ is defined so that there are no roots of equation (3.12) with $1 < |z| < 1 + \varepsilon$, and $S_{1+\varepsilon} = \{z \in \mathbb{C}: |z| = 1 + \varepsilon\}$, for more details see Chapter 2.

Remark 3.13 (Zero at 1). If equation (3.12) does not have a zero at 1, one needs to change the left-hand side of (3.14) to just $\eta_j(\bar{z}_1, \dots, \bar{z}_n)$. We explicitly consider the case that 1 is a root since the denominators of the pgfs analysed in Section 3.2 have a zero at 1.

Remark 3.14 (The functions F_k^n and contour integrals). If the function $f(z)$ is a polynomial, then $F_k^n(\bar{z}_1, \dots, \bar{z}_n)$ is a finite sum of the complete homogeneous symmetric polynomials and can be found using equations (3.14), (3.13) and (3.9). The application of the Cauchy residue theorem, see (3.14), is a crucial step for going from the root-finding approach to the contour-integral approach. If the function $f(z)$ is not a polynomial, one can truncate the infinite summation in $F_k^n(\bar{z}_1, \dots, \bar{z}_n)$ using the following bound:

$$\left| F_k^n(\bar{z}_1, \dots, \bar{z}_n) - \sum_{k=0}^M \alpha_k \zeta_k(\bar{z}_1, \dots, \bar{z}_n) \right| \leq C \binom{M+n}{n-1} \frac{(qr)^{M+1}}{(1-qr)^n}, \quad (3.15)$$

where $|\bar{z}_k| \leq q$ for $k = 1, \dots, n$, $|\alpha_l| \leq Cr^l$ for $l = 0, 1, \dots$, $f(z) = \sum_{k=0}^{\infty} \alpha_k z^k$; and $qr < 1$, see the proof in Appendix 3.A.4. Therefore, the determinant of the matrix $\Phi(z_1, \dots, z_n)$, see (3.11), can be found without computing the roots.

Remark 3.15 (Functions F_k^n and minimal polynomials). For the sake of completeness, we also give a different way of finding F_k^n at $(\bar{z}_1, \dots, \bar{z}_n)$ without truncation (3.15). Let $Q(z) = \prod_{i=1}^n (z - \bar{z}_i)$ be the minimal polynomial with the roots $\bar{z}_1, \dots, \bar{z}_n$. Due to (3.8), $Q(z) = \sum_{j=0}^n (-1)^j \sigma_j(\bar{z}_1, \dots, \bar{z}_n) z^{n-j}$, which means that this polynomial can be found without computing the roots, see equations (3.14) and (3.13). Then, see Bekker et al. (2016),

$$\zeta_m(\bar{z}_1, \dots, \bar{z}_n) = \sum_{l=1}^n \frac{\bar{z}_l^{n+m-1}}{Q'(\bar{z}_l)}.$$

By applying the Cauchy residue theorem, we obtain

$$\zeta_m(\bar{z}_1, \dots, \bar{z}_n) = \frac{1}{2\pi i} \oint_{S_1} \frac{z^{n+m-1}}{Q(z)} dz,$$

where S_1 is the unit circle, i.e., $S_1 = \{z \in \mathbb{C}: |z| = 1\}$. Hence, we find

$$F_k^n(\bar{z}_1, \dots, \bar{z}_n) = \frac{1}{2\pi i} \oint_{S_1} \frac{f(z) - \sum_{l=0}^{k-1} \alpha_l z^l}{Q(z)} z^{n-k-1} dz.$$

Using this formula for each entry of the matrix (3.11), i.e., n^2 times, can be computationally demanding. Still, it can be useful in some special cases.

Remark 3.16 (Additive functions and contour integrals). Suppose that the function $f(z_1, \dots, z_n)$ is an additive function, i.e., $f(z_1, \dots, z_n) = \sum_{k=1}^n g(z_k)$. If the function $g(z)$ is analytic in $D_{1+\epsilon} \setminus \{1\}$, then the value of $f(\bar{z}_1, \dots, \bar{z}_n)$ is given by one contour integral:

$$f(\bar{z}_1, \dots, \bar{z}_n) = \frac{1}{2\pi i} \oint_{S_{1+\varepsilon}} \frac{D'(z)}{D(z)} g(z) dz - r_1,$$

where $\varepsilon < \epsilon$ is defined as in (3.14), and r_1 is the residue of the function $D'(z)g(z)/D(z)$ at 1. In this way, we found the expectation $X'(1)$ in Chapter 2 and the power sums η_k , see (3.14).

3.2. Pgf as a symmetric function of roots

In this section, we consider pgfs of the form (3.1). Recall that such a pgf has a rational form:

$$X(z) = \frac{\sum_{j=0}^n x_j f_j(z)}{D(z)} (z-1),$$

where coefficients x_j can be found from equation (3.5), $f_j(z)$ are analytic functions such that $f_j(1) \neq 0$ at least for some j , and $D(z)$ is an analytic function with $n+1$ zeroes inside the unit disk including 1, which we denote by $\bar{z}_0 = 1, \bar{z}_1, \dots, \bar{z}_n$. The goal of this section is to represent the pgf $X(z)$ as a symmetric function of the roots, see Theorem 3.21 below. Such a representation together with Remark 3.14 allows us to find the pgf $X(z)$ without finding the roots.

Remark 3.17 (Alternative representation). In the representation of the pgf, the term $(z-1)$ is usually included in the functions $f_j(z)$. One can also rewrite (3.1) as

$$X(z) = \frac{\sum_{j=0}^n x_j f_j(z)}{\tilde{D}(z)},$$

where $\tilde{D}(z) = D(z)/(z-1)$ is a function with n zeroes, $\bar{z}_1, \dots, \bar{z}_n$, inside the unit disk. Since $z=1$ is a zero of $D(z)$, if the function $D(z)$ is analytic in some disk $D_r = \{z: |z| < r\}$ with $r > 1$, then so is $\tilde{D}(z)$.

As we showed in the introduction, the unknowns in the numerator can be found using an alternant matrix

$$M(z, z_1, \dots, z_n) = \begin{pmatrix} f_0(z) & f_1(z) & \dots & f_n(z) \\ f_0(z_1) & f_1(z_1) & \dots & f_n(z_1) \\ \vdots & \vdots & \ddots & \vdots \\ f_0(z_n) & f_1(z_n) & \dots & f_n(z_n) \end{pmatrix}.$$

In the following lemma, we represent the numerator of the pgf in terms of the matrix $M(z, z_1, \dots, z_n)$. As a by-product of this representation, we obtain the

following interesting result: it is not necessary to know x_j , $j = 0, \dots, n$, to evaluate the numerator of (3.1). The solution (3.17) is given only as it is explicitly obtained in the proof.

Lemma 3.18. *Suppose the matrix $M(1, \bar{z}_1, \dots, \bar{z}_n)$ is non-singular, and $(x_0, \dots, x_n)$ is a solution of (3.5), then*

$$\sum_{j=0}^n x_j f_j(z) = D'(1) \frac{\det M(z, \bar{z}_1, \dots, \bar{z}_n)}{\det M(1, \bar{z}_1, \dots, \bar{z}_n)}, \quad (3.16)$$

and

$$x_j = (-1)^j D'(1) \frac{\det M_j(\bar{z}_1, \dots, \bar{z}_n)}{\det M(1, \bar{z}_1, \dots, \bar{z}_n)}, \quad (3.17)$$

where $M_j(z_1, \dots, z_n)$ is the matrix $M(z, z_1, \dots, z_n)$ without the first row and the $(j+1)^{\text{st}}$ column.

Proof. Since the matrix $M(1, \bar{z}_1, \dots, \bar{z}_n)$ is non-singular, (3.5) has a unique solution. Using the Laplace expansion, we readily obtain that (3.17) gives (3.16). Plugging (3.16) in the left-hand side of (3.3) and (3.4) completes the proof. $\square$

Remark 3.19 (Beyond pgf (3.1)). The proof of Lemma 3.18 does not require the precise form of the matrix $M(z, z_1, \dots, z_n)$. We used the facts that only the first row depends on z and the right-hand side is non-zero only in the first entry. Thus, instead of the matrix $M(z, z_1, \dots, z_n)$ defined in (3.6), we can apply Lemma 3.18 for other matrices. For example, instead of a row, a column of the matrix may depend on a root of a certain equation. Such systems also occur in queueing theory, see, e.g., Chaudhry et al. (2012).

Remark 3.20 (On symmetry in (3.16)). The determinant $\det M(z, z_1, \dots, z_n)$ is a skew-symmetric function of the roots. Hence, the right-hand side of (3.16), which is equal to the numerator of (3.1), is a symmetric function. However, the form (3.16) does not show how to find the value of the numerator without finding the roots $\bar{z}_1, \dots, \bar{z}_n$. Therefore, we need an equivalent representation, also see Remark 3.5.

Consider the matrix

$$\bar{M}(z, z_1, \dots, z_n) = \begin{pmatrix} f_0(z) & f_1(z) & \dots & f_n(z) \\ F_{0,0}^n & F_{1,0}^n & \dots & F_{n,0}^n \\ \vdots & \vdots & \ddots & \vdots \\ F_{0,n-1}^n & F_{1,n-1}^n & \dots & F_{n,n-1}^n \end{pmatrix}.$$

From Lemma 3.11, we find that

$$\det M(z, z_1, \dots, z_n) = V(z_1, \dots, z_n) \det \bar{M}(z, z_1, \dots, z_n),$$

where $V(z_1, \dots, z_n)$ is the *Vandermonde determinant*. In particular,

$$\frac{\det M(z, z_1, \dots, z_n)}{\det M(1, z_1, \dots, z_n)} = \frac{h(z)}{h(1)},$$

where

$$h(z) = h(z, z_1, \dots, z_n) = \det \bar{M}(z, z_1, \dots, z_n) \quad (3.18)$$

is a symmetric function of the roots. Therefore, from [Lemma 3.18](#), we obtain the following theorem.

Theorem 3.21. *Suppose the matrix $M(1, \bar{z}_1, \dots, \bar{z}_n)$ is non-singular, and the vector $(x_0, \dots, x_n)$ is a solution of (3.5), then*

$$\sum_{k=0}^n x_k f_k(z) = D'(1) \frac{h(z, \bar{z}_1, \dots, \bar{z}_n)}{h(1, \bar{z}_1, \dots, \bar{z}_n)},$$

and

$$X(z) = \frac{D'(1)h(z, \bar{z}_1, \dots, \bar{z}_n)}{D(z)h(1, \bar{z}_1, \dots, \bar{z}_n)}(z-1).$$

Using [Remark 3.14](#), we can find $h(z, \bar{z}_1, \dots, \bar{z}_n)$ without knowing the roots $\bar{z}_1, \dots, \bar{z}_n$ by computing n contour integrals. In [Section 3.3](#), we consider a special case, in which the mean, $X'(1)$, may be found using only one contour integral.

3.3. Factorisation of the pgf

In this section, we give necessary and sufficient conditions for the pgf $X(z)$ of the form (3.1) to have some special properties. The first property is the *factorisation property*, i.e., that the numerator of the pgf (3.1) can be represented as a product, where each of the terms depends on not more than one root:

$$X(z) = D'(1) \frac{g(z) \prod_{k=1}^n g(z, \bar{z}_k)}{D(z)}(z-1) \quad (3.19)$$

for some functions $g(z)$ and $g(z, w)$. This representation of the sum as a product is analogous to Vieta's formulas, see (3.8). For an example of the numerator that cannot be represented as such a product, consider three functions $f_0(z) = 1$, $f_1(z) = z$ and $f_2(z) = z^3$. Then, the numerator of (3.1) has three roots: z_1 , z_2 and $w = w(z_1, z_2)$. Therefore, the numerator is equal up to a constant to the polynomial $(z-z_1)(z-z_2)(z-w)$, which can be represented as a linear combination of the functions $f_0(z)$, $f_1(z)$ and $f_2(z)$. Hence, $w = -z_1 - z_2$, which makes the product form impossible.

The second property is the *additive-mean property*, i.e., $X'(1)$, which represents the mean of the corresponding random variable, is an *additive* function of the

roots. One can easily see that the factorisation property implies the additive-mean property. Indeed, from (3.19), L'Hospital's rule gives us

$$1 = X(1) = g(1) \prod_{k=1}^n g(1, \bar{z}_k), \quad (3.20)$$

and the mean of the corresponding random variable is given by a symmetric additive function of the roots:

$$X'(1) = -\frac{D''(1)}{2D'(1)} + \frac{g'(1)}{g(1)} + \sum_{k=1}^n \frac{\partial}{\partial z} \frac{g(z, \bar{z}_k)}{g(1, \bar{z}_k)} \Big|_{z=1}. \quad (3.21)$$

To find the above equation, we took the derivative of (3.19) and used equality (3.20). An additive form, as in (3.21), implies that under some conditions, the mean value can be found using one contour integral, see Remark 3.16. Note that for certain functions $g(z, w)$, it is possible to represent the pgf as an exponent of a contour integral, see Boon et al. (2019).

The following theorem gives the relation between factorisation and additive-mean properties in terms of the function $h(z)$, and Corollary 3.24 summarises the necessary and sufficient conditions for these properties.

Theorem 3.22. *Suppose the functions $f_k(z)$ are analytic in D_1 , the matrix $M(z, z_1, \dots, z_n)$ is defined by (3.6), and the function $h(z)$ is given by (3.18). Consider the following conditions:*

(a) *There exist a non-singular matrix*

$$A = \begin{pmatrix} a_{0,0} & \dots & a_{0,n} \\ \vdots & \ddots & \vdots \\ a_{n,0} & \dots & a_{n,n} \end{pmatrix},$$

an analytic function $B(z)$ in D_1 and a non-constant meromorphic function $C(z)$ in D_1 such that $f_j(z) = \sum_{k=0}^n a_{j,k} \tilde{f}_k(z)$, where

$$\tilde{f}_k(z) = B(z)(C(z))^k. \quad (3.22)$$

(b) *There exist meromorphic functions $g(z)$ and $g(z, w)$ such that*

$$\frac{h(z)}{h(1)} = \frac{h(z, z_1, \dots, z_n)}{h(1, z_1, \dots, z_n)} = g(z) \prod_{k=1}^n g(z, z_k). \quad (3.23)$$

(c) *There exist a constant c and a meromorphic function $f(z)$ such that*

$$\frac{h'(1)}{h(1)} = \frac{\partial}{\partial z} \frac{h(z, z_1, \dots, z_n)}{h(1, z_1, \dots, z_n)} \Big|_{z=1} = c + \sum_{k=1}^n f(z_k). \quad (3.24)$$

If

(*) the matrix $M(1, \hat{z}_1, \dots, \hat{z}_n)$ is non-singular for some $\hat{z}_1, \dots, \hat{z}_n$,

then conditions (a) and (b) are equivalent and (c) follows from them. If, moreover,

(**) there exist $1 \leq i < j \leq n$ such that $f'_i(1)f_j(1) \neq f'_j(1)f_i(1)$,

then all three conditions are equivalent. Also, if (**) does not hold, then (c) is satisfied for a constant function $f(z)$. Furthermore, if conditions (a) and (*) hold, then conditions (b) and (c) hold for

$$g(z) = \frac{B(z)}{B(1)}, \quad g(z, w) = \frac{C(z) - C(w)}{C(1) - C(w)}, \quad (3.25)$$

$$c = \frac{B'(1)}{B(1)}, \quad f(w) = \frac{C'(1)}{C(1) - C(w)}. \quad (3.26)$$

The proof of the theorem is given in the following subsections, where we prove implications (a) $\Rightarrow$ (b) (under (*)), (b) $\Rightarrow$ (c), (c) $\Rightarrow$ (a) (under conditions (*) and (**)) and (b) $\Rightarrow$ (a) (under (*)).

Remark 3.23 (Condition (*)). The non-singularity condition (*) means that equation (3.5) is well-defined at least for one choice of $\bar{z}_1, \dots, \bar{z}_n$, which is a natural requirement for the analysis of equation (3.1). Since $f_j(1) \neq 0$ for some $j = 0, \dots, n$, condition (*) is equivalent to linear independence between the functions $f_0(z), \dots, f_n(z)$, see Property 3.7 for $\Lambda_a = M(1, z_1, \dots, z_n)$.

Corollary 3.24. Suppose that the functions $f_k(z)$ are analytic in D_1 and linearly independent. Then,

- the pgf (3.1) has the factorisation property if and only if the functions $f_k(z)$ up to a linear transformation form a geometric sequence (3.22),
- the pgf (3.1) has the additive-mean property if and only if the functions $f_k(z)$ either satisfy

$$f'_i(1)f_j(1) = f'_j(1)f_i(1)$$

for all $i, j = 0, \dots, n$, or form up to a linear transformation a geometric sequence (3.22).

Remark 3.25 (Comparison with Chapter 2). in Chapter 2, Theorem 3.22 was partially proven for a specific function $C(z) = z/B(z)$. There, the focus was on proving (3.24) for certain systems such as the bulk-service queue and the FCTL queue. This result allows to use contour integrals for finding the average queue length. Theorem 3.22 generalises the result of Chapter 2 and describes all systems for which (3.24) applies. However, it does not mean that these are the only queueing systems for which the mean value can be found using one contour integral. In

this chapter, we have considered a special class of the pgf, see (3.1). If one is able to find the pgf in a factorisation form, e.g., as in (3.19), then the mean will be an additive function of the roots. For example, this result can be applied to the $G_D/G_D/1$ queue considered in van Ommeren (1991).

Remark 3.26 (Degenerate case). Note that from definition (3.18) of the function $h(z)$, it follows that if condition (**) does not hold, then (3.24) holds for $f(z) = 0$ and $c = f'_0(1)/f_0(1)$. Below, we give an example of a queueing system without condition (**). Consider a special **bulk-service queue** with vacations depending on the queue size. The arrivals are Poisson with rate 1. The size of the batch is 3 and the service time is deterministic and equal to some d such that $2 < d < 3$. If the server visits the queue and finds at least three customers, it immediately starts serving the first three customers. If upon a visit the server finds the queue with j customers, $j < 3$, it takes a vacation of deterministic time v_j with

$$\begin{aligned} v_0 &= -1 + \sqrt{d^2 - 4d + 7}, & v_1 &= -1 + \sqrt{d^2 - 3d + 3}, \\ v_2 &= d - 2. \end{aligned}$$

Note that for $d > 2$, time v_j is positive for $j = 0, 1, 2$. It is possible to find the pgf $X(z)$ of the queue length at the times when the server visits the queue, i.e., after a service or a vacation,

$$X(z) = \frac{\sum_{j=0}^2 \pi_j \hat{f}_j(z)}{z^3 - e^{d(z-1)}},$$

where π_j is the probability of finding j customers in the queue upon a visit, and the functions $\hat{f}_j(z) = (z-1)f_j(z)$ are defined as follows:

$$\hat{f}_j(z) = e^{v_j(z-1)} z^3 - z^j e^{d(z-1)}.$$

One can check that $f'_j(1) = \hat{f}''_j(1)/2 = 2f_j(1) = \hat{f}'_j(1)$ for all $j = 0, 1, 2$, which means that (**) does not hold, and, therefore, that (c) holds for $f(z) = 0$. Thus, the mean queue length upon the server arrival is independent of the roots of the characteristic equation $z^3 = e^{d(z-1)}$:

$$\begin{aligned} X'(1) &= \left(\frac{z-1}{z^3 - e^{d(z-1)}} \right)' \bigg|_{z=1} = (3-d) + \frac{\sum_{j=0}^2 \pi_j f'_j(1)}{\sum_{j=0}^2 \pi_j f_j(1)} = \\ &= -\frac{6-d^2}{6-2d} + 2 = \frac{d^2-4d+6}{6-2d}. \end{aligned}$$

3.3.1. Proof of Theorem 3.22: (a) $\Rightarrow$ (b)

Consider the matrix $M(z, z_1, \dots, z_n)$ at some point $(z, z_1, \dots, z_n)$ such that the matrix is non-singular. Given (a), the matrix $M(z, z_1, \dots, z_n)$ is a linear transfor-

mation of an **alternant** matrix with the functions $\tilde{f}_k(z) = B(z)(C(z))^k$:

$$M(z, z_1, \dots, z_n)(A^T)^{-1} = \begin{pmatrix} B(z) & \dots & B(z)C(z)^n \\ \vdots & \ddots & \vdots \\ B(z_n) & \dots & B(z_n)C(z_n)^n \end{pmatrix},$$

which is similar to a **Vandermonde matrix**. Hence, its determinant is equal to

$$\det(M(z, z_1, \dots, z_n)(A^T)^{-1}) = B(z) \prod_{k=1}^n B(z_k) V_C(z, \dots, z_n),$$

where $V_C(z, \dots, z_n) = V(C(z), C(z_1), \dots, C(z_n))$ is the **Vandermonde determinant** for variables $C(z), C(z_1), \dots, C(z_n)$. Therefore, given the fact that the matrices A and $M(1, z_1, \dots, z_n)$ are non-singular, we obtain

$$\frac{h(z)}{h(1)} = \frac{\det(M(z, z_1, \dots, z_n)(A^T)^{-1})}{\det(M(1, z_1, \dots, z_n)(A^T)^{-1})} = \frac{B(z)}{B(1)} \prod_{k=1}^n \frac{C(z) - C(z_k)}{C(1) - C(z_k)}. \quad (3.27)$$

This is exactly equation (3.23) with the functions $g(z)$ and $g(z, w)$ defined as in (3.25). Note that due to the continuity of the functions on the left-hand side and the right-hand side of (3.27) in their support, the equality holds also at points where the matrix $M(1, z_1, \dots, z_n)$ is singular, but $h(1) \neq 0$, see also **Remarks 3.5** and **3.12**.

3.3.2. Proof of **Theorem 3.22**: $(b) \Rightarrow (c)$

This implication does not require $(*)$ or $(**)$. Note that from (3.23) it follows that $g(1) \prod_{k=1}^n g(1, z_k) = 1$. Hence,

$$\frac{h'(1)}{h(1)} = \frac{\partial}{\partial z} \left(\frac{g(z)}{g(1)} \prod_{k=1}^n \frac{g(z, z_k)}{g(1, z_k)} \right) \Big|_{z=1} = \frac{g'(1)}{g(1)} + \sum_{k=1}^n \frac{\partial}{\partial z} \frac{g(z, z_k)}{g(1, z_k)} \Big|_{z=1}.$$

Thus, one can choose $c = g'(1)/g(1)$ and $f(w) = \partial/\partial z g(z, w)/g(1, w)|_{z=1}$. If the functions $g(z)$ and $g(z, w)$ are defined as in (3.25), then c and $f(w)$ are defined as in (3.26).

3.3.3. Proof of **Theorem 3.22**: $(c) \Rightarrow (a)$

This is the most involved part of the proof. First, we consider a linear transformation of the functions $f_k(z)$, given by the following lemma, see the proof in **Appendix 3.A.5**.

Lemma 3.27. *If the functions $f_k(z)$ are analytic in D_1 and linearly independent, then there exist a point $z^* \in D_1$ and the functions $\tilde{f}_j(z)$ such that $f_j(z) = \sum_{i=0}^n \tilde{a}_{j,i} \tilde{f}_i(z)$ and*

$$\tilde{f}_i(z) = (z - z^*)^i + o((z - z^*)^i) \text{ as } z \rightarrow z^*. \quad (3.28)$$

Moreover, z^* can be any point in D_1 except a finite set of points.

From the proof of the lemma, it follows that, given $(*)$, the function $\tilde{f}_{n-1}(z)$ can be chosen such that $\tilde{f}_{n-1}(1) \neq 0$ for all possible z^* except a finite set of points, see Remark 3.32 in Appendix 3.A.5. In our case, we can apply Lemma 3.27, because the matrix $M(1, \hat{z}_1, \dots, \hat{z}_n)$ is non-singular, and, therefore, the functions $f_0(z), \dots, f_n(z)$ are linearly independent. Since z^* can be any point in the unit disk except a finite number of points, we can assume, without loss of generality, that $z^* = 0$, $a_{i,j} = \delta_{ij}$ and $\tilde{f}_{n-1}(1) \neq 0$, where δ_{ij} is Kronecker delta.

Now, suppose that (3.24) holds. We will focus on determining $f_{k+1}(z)/f_k(z)$. For this, we will consider the cases where $z_{k+1} = \dots = z_n = 0$ for $k = 1, \dots, n$. The following lemma gives the value of $h(z)/h(1)$ for each k . The proof is given in Appendix 3.A.6.

Lemma 3.28. *For $k > 0$,*

$$\frac{h(z, z_1, \dots, z_k, 0, \dots, 0)}{h(1, z_1, \dots, z_k, 0, \dots, 0)} = \frac{\det \Lambda_k(z, z_1, \dots, z_k)}{\det \Lambda_k(1, z_1, \dots, z_k)}, \quad (3.29)$$

where

$$\Lambda_k(z, z_1, \dots, z_k) = \begin{pmatrix} f_{n-k}(z) & \dots & f_n(z) \\ f_{n-k}(z_1) & \dots & f_n(z_1) \\ \vdots & \ddots & \vdots \\ f_{n-k}(z_k) & \dots & f_n(z_k) \end{pmatrix}.$$

To find the function $f(z)$ in (3.24), we consider $k = 1$. Using Lemma 3.28, we find

$$\frac{h(z, z_1, 0, \dots, 0)}{h(1, z_1, 0, \dots, 0)} = \frac{f_n(z_1)f_{n-1}(z) - f_n(z)f_{n-1}(z_1)}{f_n(z_1)f_{n-1}(1) - f_n(1)f_{n-1}(z_1)}. \quad (3.30)$$

Let $C(z) = f_n(z)/f_{n-1}(z)$. Note that the function $C(z)$ is not a constant since the functions $f_n(z)$ and $f_{n-1}(z)$ are linearly independent. Using the function $C(z)$, we can rewrite (3.30) as

$$\frac{h(z, z_1, 0, \dots, 0)}{h(1, z_1, 0, \dots, 0)} = \frac{f_{n-1}(z) C(z) - C(z_1)}{f_{n-1}(1) C(1) - C(z_1)}. \quad (3.31)$$

Note that $C(1)$ and the right-hand side of (3.31) are well-defined since $f_{n-1}(1) = \tilde{f}_{n-1}(1) \neq 0$, see the choice of z^* . Taking the derivative gives us

$$\left. \frac{\partial}{\partial z} \frac{h(z, z_1, 0, \dots, 0)}{h(1, z_1, 0, \dots, 0)} \right|_{z=1} = \frac{f'_{n-1}(1)}{f_{n-1}(1)} + \frac{C'(1)}{C(1) - C(z_1)}.$$

Equation (3.24) defines the constant c and the function $f(w)$ up to a constant, i.e., constant c can be arbitrarily chosen. Therefore, we redefine $f(w)$ as $C'(1)/(C(1) - C(w))$. This leads to

$$c = \frac{f'_{n-1}(1)}{f_{n-1}(1)} - (n-1) \frac{C'(1)}{C(1)}$$

since $f(0) = C'(1)/C(1)$. Here, we used the fact that

$$C(0) = \lim_{z \rightarrow 0} f_n(z)/f_{n-1}(z) = \lim_{z \rightarrow 0} z^n/z^{n-1} = 0.$$

Now, it is left to prove that condition (a) follows from the equation

$$\left. \frac{\partial}{\partial z} \frac{h(z, z_1, \dots, z_n)}{h(1, z_1, \dots, z_n)} \right|_{z=1} = \frac{f'_{n-1}(1)}{f_{n-1}(1)} - (n-1) \frac{C'(1)}{C(1)} + \sum_{k=1}^n \frac{C'(1)}{C(1) - C(z_k)} \quad (3.32)$$

with $B(z) = f_n(z)/C(z)^n$. Recursive application of the following lemma together with a linear transformation of the functions $f_k(z)$ concludes the proof of implication (c) $\Rightarrow$ (a). The proof of Lemma 3.29 is given in Appendix 3.A.7.

Lemma 3.29. *Consider $k \geq 2$. Suppose that (**) is satisfied and equation (3.32) holds for $z_{k+1} = \dots = z_n = 0$. Suppose also that $f_{j+1}(z)/f_j(z) = C(z)$ for $j = n - k + 1, \dots, n - 1$. Then, there exist coefficients β_j , $j = 0, \dots, k$, such that*

$$f_{n-k}(z) = \beta_0 f_{n-k+1}(z)/C(z) + \sum_{j=1}^k \beta_j f_{n-k+j}(z). \quad (3.33)$$

Remark 3.30 (Importance of linear transformation). It would be sufficient to prove that $f_{k+1}(z) = C(z)f_k(z)$ for $k = 0, \dots, n - 2$. However, it may be not true. For example, the functions $1 + z$, z and z^2 satisfy conditions (a) - (c) and (3.28), but $z/(1 + z) \neq z = z^2/z$.

3.3.4. Proof of Theorem 3.22: (b) $\Rightarrow$ (a)

We prove (b) $\Rightarrow$ (a) under (*) similar to (c) $\Rightarrow$ (a) under (*) and (**). We proceed from equation (3.31) and define $g(z, w)$ as in (3.25). Note that equality (3.31) does not require condition (**), which we used only in Lemma 3.29. Using (3.31) and the definition of $g(z, w)$, we find that

$$g(z) = \frac{f_{n-1}(z)(C(1))^{n-1}}{f_{n-1}(1)(C(z))^{n-1}}.$$

Thus, from (3.23), we obtain

$$\frac{h(z, z_1, \dots, z_n)}{h(1, z_1, \dots, z_n)} = \frac{f_{n-1}(z)(C(1))^{n-1}}{f_{n-1}(1)(C(z))^{n-1}} \prod_{k=1}^n \frac{C(z) - C(z_k)}{C(1) - C(z_k)}. \quad (3.34)$$

Similar to the previous subsection, we conclude the proof by recursively applying the following lemma together with a linear transformation of the functions $f_k(z)$. The proof of Lemma 3.31 is similar to that of Lemma 3.29 and is given in Appendix 3.A.8.

Lemma 3.31. *Consider $k \geq 2$. Suppose that equation (3.34) holds for $z_{k+1} = \dots = z_n = 0$, and that $f_{j+1}(z)/f_j(z) = C(z)$ for $j = n - k + 1, \dots, n - 1$. Then, there exist coefficients β_j , $j = 0, \dots, k$, such that equation (3.33) holds.*

3.4. Conclusions

In this chapter, we analysed a class of pgfs with a rational form that often occurs in the analysis of queueing systems. The pgf contains several unknown coefficients that can be found using the roots of the characteristic equation. We gave an explicit matrix representation of the pgf in terms of the roots, see Theorem 3.21, where the matrix entries are symmetric functions of the roots. Our representation allows one to use the roots even if they are close to each other or coincide without encountering the corresponding sensitivity problems. Moreover, it is possible to find the pgf using contour integrals instead of computing the roots, which can further improve the accuracy.

We studied the cases where the pgf has a special product form, and where the mean value is an additive function of the roots. We showed that these properties are equivalent under a non-degeneracy condition and gave a necessary and sufficient condition for them, see Theorem 3.22. For systems with these properties, both the pgf at a point and the mean may be found using one contour integral. If the non-degeneracy condition does not hold, the mean is independent of the roots.

In the following Part III, we develop discrete-time models for traffic networks. The pgf of the queue length in the steady-state model of Chapters 4 and 5 has the rational form we analysed in this chapter. Therefore, we can apply results of Subsection 3.2 for this model.

3.A. Proofs of the auxiliary results

In this appendix, we give the proofs of the auxiliary results: Properties 3.7 and 3.8, Lemmas 3.11, 3.27 - 3.29 and 3.31 and bound (3.15).

3.A.1. Proof of Property 3.7

In this subsection, we prove Property 3.7. We use induction by n . Consider $n = 1$. Suppose the matrix

$$\begin{pmatrix} a_0 & a_1 \\ f_0(z_1) & f_1(z_1) \end{pmatrix}$$

is singular for all z_1 . Then, $a_0 f_1(z_1) = a_1 f_0(z_1)$ for all z_1 . If either a_0 or a_1 is non-zero, the functions are linearly dependent. Suppose we proved the statement

for $n - 1$. Using Laplace expansion, we find $0 = \det \Lambda_a = \sum_{k=0}^n \det \Lambda_{a,k} f_k(z_1)$, where $\Lambda_{a,k}$ is the matrix Λ_a without second row and $(k + 1)^{\text{st}}$ column. If $\det \Lambda_{a,k}$ as a function of $z_2, \dots, z_n$ is not identically 0 for some k , then the functions are linearly dependent. Now suppose that $\det \Lambda_{a,k}$ is identically 0 for all $k = 0, \dots, n$. Without loss of generality, we can assume $a_1 \neq 0$. Applying the property for the $n \times n$ matrix $\Lambda_{a,0}$, we find that the functions $f_1(z), \dots, f_n(z)$ are linearly dependent, and so are the functions $f_0(z), \dots, f_n(z)$.

3.A.2. Proof of Property 3.8

In this subsection, we prove [Property 3.8](#). Recall that we need to prove that

$$\sum_{j=0}^n (-1)^j \sigma_j \zeta_{m-j} = 0$$

for any $m > 0$. The proof will be done using generating functions. First, we note that

$$\sum_{m=0}^{\infty} \zeta_m z^m = \prod_{i=1}^{n-1} \sum_{k=0}^{\infty} (z_i z)^k = \prod_{i=1}^{n-1} \frac{1}{1 - z_i z}.$$

The above equality holds for sufficiently small z , i.e., for $|z| < \min_{i=1, \dots, n-1} 1/|z_i|$. Second, from (3.8), we get

$$\prod_{i=1}^{n-1} (1 - z_i z) = z^{n-1} \prod_{i=1}^{n-1} (z^{-1} - z_i) = \sum_{j=0}^{n-1} (-1)^j \sigma_j z^j.$$

Hence,

$$1 = \prod_{i=1}^{n-1} \frac{1 - z_i z}{1 - z_i z} = \sum_{m=0}^{\infty} \zeta_m z^m \sum_{j=0}^{n-1} (-1)^j \sigma_j z^j.$$

Note that the last equation is an equality of two analytic functions. Thus, the coefficients at powers of z should coincide. Result (3.9) follows from considering the coefficient at z^m for $m > 0$.

3.A.3. Proof of Lemma 3.11

In this subsection, we prove [Lemma 3.11](#). We use the first Jacobi-Trudi formula, see [Bekker et al. \(2016\)](#), which can be written as

$$\det \begin{pmatrix} z_1^{m_1} & \cdots & z_1^{m_n} \\ \vdots & \ddots & \vdots \\ z_n^{m_1} & \cdots & z_n^{m_n} \end{pmatrix} = V(z_1, \dots, z_n) \det \begin{pmatrix} \zeta_{m_1} & \cdots & \zeta_{m_n} \\ \vdots & \ddots & \vdots \\ \zeta_{m_1-n+1} & \cdots & \zeta_{m_n-n+1} \end{pmatrix}. \quad (3.35)$$

It is used for the Schur polynomials, for which $m_1 > \dots > m_n$. However, the result is general. In particular, a permutation of rows gives the result for any $m_1, \dots, m_n$

such that $m_i \neq m_j$ for any $i \neq j$. Note also that if $m_i = m_j$ for $i \neq j$, then both sides of (3.35) are equal to 0.

Lemma 3.11 follows from equation (3.35) by summing it for all possible combinations $(m_1, \dots, m_n)$ with coefficients $\prod_{k=1}^n \alpha_k z^k$. Note also that (3.35) is a special case of Lemma 3.11.

3.A.4. Proof of bound (3.15)

In this subsection, we prove bound (3.15). Suppose $|\bar{z}_k| \leq q$ for $k = 1, \dots, n$, and $|\alpha_l| \leq Cr^l$ for $l = 0, 1, \dots$, where $f(z) = \sum_{k=0}^{\infty} \alpha_k z^k$. The latter condition holds if the function $f(z)$ is analytic in a disk with radius more than $1/r$. Suppose also that $qr < 1$. Then, we can give the following bound

$$\begin{aligned}
 \left| F_k^n(\bar{z}_1, \dots, \bar{z}_n) - \sum_{k=0}^M \alpha_k \zeta_k(\bar{z}_1, \dots, \bar{z}_n) \right| &= \left| \sum_{k=M+1}^{\infty} \alpha_k \zeta_k(\bar{z}_1, \dots, \bar{z}_n) \right| \leq \\
 &\leq \sum_{k=M+1}^{\infty} |\alpha_k| |\zeta_k(\bar{z}_1, \dots, \bar{z}_n)| \leq C \sum_{k=M+1}^{\infty} r^k \zeta_k(|\bar{z}_1|, \dots, |\bar{z}_n|) \leq \\
 &\leq C \sum_{k=M+1}^{\infty} r^k \zeta_k(q, \dots, q) \leq C \sum_{k=M+1}^{\infty} \binom{k+n-1}{k} (qr)^k \stackrel{!}{=} \\
 &\stackrel{!}{=} C \binom{M+n}{M} \frac{\sum_{l=1}^n (-1)^{l+1} \binom{n}{l} \frac{1}{M+l} (qr)^{M+l}}{(1-qr)^n} = \\
 &= C \binom{M+n}{M} \frac{n \sum_{l=0}^{n-1} (-1)^l \binom{n-1}{l} \frac{1}{M+l+1} (qr)^{M+l+1}}{(1-qr)^n} = \\
 &= C \binom{M+n}{n} \frac{n \int_0^{qr} x^M (1-x)^{n-1} dx}{(1-qr)^n} \leq C \binom{M+n}{n} \frac{n \int_0^{qr} x^M dx}{(1-qr)^n} = \\
 &= C \binom{M+n}{n} \frac{n (qr)^{M+1}}{(M+1)(1-qr)^n} = C \binom{M+n}{n-1} \frac{(qr)^{M+1}}{(1-qr)^n}.
 \end{aligned}$$

Equality $\stackrel{!}{=}$ can be proven using induction by M as follows. Consider $S_{n,M} = S_{n,M}(x) = \sum_{k=M+1}^{\infty} \binom{k+n-1}{k} x^k$. For $M = 0$, we get

$$S_{n,0} = \left(\sum_{j=0}^{\infty} x^j \right)^n - 1 = \frac{1}{(1-x)^n} - 1 = \frac{\sum_{l=1}^n (-1)^{l+1} \binom{n}{l} x^l}{(1-x)^n}.$$

Now suppose we have proved the statement for $S_{n,M-1}$. Consider $S_{n,M}$:

$$S_{n,M}(1-x)^n = \left(S_{n,M-1} - \binom{M+n-1}{M} x^M \right) (1-x)^n =$$

$$\begin{aligned}
&= \binom{M+n-1}{M-1} \sum_{l=1}^n (-1)^{l+1} \binom{n}{l} \frac{l}{M+l-1} x^{M+l-1} - \binom{M+n-1}{M} x^M (1-x)^n = \\
&= \binom{M+n-1}{M-1} \sum_{l=0}^{n-1} (-1)^l \binom{n}{l+1} \frac{l+1}{M+l} x^{M+l} - \binom{M+n-1}{M} \sum_{l=0}^n (-1)^l \binom{n}{l} x^{M+l} = \\
&= \sum_{l=0}^n (-1)^l \left[\frac{(M+n-1)!(n-l)}{(M-1)!!(n-l)!(M+l)} - \frac{(M+n-1)!n}{M!!(n-l)!} \right] x^{M+l} = \\
&= \sum_{l=0}^n (-1)^l \frac{(M+n-1)!}{(M-1)!!(n-l)!} \left(\frac{n-l}{M+l} - \frac{n}{M} \right) x^{M+l} = \\
&= \sum_{l=0}^n (-1)^{l+1} \frac{(M+n)!}{M!!(n-l)!} \frac{l}{M+l} x^{M+l} = \binom{M+n}{M} \sum_{l=1}^n (-1)^{l+1} \binom{n}{l} \frac{l}{M+l} x^{M+l}.
\end{aligned}$$

3.A.5. Proof of Lemma 3.27

In this subsection, we prove [Lemma 3.27](#). Recall that we need to prove that there exists a point $z^* \in D_1$ such that the functions $f_j(z)$ after a linear transformation give the functions $\tilde{f}_i(z)$ that are locally equal to $(z - z^*)^i + o((z - z^*)^i)$.

Functions $f_j(z)$, $j = 0, \dots, n$, are analytic and linearly independent. Therefore, the [Wronskian](#)

$$\det W(z) = \det \begin{pmatrix} f_0(z) & f_1(z) & \dots & f_n(z) \\ f'_0(z) & f'_1(z) & \dots & f'_n(z) \\ \vdots & \vdots & \ddots & \vdots \\ f_0^{(n)}(z) & f_1^{(n)}(z) & \dots & f_n^{(n)}(z) \end{pmatrix}$$

is not identically 0, see [Property 3.6](#). Note that $\det W(z)$ is an analytic function in D_1 and, therefore, it has not more than a finite number of zeroes in D_1 . Let $z^* \in D_1$ be such that $\det W(z^*) \neq 0$. Define the functions $\tilde{f}_0(z), \dots, \tilde{f}_n(z)$ by the following linear combination:

$$\begin{aligned}
(0! \cdot \tilde{f}_0(z) \quad 1! \cdot \tilde{f}_1(z) \quad \dots \quad n! \cdot \tilde{f}_n(z)) &= \\
&= (f_0(z) \quad f_1(z) \quad \dots \quad f_n(z)) (W(z^*))^{-1}. \quad (3.36)
\end{aligned}$$

Consider the matrix $\tilde{W}(z)$ for the functions $0! \tilde{f}_0(z), \dots, n! \tilde{f}_n(z)$ defined as

$$\tilde{W}(z) = \begin{pmatrix} \tilde{f}_0(z) & \tilde{f}_1(z) & \dots & n! \cdot \tilde{f}_n(z) \\ \tilde{f}'_0(z) & \tilde{f}'_1(z) & \dots & n! \cdot \tilde{f}'_n(z) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{f}_0^{(n)}(z) & \tilde{f}_1^{(n)}(z) & \dots & n! \cdot \tilde{f}_n^{(n)}(z) \end{pmatrix}.$$

At point z^* , we get that $\tilde{W}(z^*) = W(z^*)(W(z^*))^{-1}$ becomes the identity matrix. Thus, $\tilde{f}_k^{(l)}(z^*) = 0$ for $l < k$ and $k! \tilde{f}_k^{(k)}(z^*) = 1$, which means that the functions $\tilde{f}_0(z), \dots, \tilde{f}_n(z)$ satisfy (3.28).

Remark 3.32 (Non-zero value at 1). Given condition (*), see Theorem 3.22, one can choose z^* and the functions $\tilde{f}_k(z)$ such that for a particular index k , $\tilde{f}_k(1) \neq 0$. To see this, let us first consider the case $k = n$. Since $f_j(1) \neq 0$ for at least one j , we can consider a linear transformation of the functions $f_k(z)$ such that $f_0(1) \neq 0$, and $f_k(1) = 0$ for all $k \neq 0$. Then $\tilde{f}_n(1) \neq 0$ if and only if the coefficient of $f_0(z)$ in the definition of $\tilde{f}_n(z)$, see (3.36), is non-zero. This coefficient, up to multiplication by $\det W(z^*)$ and a sign, is equal to the Wronskian for the functions $f_1(z), \dots, f_n(z)$ at point z^* , which is non-zero for all possible z^* except a finite set. Note that if $\tilde{f}_n(1) \neq 0$, then either $\tilde{f}_k(1) \neq 0$ or $\tilde{f}_k(1) + \tilde{f}_n(1) \neq 0$. Therefore, the result for all k follows from the fact that $\tilde{f}_k(z) + \tilde{f}_n(z)$ satisfies (3.28) for $i = k$.

3.A.6. Proof of Lemma 3.28

In this subsection, we prove Lemma 3.28. We need to prove (3.29), i.e.,

$$\frac{h(z, z_1, \dots, z_k, 0, \dots, 0)}{h(1, z_1, \dots, z_k, 0, \dots, 0)} = \frac{\det \Lambda_k(z, z_1, \dots, z_k)}{\det \Lambda_k(1, z_1, \dots, z_k)},$$

where $\Lambda_k(z, z_1, \dots, z_k)$ is an [alternant](#) matrix constructed using the functions $f_{n-k}(z), \dots, f_n(z)$ and the points $z, z_1, \dots, z_k$.

Recall that the function $h(z, z_1, \dots, z_k, 0, \dots, 0)$ is the determinant of the matrix

$$\bar{M}(z, z_1, \dots, z_k, 0, \dots, 0),$$

which entries are $F_{j,m}^n(z_1, \dots, z_k, 0, \dots, 0)$. From Property 3.9, we get

$$F_{j,m}^n(z_1, \dots, z_k, 0, \dots, 0) = F_{j,m}^k(z_1, \dots, z_k).$$

Now, according to Property 3.10,

$$F_{j,m}^k + \sum_{l=1}^k (-1)^l \sigma_l(z_1, \dots, z_k) F_{j,m+l}^k = \alpha_{j,m},$$

where $f_j(z) = \sum_{l=0}^{\infty} \alpha_{j,l} z^l$. Thus, after a linear transformation, we get that the matrix $\bar{M}(z, z_1, \dots, z_k, 0, \dots, 0)$ changes to

$$\begin{pmatrix} f_0(z) & \dots & f_n(z) \\ \alpha_{0,0} & \dots & \alpha_{n,0} \\ \vdots & \ddots & \vdots \\ \alpha_{0,n-k-1} & \dots & \alpha_{n,n-k-1} \\ F_{0,n-k}^k & \dots & F_{n,n-k}^k \\ \vdots & \ddots & \vdots \\ F_{0,n-1}^k & \dots & F_{n,n-1}^k \end{pmatrix}. \quad (3.37)$$

Here, we added the $(m + l + 2)^{\text{nd}}$ row multiplied by $(-1)^l \sigma_l(z_1, \dots, z_k)$ to the $(m + 2)^{\text{nd}}$ row for $l = 1, \dots, k$ and $m = 0, \dots, n - k - 1$. Now, note that $\alpha_{j,k} = 0$ for $j > k$ and $\alpha_{j,j} = 1$, which means that the matrix in (3.37) is equal to

$$\begin{pmatrix} f_0(z) & \dots & f_{n-k-1}(z) & f_{n-k}(z) & \dots & f_n(z) \\ 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \alpha_{0,n-k-1} & \dots & 1 & 0 & \dots & 0 \\ F_{0,n-k}^k & \dots & F_{n-k-1,n-k}^k & F_{n-k,n-k}^k & \dots & F_{n,n-k}^k \\ \dots & \dots & \dots & \dots & \dots & \dots \\ F_{0,n-1}^k & \dots & F_{n-k-1,n-1}^k & F_{n-k,n-1}^k & \dots & F_{n,n-1}^k \end{pmatrix}.$$

Hence, the determinant is equal to

$$h(z, z_1, \dots, z_k, 0, \dots, 0) = (-1)^{n-k} \det \begin{pmatrix} f_{n-k}(z) & \dots & f_n(z) \\ F_{n-k,n-k}^k & \dots & F_{n,n-k}^k \\ \vdots & \ddots & \vdots \\ F_{n-k,n-1}^k & \dots & F_{n,n-1}^k \end{pmatrix}. \quad (3.38)$$

At this moment, we can use Lemma 3.11, to find that up to multiplication by the Vandermonde determinant $V(z_1, \dots, z_k)$, the determinant of the matrix on the right-hand side of (3.38) is equal to the determinant of an almost alternant matrix

$$V(z_1, \dots, z_k) \det \begin{pmatrix} f_{n-k}(z) & \dots & f_n(z) \\ F_{n-k,n-k}^k & \dots & F_{n,n-k}^k \\ \vdots & \ddots & \vdots \\ F_{n-k,n-1}^k & \dots & F_{n,n-1}^k \end{pmatrix} = \det \begin{pmatrix} f_{n-k}(z) & \dots & f_n(z) \\ \frac{f_{n-k}(z_1)}{z_1^{n-k}} & \dots & \frac{f_n(z_1)}{z_1^{n-k}} \\ \vdots & \ddots & \vdots \\ \frac{f_{n-k}(z_k)}{z_k^{n-k}} & \dots & \frac{f_n(z_k)}{z_k^{n-k}} \end{pmatrix}, \quad (3.39)$$

which is not identically equal to 0 due to the linear independence of the functions $f_{n-k}(z), \dots, f_n(z)$. Here, we used the fact that $f_l(z)$, $l = n - k, \dots, n$, satisfies (3.28), and, therefore, has first $n - k - 1$ coefficients in the Taylor expansion equal to 0. Hence, the (m, k) -transformation of the function $f_l(z)/z^{n-k} = \sum_{j=n-k}^{\infty} \alpha_j z^{j-n+k}$ is equal to $F_{l,n-k+m}^k = \sum_{j=n-k+m}^{\infty} \alpha_j \zeta_{j-n+k-m}$. Combining (3.38) and (3.39), gives (3.29) and, therefore, concludes the proof.

3.A.7. Proof of Lemma 3.29

In this subsection, we prove Lemma 3.29. In this lemma, we assume that equation (3.32) holds for $z_{k+1} = \dots = z_n = 0$, i.e.,

$$\left. \frac{\partial}{\partial z} \frac{h(z, z_1, \dots, z_k, 0, \dots, 0)}{h(1, z_1, \dots, z_k, 0, \dots, 0)} \right|_{z=1} = \frac{f'_{n-1}(1)}{f_{n-1}(1)} - (k-1) \frac{C'(1)}{C(1)} + \sum_{j=1}^k \frac{C'(1)}{C(1) - C(z_j)},$$

and that $f_{j+1}(z) = C(z)f_j(z)$ for $j = n - k + 1, \dots, n - 1$. We need to prove that the function $f_{n-k}(z)$ is equal (up to a linear combination of the functions $f_{n-k+1}(z), \dots, f_n(z)$) to $\beta_0 f_{n-k+1}/C(z)$.

First, we apply Lemma 3.28 and get

$$\frac{h(z, z_1, \dots, z_k, 0, \dots, 0)}{h(1, z_1, \dots, z_k, 0, \dots, 0)} = \frac{f_{n-k+1}(z) \det M_G(z, z_1, \dots, z_k)}{f_{n-k+1}(1) \det M_G(1, z_1, \dots, z_k)}, \quad (3.40)$$

where

$$M_G(z, z_1, \dots, z_k) = \begin{pmatrix} \frac{1}{G(z)} & 1 & \dots & C(z)^{k-1} \\ \frac{1}{G(z_1)} & 1 & \dots & C(z_1)^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{G(z_k)} & 1 & \dots & C(z_k)^{k-1} \end{pmatrix}, \quad (3.41)$$

and $G(z) = f_{n-k+1}(z)/f_{n-k}(z)$.

Second, we find the derivative of (3.40) at 1:

$$\begin{aligned} \left. \frac{\partial}{\partial z} \frac{h(z, z_1, \dots, z_k, 0, \dots, 0)}{h(1, z_1, \dots, z_k, 0, \dots, 0)} \right|_{z=1} &= \\ &= \frac{f'_{n-k+1}(1)}{f_{n-k+1}(1)} + \left. \frac{\partial}{\partial z} \frac{\det M_G(z, z_1, \dots, z_k)}{\det M_G(1, z_1, \dots, z_k)} \right|_{z=1}. \end{aligned} \quad (3.42)$$

Note that $f_{n-1}(z) = f_{n-k+1}(z)C(z)^{k-2}$. Thus,

$$\frac{f'_{n-1}(1)}{f_{n-1}(1)} = \frac{f'_{n-k+1}(1)}{f_{n-k+1}(1)} + (k-2) \frac{C'(1)}{C(1)}. \quad (3.43)$$

Combining equations (3.32), (3.42) and (3.43), we get that

$$\left. \frac{\partial}{\partial z} \frac{\det M_G(z, z_1, \dots, z_k)}{\det M_G(1, z_1, \dots, z_k)} \right|_{z=1} = -\frac{C'(1)}{C(1)} + \sum_{l=1}^k \frac{C'(1)}{C(1) - C(z_l)}. \quad (3.44)$$

Third, we prove that the functions $C(z_1)/G(z_1), 1, \dots, C(z_1)^k$ are linearly dependent if $C'(1) \neq 0$, which we show later. Let $\mu_j(z)$ be the determinant of the matrix $M_G(z, z_1, \dots, z_k)$ without the second row and $(j+1)^{\text{st}}$ column. Note that $\mu_j(z)$ does not depend on z_1 . Fix any $z_2, \dots, z_k$ such that $\mu_0(1) \neq 0$, which is possible since the function $C(z)$ is not constant. By multiplying both sides of (3.44) by $\det M_G(1, z_1, \dots, z_k)$, we get

$$\begin{aligned} \frac{1}{G(z_1)} \mu'_0(1) + \sum_{l=0}^{k-1} \mu'_{l+1}(1) C(z_1)^l &= -\frac{1}{G(z_1)} \mu_0(1) \frac{C'(1)}{C(1)} + \\ &+ \frac{1}{G(z_1)} \mu_0(1) \sum_{l=1}^k \frac{C'(1)}{C(1) - C(z_l)} + \end{aligned}$$

$$+ \sum_{l=0}^{k-1} \mu_{l+1}(1) C(z_1)^l \left(-\frac{C'(1)}{C(1)} + \sum_{l=1}^k \frac{C'(1)}{C(1) - C(z_l)} \right). \quad (3.45)$$

Note that $\mu_0(z) = V(C(z), C(z_2), \dots, C(z_k))$. Hence,

$$\mu'_0(1) = \mu_0(1) \sum_{l=2}^k \frac{C'(1)}{C(1) - C(z_l)}.$$

Therefore, we can rewrite equation (3.45) as

$$\begin{aligned} \frac{C(z_1)}{G(z_1)} \mu_0(1) \frac{C'(1)}{C(1)(C(1) - C(z_1))} = \\ = \sum_{l=0}^{k-1} C(z_1)^l \left(\mu'_{l+1}(1) + \mu_{l+1}(1) \frac{C'(1)}{C(1)} - \mu_{l+1}(1) \sum_{l=1}^k \frac{C'(1)}{C(1) - C(z_l)} \right). \end{aligned} \quad (3.46)$$

Note that multiplying both sides by $C(1) - C(z_1)$ will give a linear dependency between the functions $C(z_1)/G(z_1)$, $1, \dots, C(z_1)^k$, in which the coefficient for $C(z_1)/G(z_1)$ is non-zero (if $C'(1) \neq 0$). To get (3.33), one needs to multiply both sides of equation (3.46) by $f_{n-k+1}(z_1) C(1) (C(1) - C(z_1)) / (C(z_1) \mu_0(1) C'(1))$.

Finally, we prove that $C'(1) \neq 0$. Suppose $C'(1) = 0$. We will prove that this contradicts (**). Note that $c = f'_{n-1}(1)/f_{n-1}(1)$ and, due to (3.32),

$$h'(1, z_1, \dots, z_n) = c h(1, z_1, \dots, z_n).$$

This means that the matrix

$$\begin{pmatrix} f'_0(1) - c f_0(1) & \dots & f'_n(1) - c f_n(1) \\ f_0(z_1) & \dots & f_n(z_1) \\ \vdots & \ddots & \vdots \\ f_0(z_n) & \dots & f_n(z_n) \end{pmatrix}$$

is singular for all $z_1, \dots, z_n$. Since the functions $f_0(z), \dots, f_n(z)$ are linearly independent, we get that $f'_i(1) = c f_i(1)$ for $i = 0, \dots, n$, which contradicts with (**).

3.A.8. Proof of Lemma 3.31

In this subsection, we prove Lemma 3.31 similar to Lemma 3.29. In this lemma, we assume that equation (3.34) holds for $z_{k+1} = \dots = z_n = 0$, i.e.,

$$\frac{h(z, z_1, \dots, z_k, 0, \dots, 0)}{h(1, z_1, \dots, z_k, 0, \dots, 0)} = \frac{f_{n-1}(z)(C(1))^{k-1}}{f_{n-1}(1)(C(z))^{k-1}} \prod_{j=1}^k \frac{C(z) - C(z_j)}{C(1) - C(z_j)}.$$

and that $f_{j+1}(z) = C(z)f_j(z)$ for $j = n - k + 1, \dots, n - 1$. We need to prove that the function $f_{n-k}(z)$ is equal (up to a linear combination of the functions $f_{n-k+1}(z), \dots, f_n(z)$) to $\beta_0 f_{n-k+1}/C(z)$.

Using (3.40) and $f_{n-1}(z) = f_{n-k+1}(z)C(z)^{k-2}$, we obtain

$$\frac{\det M_G(z, z_1, \dots, z_k)}{\det M_G(1, z_1, \dots, z_k)} = \frac{C(1)}{C(z)} \prod_{j=1}^k \frac{C(z) - C(z_j)}{C(1) - C(z_j)}, \quad (3.47)$$

where the matrix $M_G(z, z_1, \dots, z_k)$ is defined in (3.41).

From (3.47), we prove that the functions $C(z)/G(z)$, $1, \dots, C(z)^k$ are linearly dependent. Let μ_j be the determinant of the matrix $M_G(z, z_1, \dots, z_k)$ without the *first* row and $(j+1)^{\text{st}}$ column. Fix any $z_1, \dots, z_k$ such that $\mu_0 \neq 0$, which is possible since the function $C(z)$ is not constant. By multiplying both sides of (3.47) by $\det M_G(1, z_1, \dots, z_k)C(z)$, we obtain

$$\frac{C(z)}{G(z)}\mu_0 + \sum_{l=1}^k \mu_l C(z)^l = C(1) \left(\frac{C(1)}{G(1)}\mu_0 + \sum_{l=1}^k \mu_l C(1)^l \right) \prod_{j=1}^k \frac{C(z) - C(z_j)}{C(1) - C(z_j)}.$$

Note that the product on the right-hand side can be rewritten as a linear combination of the functions $1, \dots, C(z)^k$. This observation concludes the proof.

Remark 3.33 (Comparison to the proof of Lemma 3.29). In the proof in Appendix 3.A.7, we expand an *alternant* matrix over the row that depends on z_1 . In this case, the result requires an additional condition $C'(1) \neq 0$, which follows from (**). In this subsection, we can expand a similar matrix over the first row that depends on z (in the proof of Lemma 3.29 this is a constant row and we cannot use such an expansion). For this reason, we do not need the extra condition (**).

Part III

Traffic networks

Chapter 4

A tandem of intersections under semi-actuated and fixed control

‘If you understand how the universe operates, you control it, in a way.’

Stephen Hawking

With increasing congestion in urban areas, it is important to understand the differences between existing traffic control policies in terms of the performance and the optimal settings choice. This requires a thorough analysis of possible control policies. In this chapter, we develop a stochastic model that enables performance analysis of a tandem of [traffic-light intersections](#) under *fixed* and *semi-actuated* control. Under fixed control, the green and red times of a *lane* are fixed and independent of the actual traffic conditions. When traffic detectors are available at minor lanes, one may use a semi-actuated control policy. Then, the traffic light for a minor lane switches to yellow as soon as the lane empties or the green-time limit for this lane is reached. All saved green time is added to the green time for the busiest directions, thus providing a higher capacity compared to fixed control. In the traffic literature, this type of control is referred as semi-actuated control with *floating force-off*, see [Koonce and Rodegerdts \(2008\)](#).

Our model is an extension and generalisation of the [FCTL](#) model, see [Chapter 2](#). Compared to the FCTL model, we include a more realistic service process that provides longer inter-departure times for the first vehicles departing from a lane after the traffic light switches to green. Moreover, to model drivers’ distraction, we introduce randomness in the departures. The main challenge for the analysis of closely-spaced intersections is that the arrivals at a downstream intersection are correlated. We resolve this problem by representing the arrivals as a discrete-time [MAP](#). Another important contribution of our model is the inclusion

Figure 4.1: A tandem of two traffic-light intersections. There is an arterial road (west-east) and the minor roads (from north and south). The major road has one lane for each direction, i.e., for right-turning, straight-going and left-turning movements. At minor lanes the induction-loop detectors are installed.

of the [offsets](#) between the intersections. The properly chosen offsets can result in smooth flow of the vehicles in the system and short delays, see also [Chapter 5](#).

In the numerical results, we validate our model using the traffic microsimulation suite [SUMO](#), which was previously used for comparison of different traffic-light control policies, see, e.g., [Krajzewicz et al. \(2005\)](#) and [Liang et al. \(2018\)](#). We show that our model accurately predicts the delay for each lane and the average delay of an arbitrary vehicle. Compared to SUMO, our model is numerically substantially less demanding. For example, one scenario with fixed control is evaluated in less than 0.5 seconds using our model and in 20 minutes using SUMO, which makes our model well-suited for what-if analysis and optimisation.

To evaluate which type of control gives the best performance, we consider a tandem of two intersections with one main road and several minor roads, see [Figure 4.1](#). We show that semi-actuated control provides more capacity and shorter delays for the main direction but is disadvantageous for the minor lanes. The effect on the average delay per vehicle depends on the settings and traffic intensities. When the load at the minor lanes is high, the average delay under fixed control is lower than under semi-actuated control. We use our model to find the optimal traffic-light settings, i.e., green times and offsets, and show that they heavily depend on the type of control. Under semi-actuated control, it may be better to allocate more green time for the minor lanes, while, under fixed control, the major lanes may require longer green times. The optimal offsets can vary significantly depending on the load in the system for semi-actuated control and are almost the

Figure 4.2: An example of traffic-light phases for an intersection in Figure 4.1.

same for all loads under fixed control.

This chapter is structured as follows. In Section 4.1, we present our analytical model for both fixed and semi-actuated control. Section 4.2 provides numerical results. First, we justify our assumptions, find model parameters and validate our model using SUMO. Then, we compare fixed and semi-actuated control in terms of performance and optimal settings. In Section 4.3, we conclude the chapter.

4.1. Model

In this section, we present an analytical model for a tandem of two intersections under fixed or semi-actuated control. For both controls, the traffic lights change according to a *phase schedule*, where a *phase* is a combination of traffic lights that have green time simultaneously, see, for example, Figure 4.2. In the case of semi-actuated control, we distinguish between *major lanes* without traffic detectors and *minor lanes* with detectors. The phase schedule has one or more parts, each consisting of subsequent phases for minor lanes, e.g., phases 1 and 2 in Figure 4.2, and ending with one phase for major lanes, e.g., phase 3. The time saved by terminating the phases early due to detection of emptied queues is used for the major lanes, see Figure 4.3 for an example of a phase schedule and its realisation. There, phase 4 forms the second part of the phase schedule. Note that if a part of the phase schedule consists of only one phase, the lanes corresponding to this phase have fixed green time. Therefore, fixed control can be seen as a special case of semi-actuated control for which each part of the phase schedule consists of one phase. For this reason, in what follows, we consider only semi-actuated control.

As in the FCTL model, we represent each lane as a queue and focus on the steady-state queue length at this lane. Compared to the case of an isolated intersection under fixed control, the queues in the network are correlated. We include this correlation in the arrival and service processes of the queues. Queues for minor and major lanes have different service processes and are analysed in Subsections 4.1.1 and 4.1.2, respectively. In Subsection 4.1.3, we examine how the outputs of these queues for one intersection form the input of a downstream queue

Figure 4.3: An example of a phase schedule with semi-actuated control and its realisation. Lanes 1 - 4 correspond to phases 1 - 4 in Figure 4.2, respectively. Lanes 1 and 2 have detectors and green-time limits g_1 , g_2 , respectively. Lane 1 empties before the limit is reached, and the traffic light switches to green for lane 2. The green-time limit for lane 2 is reached. Thus, the traffic light switches to green for lane 3, even though there may be a non-empty queue at lane 2. Lane 3 empties before the end of the green time, and there are free departures during the rest of its green time. Lane 4 has a fixed green-time length. The (random) beginning time of green periods at lanes u is denoted by B_u . The departure of the k^{th} delayed vehicle from lane u occurs at second $B_u + d_k$. During the switch time between lanes u and $u + 1$, the traffic light is yellow (amber) for lane u .

at the other intersection. Below, we give the common modelling assumptions for both types of queues.

We consider a discrete-time model with time split in seconds. The numeration of seconds in a cycle starts with 0. We will identify a second with its beginning. In general, the lengths of the red and green times of a lane are random because they depend on the random green time of the minor lanes. In special cases, these lengths can be deterministic, for example, if a part of phase schedule consists of only one phase, see phase 4 in Figure 4.3. In such cases, the corresponding random variables are constant. In our model, the cycle length is fixed and equal to c seconds, which enables coordination of the cycles at subsequent intersections. We do not explicitly model yellow (amber) time. Instead, we use switch time

between phases, so that the intersection can be cleared before the following phase starts.

We assume that *queued vehicles*, i.e., vehicles waiting in the queue, can depart only at predetermined times $\{B + d_k\}$, $k = 1, 2, \dots$, during a green period, where B is the random variable of the green-time beginning, $0 \leq B < c$, and the departure moments d_k are deterministic. More precisely, if at second $B + d_k$ the traffic light for this lane is still green and there is a queue at the lane, there will be one departure. Such departures we call *delayed departures*. The moments d_k , $k = 1, 2, \dots$, depend on the average vehicle characteristics such as maximum speed and acceleration. We can view second $B + d_{k-1} + 1$ as *the beginning of the k^{th} departure*, where, for convenience, we set $d_0 = -1$. This departure takes $d_k - d_{k-1}$ seconds and finally occurs (if the green light is still on) at second $B + d_k$. Using d_k allows us to model different inter-departure times for the vehicles at the front of the queue. Later, in [Remark 4.6](#), we include drivers' distraction by introducing extra randomness at the beginning of the first delayed departure, while keeping the inter-departure times deterministic, see also [Remark 4.9](#). For a lane without a detector, if the queue is empty during a green second, all arriving vehicles can proceed without stopping, see [Figure 4.3](#). These departures we call *free departures*.

Remark 4.1 (Straight-going and turning flows). In [Chapter 2](#), we showed that the difference in the delay between straight-going and turning flows is small. For this reason, in the models of [Part III](#), we do not distinguish these flows and use free departures similar to the [FCTL](#) model.

For the beginning of the green times, we make the following assumption:

Assumption 4.2 (Independence of the green-time beginnings). The green-time beginning moments of one lane are independent of each other at different cycles and of the queue length at the lane. Moreover, these beginning moments are identically distributed and their distribution is known.

The distribution of the beginning time B follows from the analysis of the minor lanes that are served earlier in the phase schedule. In what follows, probability $\mathbb{P}(B = s)$ is denoted by κ_s , $s = 0, \dots, c - 1$. The assumption of independence between cycles holds in the case of fixed control since the green-time beginning B is deterministic. Under semi-actuated control, B depends on the queues at the minor lanes, which show certain correlation in subsequent cycles. Thus, the value of B in one cycle depends on its value in the last cycle. In [Subsection 4.2.1](#), we show that this dependency has a negligible effect on the numerical results. [Assumption 4.2](#) plays an important role in our analysis and allows us to maintain a tractable model, see the following [Remark 4.3](#).

Remark 4.3 (Independent cycles). [Assumption 4.2](#) for every lane and [Assumption 4.7](#) for a lane without detectors (on independence of arrivals, see below) make the beginning of a cycle a regeneration point rendering i.i.d. cycles. This allows

us to use polling-models techniques for the analysis of our model, see Subsections 4.1.1.2 and 4.1.2.3.

Remark 4.4 (Time units). Compared to the FCTL model, we use a shorter time interval of one second, which is a common time unit for traffic-light settings. This allows us to model a more realistic service process and to avoid problems discussed in Subsection 1.7.2.1.

4.1.1. Model for a lane with detectors

In this subsection, we analyse a minor lane equipped with detectors, see lanes corresponding to phases 1 and 2 in Figure 4.2. We consider the evolution of the queue length throughout a cycle in Subsection 4.1.1.1. Then, we find the pgf of the queue length in steady state at the beginning of a cycle in Subsection 4.1.1.2. Below, we present our assumptions on the arrivals and the green-time length.

For a lane with detectors, we assume *free-flow arrivals*, i.e., we assume that arrivals at the lane during different seconds are independent and identically distributed. The analysis can be easily generalised to the case of correlated arrivals, as in Subsection 4.1.2 for a lane without detectors. Let Y denote the random variable of the number of arrivals per second, with pgf $Y(z)$. We assume Bernoulli arrivals, so $Y(z) = \lambda z + 1 - \lambda$, where λ denotes the arrival rate (in vehicles per second) at the lane. The assumption of Bernoulli arrivals is realistic since inter-arrival times of less than one second would lead to dangerously short distances between the vehicles.

Under semi-actuated control, the green time for the lane with detectors terminates as soon as all vehicles are served or a limit, denoted by g , of the green-time length in seconds is reached, whichever occurs first. This time limit prevents too long *waiting times* for the other lanes. Without loss of generality, we assume that $g = d_n + 1$ for some n , i.e., second $B + g - 1 = B + d_n$ is the time of the n^{th} departure. Otherwise, the extra green time after the last delayed departure, i.e., green time at seconds $s + d_n + 1, \dots, s + g - 1$, is wasted. We will refer to n as the *capacity* of the lane.

4.1.1.1. The evolution of the queue length during a cycle

In this subsection, we describe the evolution of the queue length in a cycle. Let X_i be the random variable of the queue length in steady state at second i of a cycle with pgf $X_i(z)$, $i = 0, \dots, c - 1$. Note that $\kappa_s = \mathbb{P}(B = s) = 0$ for $s > c - g$ because a green time ends in the same cycle that it started. Now, we can split the pgf in several parts, by conditioning on B :

$$X_i(z) = \sum_{s=0}^{c-g} \kappa_s X_{i,s}(z),$$

where $X_{i,s}(z) = \mathbb{E}(z^{X_i} | B = s)$. Note that $X_{i,s_1}(z) = X_{i,s_2}(z)$ for $i \leq s_1, s_2$ since B is assumed to be independent of the arrival process and the queue length at the lane at the beginning of the cycle.

In this subsection, we focus on one realisation of B , e.g., $B = s$, and find $X_{i,s}(z)$ for $i = 0, \dots, c$. We break the cycle into the subsequent red (the first s seconds), green (the next g seconds) and red (the rest of the cycle). During each red second, Y vehicles arrive and there are no departures. Therefore,

$$X_{i,s}(z) = X_0(z)(Y(z))^i, \quad \text{for } i = 0, \dots, s. \quad (4.1)$$

During the next g seconds, we either have green time or the green time has terminated due to the service of the last vehicle in the queue. Note that in the latter case, it is possible to have a non-empty queue at second $s+g$ because some vehicles can arrive after the termination of the green time but before second $s+g$. Since the system behaves differently in these two cases, we split the pgf in two parts:

$$X_{i,s}(z) = X_{i,s}^{\text{green}}(z) + X_{i,s}^{\text{red}}(z), \quad \text{for } i = s, \dots, s+g-1,$$

where $X_{i,s}^{\text{green}}(z)$ and $X_{i,s}^{\text{red}}(z)$ denote the parts of the queue-length pgf corresponding to green and red light at second i , respectively:

$$\begin{aligned} X_{i,s}^{\text{green}}(z) &= \mathbb{E}(z^{X_i} \mathbb{1}\{\text{green at time } i\} | B = s), \\ X_{i,s}^{\text{red}}(z) &= \mathbb{E}(z^{X_i} \mathbb{1}\{\text{red at time } i\} | B = s), \end{aligned} \quad \text{for } i = s, \dots, s+d_n, \quad (4.2)$$

where $\mathbb{1}$ is the indicator function. At time $i = s$, we have

$$X_{s,s}^{\text{green}}(z) = X_{s,s}(z) - X_{s,s}(0), \quad X_{s,s}^{\text{red}}(z) = X_{s,s}(0), \quad (4.3)$$

with $X_{s,s}(z)$ specified in (4.1). This means that we have no actual green time in this cycle with probability $X_{s,s}(0)$. Note that by definition, $X_{i,s}^{\text{green}}(0) = 0$, $i = s, \dots, s+g-1$, since the traffic light is switched to red if the queue is empty.

To find $X_{i,s}^{\text{green}}(z)$ and $X_{i,s}^{\text{red}}(z)$ for second i such that $i \neq s+d_k$ for each k , we need to multiply both parts by $Y(z)$ since there are no departures:

$$\begin{aligned} X_{i+1,s}^{\text{green}}(z) &= X_{i,s}^{\text{green}}(z)Y(z), \\ X_{i+1,s}^{\text{red}}(z) &= X_{i,s}^{\text{red}}(z)Y(z), \end{aligned} \quad \text{for } i \neq s+d_k, \forall k. \quad (4.4)$$

For second i such that $i = s+d_k$ for some k , there is a departure, and the traffic light may change to red. Hence, we have

$$\begin{aligned} X_{i+1,s}^{\text{green}}(z) &= \frac{X_{i,s}^{\text{green}}(z)Y(z)}{z} - \bar{w}_{k+1,s}, \\ X_{i+1,s}^{\text{red}}(z) &= X_{i,s}^{\text{red}}(z)Y(z) + \bar{w}_{k+1,s}, \end{aligned} \quad \text{for } i = s+d_k, \quad (4.5)$$

where

$$\bar{w}_{k+1,s} = \mathbb{P}(X_{s+d_k} > 0 \text{ and } X_{s+d_k+1} = 0 | B = s)$$

is the probability that the queue empties after the k^{th} departure, which also means that the queue is empty at the beginning of the potential $(k+1)^{\text{st}}$ departure and the red time just started.

Note that for second $i = s + d_n$, the left-hand sides of (4.5) are not defined in (4.2). However, the sum of the right-hand sides yields $X_{i+1,s}(z)$ because the queue length at second $i+1 = s + g$ is not affected by the fact that the green-time limit is reached. Note that in $X_{s+g,s}(z)$, the probabilities $\bar{w}_{k,s}$ are used only for $k = 1, \dots, n$, where we define $\bar{w}_{1,s}$ as $X_{s,s}^{\text{red}}(1) = X_{s,s}(0)$.

During the remainder of the cycle, the traffic light for this lane is red, and the pgf of the queue length changes in the same way as in (4.1):

$$X_{i,s}(z) = X_{s+g,s}(z)(Y(z))^{i-s-g}, \quad \text{for } i = s + g, \dots, c, \quad (4.6)$$

where $X_{c,s}(z)$ is the conditional pgf of the queue length at the beginning of a new cycle provided that the green time during the previous cycle began at second s .

4.1.1.2. The pgf of the queue length at the beginning of a cycle

In this subsection, we find the pgf $X_0(z)$ of the queue length at the beginning of a cycle. For this, we obtain $X_c(z)$, which is the pgf of the queue length in steady state at the beginning of the next cycle. In steady state, the distribution of X_c coincides with that of X_0 , i.e., $X_c(z) = X_0(z)$.

Combining (4.3), (4.4) and (4.5) gives

$$X_{s+g,s}(z) = \frac{X_{s,s}(z)(Y(z))^g}{z^n} + \sum_{k=1}^n \bar{w}_{k,s} \left(1 - \frac{1}{z^{n-k+1}}\right) (Y(z))^{g-d_{k-1}-1}, \quad (4.7)$$

where, for convenience, $d_0 = -1$. Using (4.1), (4.6) and (4.7), we find

$$X_{c,s}(z) = \frac{X_0(z)(Y(z))^c}{z^n} + \sum_{k=1}^n \bar{w}_{k,s} \left(1 - \frac{1}{z^{n-k+1}}\right) (Y(z))^{c-s-d_{k-1}-1}.$$

Summing results for all s yields

$$\begin{aligned} X_c(z) &= \sum_{s=0}^{c-g} \kappa_s X_{c,s}(z) = \frac{X_0(z)(Y(z))^c}{z^n} + \\ &\quad + \sum_{s=0}^{c-g} \kappa_s \sum_{k=1}^n \bar{w}_{k,s} \left(1 - \frac{1}{z^{n-k+1}}\right) (Y(z))^{c-s-d_{k-1}-1}. \end{aligned} \quad (4.8)$$

Using $X_c(z) = X_0(z)$, we get an equation for the function $X_0(z)$ with unknown coefficients $\bar{w}_{k,s}$.

We may express $\bar{w}_{k,s}$ in terms of the probabilities x_j of having j vehicles in the queue at the beginning of a cycle, $j = 0, \dots, n-1$. To do so, let $\bar{a}_{jk,s}$ denote

the conditional probability that the green time terminated after $k - 1$ departures given $X_0 = j$, and $B = s$, i.e.,

$$\begin{aligned}\bar{\alpha}_{jk,s} &= \mathbb{P}(X_{s+d_{k-1}} > 0 \text{ and } X_{s+d_{k-1}+1} = 0 | X_0 = j, B = s) && \text{for } k > 1, \\ \bar{\alpha}_{jk,s} &= \mathbb{P}(X_s = 0 | X_0 = j, B = s) && \text{for } k = 1.\end{aligned}$$

Note that the queue can be empty after the $(k - 1)^{\text{st}}$ departure only if there were not more than $k - 1$ vehicles in the queue at the beginning of a cycle. Hence, from the law of total probability,

$$\bar{w}_{k,s} = \sum_{j=0}^{k-1} \bar{\alpha}_{jk,s} x_j. \quad (4.9)$$

Inserting $X_c(z) = X_0(z)$ and (4.9) in (4.8), we get

$$X_0(z) = \frac{\sum_{j=0}^{n-1} x_j f_j(z)}{z^n - (Y(z))^c}, \quad (4.10)$$

where

$$f_j(z) = \sum_{s=0}^{c-g} \kappa_s \sum_{k=j+1}^n \bar{\alpha}_{jk,s} (z^n - z^{k-1})(Y(z))^{c-s-d_{k-1}-1}. \quad (4.11)$$

From (4.10), we can obtain the pgf at any other second of a cycle, i.e., $X_i(z)$ for $i = 1, \dots, c - 1$. The algorithm of computing probabilities x_j is summarised in Remark 4.5. In Remark 4.6, we suggest a way to include randomness in the service process.

Remark 4.5 (Computation of x_j , $j = 0, \dots, n - 1$). Pgfs of form (4.10) were discussed in Chapter 3, where we proved that the unknown coefficients can be found using contour integrals, see Section 3.2. For Bernoulli arrivals, the functions $f_j(z)/(z - 1)$ are polynomials, and, therefore, no truncation problem arises, see Remark 3.14. Thus, to find the unknowns it is sufficient to compute probabilities $\bar{\alpha}_{jk,s}$ as follows. We consider the possible queue lengths up to second $s + g$ provided that there were j vehicles in the queue at the beginning of a cycle, the traffic light is still green, and $B = s$. This can be done using equations (4.1) - (4.5) starting with z^j instead of $X_0(z)$. There, equations (4.3) and (4.5) can be used to obtain probabilities $\bar{\alpha}_{jk,s}$. The numerical results of Section 4.2 were obtained prior to results of Chapter 3. For this reason, the mentioned in Section 4.2 speed of the model refers to the application of the classical root-finding approach, see Section 1.6.

Remark 4.6 (Drivers' distraction). It is possible to slightly generalise our model by including drivers' distraction as follows. Suppose that the first vehicle in the queue has an extra random delay of P seconds. Thus, the k^{th} delayed departure

occurs at second $B + P + d_k$, $k = 1, 2, \dots$. This delay P can be readily incorporated in (4.10) and (4.11) as follows:

$$X_0(z) = \frac{\sum_{j=0}^{n-1} x_j f_j(z)}{z^n - (Y(z))^c C(z)}, \quad (4.12)$$

$$f_j(z) = \sum_{\xi} v_{\xi} \sum_{s=0}^{c-g} \kappa_s \sum_{k=j+1}^{n_{\xi}} \bar{\alpha}_{jk,s\xi} (z^n - z^{k-1+n-n_{\xi}}) (Y(z))^{c-s-d_{k-1}-1-\xi \mathbf{1}_{\{k \neq 1\}}}, \quad (4.13)$$

where $C(z) = \sum_{\xi} v_{\xi} z^{n-n_{\xi}}$, ξ is a possible value of P , $v_{\xi} = \mathbb{P}(P = \xi)$, $n_{\xi} = \max\{m: \xi + d_m < g\}$, $\bar{\alpha}_{jk,s\xi}$ is the conditional probability that the green time terminated after $k-1$ departures given $X_0 = j$, $B = s$, and $P = \xi$. To see that (4.12) and (4.13) are true, note that for a fixed value of ξ , the derivation of the pgf can be used with values of d_k changed to $d_k + \xi$ (and $d_0 = -1$ as before) and value of n changes to n_{ξ} , see (4.8). The results for different ξ are then summed with probabilities v_{ξ} . Delay P was not included above for notational convenience. In Subsection 4.2.2, we propose a way to estimate the distribution of P using SUMO. It is also possible to include random inter-departure times in the model, see Chapter 6, where we model the service process using a Markov chain.

4.1.2. Model for a lane without detectors

In this subsection, we consider a lane without detectors and derive the pgf of its queue length at the beginning of a cycle. This pgf is more complicated than that for a minor lane because we analyse more general arrivals. Therefore, we will first show that the pgf has a rational form. Subsequently, we will derive an explicit expression. The model of this subsection is used for all lanes under fixed control and for lanes corresponding to phases 3 and 4 in Figure 4.2 under semi-actuated control.

Consider a lane without detectors. The green time can start at a random second due to detectors on the other lanes. We assume that it terminates at a predetermined moment e in the cycle, i.e., e is the first second of the red time after the green time ended.

The arrivals at the lane can be from outside the system or from an upstream intersection. In the latter case, the arrivals are correlated. To maintain tractability of the model, we make the following assumption.

Assumption 4.7 (Arrivals' independence). The arrivals are independent of the state of the queue at the beginning of a cycle. The arrivals during different cycles are independent and identically distributed (with a known distribution). The arrivals from different upstream lanes are independent of each other.

The numerical results in Subsection 4.2.1 show that the effect on the average delay of correlation between arrivals in different cycles is negligible except for very

high load (above 0.9). As before, we assume that there may be not more than one arrival during one second, and that the arrivals from outside the system form a Bernoulli process. However, our model is general, and we can use any arrival process consistent with [Assumption 4.7](#). In [Subsection 4.1.2.1](#), we describe the arrivals using a discrete-time MAP, see [He \(2014\)](#). To find the pgf of the queue length at the beginning of a cycle, we consider the evolution of the queue length during a cycle in [Subsection 4.1.2.2](#). Then, we present the pgf in [Subsection 4.1.2.3](#). As in the case of a lane with detectors, it has a fractional form but with a more complicated numerator, which we derive in [Subsection 4.1.2.4](#).

4.1.2.1. Arrival process

This subsection presents our model of the arrival process to a lane. There are two different types of arrivals: *external arrivals* and *internal arrivals*. External arrivals are *free-flow arrivals* from outside the system. The corresponding lane is called *external lane*. The arrivals to this lane form a Bernoulli process, i.e., each second a vehicle arrives with a certain fixed probability, independently of the previous arrivals, as in [Subsection 4.1.1](#). The internal arrivals originate from an upstream intersection, and the corresponding downstream lane is called *internal lane*. In this case, the arriving vehicles form *platoons*, where a platoon is a group of vehicles that departed from the same upstream lane during the same green period. One platoon consists of two parts that correspond to *delayed* and *free* departures. We unify the analysis of these two cases, by modelling the arrivals as a MAP. In this subsection, we present the required notation and construct the underlying Markov chain for the case of free-flow arrivals. The case of internal arrivals is treated later in [Subsection 4.1.3](#), as it requires the output of the upstream intersection. There, we also take into account the offset between intersections.

We model arrivals using an underlying periodic discrete-time Markov chain L_i , $i = 0, 1, 2, \dots$, with a finite state space. Due to [Assumption 4.7](#), the state of the MAP at second c , when the next cycle starts, is independent of the past, which makes this second a regenerative moment. Hence, we consider only seconds $i = 0, \dots, c - 1$. For each second i of a cycle, let $\mathcal{L}_i$ denote the set of possible states for this second. Each state represents the information that determines the arrivals, e.g., the number of delayed departures at an upstream lane. The precise definition of a state depends on the type of arrivals and is given below for external arrivals and in [Subsection 4.1.3](#) for internal arrivals. For $l \in \mathcal{L}_i$, let σ_i^l denote the probability of being in state l , i.e., $\sigma_i^l = \mathbb{P}(L_i = l)$. The transition probabilities only depend on the second and state and are denoted by $p_i^{ll'} = \mathbb{P}(L_{i+1} = l' | L_i = l)$.

Let Y_i , $i = 0, \dots, c - 1$, be the (random) number of arrivals at second i . Recall that there is at most one arrival per second, which means $Y_i \in \{0, 1\}$ for each i . To model the arrivals, we assign to each state $l \in \mathcal{L}_i$ an arrival rate $\lambda_i^l \in [0, 1]$ such that $\mathbb{P}(Y_i = 1 | L_i = l) = \lambda_i^l = 1 - \mathbb{P}(Y_i = 0 | L_i = l)$ independently of the history. The pgf of arrivals at second i and in state l is denoted by $A_i^l(z)$, i.e., $A_i^l(z) = \lambda_i^l z + 1 - \lambda_i^l$.

For convenience, we also consider the pgf $A_{i,i'}^l(z)$ of arrivals in seconds $i, \dots, i' - 1$ in case of event $L_i = l$. Due to the Markov property, we have

$$A_{i_1,i_2}^l(z) = A_{i_1,i'}^l(z) \sum_{l' \in \mathcal{L}_{i'}} \mathbb{P}(L_{i'} = l' | L_{i_1} = l) A_{i',i_2}^{l'}(z), \quad (4.14)$$

for all seconds $i_1 < i' < i_2$. For $i' = i_2 - 1$, we get $A_{i',i_2}^{l'}(z) = A_{i'}^{l'}(z)$, which means that $A_{i_1,i_2}^l(z)$ can be found recursively using (4.14).

For external arrivals, the MAP is trivial, i.e., there is only one state at each second, say state 0, with $\lambda_i^0 = \lambda$, $\sigma_i^0 = 1$, $p_i^{00} = 1$. Here λ is the (fixed) arrival rate. This MAP generates i.i.d. Bernoulli arrivals with probability λ of an arrival in one second.

Remark 4.8 (Alternative model). We may alternatively model arrivals using the joint distribution of $Y_0, \dots, Y_{c-1}$ (see [Boon and van Leeuwen \(2018\)](#)), but we find it numerically inconvenient to work with multi-dimensional functions. Also representation of arrivals as a MAP makes it possible to visualise the arrivals, thus providing a better understanding of the arrival process, see the examples in [Subsection 4.1.3](#).

4.1.2.2. The evolution of the queue length during a cycle

In this subsection, we consider the evolution of the queue length during a cycle. As before, X_i is the random variable of the queue length in steady state at second i in a cycle with pgf $X_i(z)$, $X_{i,s}(z) = \mathbb{E}(z^{X_i} | B = s)$, and

$$X_i(z) = \sum_{s=0}^{c-1} \kappa_s X_{i,s}(z). \quad (4.15)$$

Due to [Assumption 4.2](#), $X_{i,s_1}(z) = X_{i,s_2}(z)$ for $i \leq s_1, s_2$, as in [Subsection 4.1.1](#). In particular, $X_{0,s}(z) = X_0(z)$ for each s . For convenience, we further split the pgf $X_{i,s}(z)$ in parts corresponding to different states of the MAP of arrivals:

$$X_{i,s}(z) = \sum_{l \in \mathcal{L}_i} \sigma_i^l X_{i,s}^l(z), \quad (4.16)$$

where

$$X_{i,s}^l(z) = \mathbb{E}(z^{X_i} | L_i = l, B = s).$$

We will now consider all possible types of seconds during a cycle: red second, green second with delayed departure and green second without delayed departure. Consider a red second i . During this second there may be an arrival, and there are no departures. Therefore, the pgf of the queue length changes as

$$X_{i+1,s}^{l'}(z) = \sum_{l \in \mathcal{L}_i} \mathbb{P}(L_i = l | L_{i+1} = l') X_{i,s}^l(z) A_i^l(z). \quad (4.17)$$

During a green time, it is possible that the queue is empty and the arrivals proceed without delay. Let $w_{i,s}^l$ denote the conditional probability of an empty queue at second i given $L_i = l$ and $B = s$, i.e.,

$$w_{i,s}^l = \mathbb{P}(X_{i,s} = 0 | L_i = l) = X_{i,s}^l(0).$$

Note that the total probability of an empty queue, i.e., $\sum_{l \in \mathcal{L}_i} \sigma_i^l w_{i,s}^l$, does not change between seconds $s + d_k + 1$ and $s + d_{k+1}$ due to free departures.

The delayed departures occur at seconds $s + d_k$, $k = 1, \dots, n_s$. First, consider a green second i without delayed departures, i.e., $s + d_{k-1} < i < s + d_k$ for some k . Note that arrivals join the queue only if the queue is not empty since otherwise arrivals proceed without delay and the queue remains empty. Thus, we get

$$X_{i+1,s}'(z) = \sum_{l \in \mathcal{L}_i} \mathbb{P}(L_i = l | L_{i+1} = l') \left[(X_{i,s}^l(z) - w_{i,s}^l) A_i^l(z) + w_{i,s}^l \right]. \quad (4.18)$$

Now, consider the second of the potential k^{th} departure, i.e., $i = s + d_k$. After a possible arrival, the pgf changes as in the right-hand side of (4.18). Afterwards, there is a departure if the queue is not empty. Therefore, the pgf changes as follows:

$$X_{i+1,s}'(z) = \sum_{l \in \mathcal{L}_i} \mathbb{P}(L_i = l | L_{i+1} = l') \left[(X_{i,s}^l(z) - w_{i,s}^l) \frac{A_i^l(z)}{z} + w_{i,s}^l \right]. \quad (4.19)$$

Equations (4.17), (4.18) and (4.19) describe all possible changes of the queue-length pgf during a cycle. From these equations, we find an equation for $X_0(z)$ in the next Subsection 4.1.2.3.

4.1.2.3. The pgf of the queue length at the beginning of a cycle

In this subsection, we prove that the pgf $X_0(z)$ of the queue length at the beginning of a cycle, similar to Subsection 4.1.1, can be represented as a fraction:

$$X_0(z) = \frac{\sum_{j=0}^{n-1} x_j f_j(z)}{z^n - A(z)C(z)}, \quad (4.20)$$

where n is the maximum number of delayed departures per cycle, x_j , $j = 0, \dots, n-1$, are probabilities of having j vehicles in the queue at the beginning of a cycle, $A(z) = \sum_{l \in \mathcal{L}_0} \sigma_0^l A_{0,c}^l(z)$ is the pgf of all arrivals during the cycle, $f_j(z)$ and $C(z)$ are analytic functions. More precisely,

$$C(z) = \sum_{s=0}^{c-1} \kappa_s z^{n-n_s}, \quad (4.21)$$

where n_s is the maximum number of delayed departures per cycle in case $B = s$, which means that $n_s = \max\{k : s + d_k < c\}$, and $n = \max_s n_s$. The function $C(z)$

can be seen as a pgf of the capacity loss due to randomness of the green-time beginning since the *capacity* of the lane is given by $\sum_{s=0}^{c-1} \kappa_s n_s = n - C'(1)$.

To see that (4.20) holds, observe that in equations (4.17), (4.18) and (4.19) the pgf of the queue length at the previous second is always multiplied by arrivals during that second and then maybe divided by z in the case of a departure. Therefore, after summing up the components $\sigma_c^{l'} X_{c,s}^{l'}(z)$ for all $l' \in \mathcal{L}_0 = \mathcal{L}_c$, we get that $X_{0,s}(z)$ is multiplied by the pgf of all arrivals $A(z)$ and divided by z^{n_s} . The equality $X_{0,s}(z) = X_0(z)$ and the definition (4.21) of $C(z)$ imply that $X_0(z)$ is multiplied by $A(z)C(z)/z^n$. Hence, we only need to show that the numerator in (4.20) is correct. Note that in the numerator there should be a sum of probabilities $w_{i,s}^l$ multiplied by some polynomials. As before, the probability $w_{i,s}^l$ can be written in terms of x_j , $j = 0, \dots, n-1$. Therefore, $X_0(z)$ has the fractional form (4.20). We continue the analysis of $X_0(z)$ by determining the numerator, see the following Subsection 4.1.2.4.

4.1.2.4. The numerator of the pgf

In this subsection, we derive the numerator of the pgf in (4.20). Consider the evolution of the queue length between potential $(k-1)^{\text{st}}$ and k^{th} delayed departures, i.e., during seconds $s+d_{k-1}+1, \dots, s+d_k$. From our assumption on free departures and the definition of $A_{i,i'}^l(z)$, it follows that for seconds $i = s+d_{k-1}+1$ and any $i' \leq s+d_k$, the pgf of the queue length changes as follows:

$$X_{i',s}(z) = \sum_{l \in \mathcal{L}_i} \sigma_i^l \left((X_{i,s}^l(z) - \bar{w}_{k,s}^l) A_{i,i'}^l(z) + \bar{w}_{k,s}^l \right), \quad (4.22)$$

where $\bar{w}_{k,s}^l = w_{s+d_{k-1}+1,s}^l$. After the k^{th} departure occurs at second $s+d_k$, i.e., for $i' = s+d_k+1$, the resulting pgf is given by

$$X_{i',s}(z) = \sum_{l \in \mathcal{L}_i} \sigma_i^l \left(\frac{X_{i,s}^l(z) A_{i,i'}^l(z)}{z} + \bar{w}_{k,s}^l \left(1 - \frac{A_{i,i'}^l(z)}{z} \right) \right). \quad (4.23)$$

We may now relate $X_{c,s}(z)$ to $X_0(z)$. By analogy with the derivation in Subsection 4.1.1.2, observe that after the k^{th} departure $\bar{w}_{k,s}^l$ from equation (4.23), as a part of the queue-length pgf, will be multiplied by the pgf of (possible) arrivals and divided by z^{n_s-k} since n_s-k is the number of potential delayed departures subsequent to the k^{th} delayed departure, see equations (4.17), (4.18) and (4.19) for $i \geq s+d_k+1$. Thus, for a fixed value of s , we get

$$\begin{aligned} X_{c,s}(z) &= X_0(z) A(z) \frac{z^{n-n_s}}{z^n} + \\ &+ \sum_{k=1}^{n_s+1} \sum_{l \in \mathcal{L}_{k,s}} \bar{\sigma}_{k,s}^l \bar{w}_{k,s}^l \left(g_{k,s}^l(z) z^{1\{k \neq n_s+1\}} - f_{k,s}^l(z) \right) z^{k-n_s-1}, \end{aligned} \quad (4.24)$$

where, for simplicity, $\mathcal{L}_{s+d_{k-1}+1}$ is denoted by $\bar{\mathcal{L}}_{k,s}$ and $\sigma_{s+d_{k-1}+1}^l$ by $\bar{\sigma}_{k,s}^l$;

$$g_{k,s}^l(z) = \sum_{l' \in \bar{\mathcal{L}}_{k,s}} \mathbb{P}(L_{i'} = l' | L_{s+d_{k-1}+1} = l) A_{i',c}^{l'}(z),$$

with $i' = s + d_k + 1$ if $k \leq n_s + 1$, and $i' = e$ if $k = n_s + 1$; and $f_{k,s}^l(z) = A_{s+d_{k-1}+1,c}^l(z)$, see Subsection 4.1.2.1 for definition of $A_{i_1,i_2}^l(z)$. The functions $g_{k,s}^l(z)$ and $f_{k,s}^l(z)$ are the conditional pgfs of arrivals during the rest of the cycle starting from second $s + d_k + 1$ (or e if $k = n_s + 1$) and $s + d_{k-1} + 1$, respectively, given $L_{s+d_{k-1}+1} = l$ and $B = s$. Note that if $k = n_s + 1$, i.e., if red time starts before the potential k^{th} delayed departure can happen, the pgf changes first as in (4.22) and then as in (4.17). Thus, no division by z is needed, which explains the indicator function in term $z^{\mathbb{1}\{k \neq n_s+1\}}$ in (4.24). Summing (4.24) for all values of s and using $X_c(z) = X_0(z)$, we find

$$X_0(z) = \frac{\sum_{s=0}^{c-1} \kappa_s \sum_{k=1}^{n_s+1} \sum_{l \in \bar{\mathcal{L}}_{k,s}} \bar{\sigma}_{k,s}^l \bar{w}_{k,s}^l \left(g_{k,s}^l(z) z^{\mathbb{1}\{k \neq n_s+1\}} - f_{k,s}^l(z) \right) z^{n-n_s+k-1}}{z^n - A(z)C(z)}.$$

To prove that there exist functions $f_j(z)$ such that the numerator of (4.20) can be written as $\sum_{j=0}^{n-1} x_j f_j(z)$, we need to express $\bar{w}_{k,s}^l$ in terms of probabilities x_j of having j vehicles in the queue at the beginning of a cycle for $j = 0, \dots, n-1$. Let $\bar{\alpha}_{jk,s}^l$ denote the probability of having an empty queue at second $s + d_{k-1} + 1$ given events $L_{s+d_{k-1}+1} = l$, $X_0 = j$ and $B = s$. Then,

$$\bar{w}_{k,s}^l = \sum_{j=0}^{k-1} x_j \bar{\alpha}_{jk,s}^l. \quad (4.25)$$

Observe that $\bar{\alpha}_{jk,s}^l = 0$ for $j > n_s$. Interchanging the summations over j and k , we can rewrite $X_0(z)$ as

$$X_0(z) = \frac{\sum_{j=0}^n x_j \tilde{f}_j(z)}{z^n - A(z)C(z)}, \quad (4.26)$$

where

$$\tilde{f}_j(z) = \sum_{s=0}^{c-1} \kappa_s \sum_{k=j+1}^{n_s+1} \sum_{l \in \bar{\mathcal{L}}_{k,s}} \bar{\sigma}_{k,s}^l \bar{\alpha}_{jk,s}^l \left(g_{k,s}^l(z) z^{\mathbb{1}\{k \neq n_s+1\}} - f_{k,s}^l(z) \right) z^{n+k-n_s-1}.$$

As the final step, let us consider the term for $j = n$ in the numerator of (4.26). For $k = n_s + 1$, i.e., when the green time is longer than $d_{n_s} + 1$ seconds, but shorter than $d_{n_s+1} + 1$ seconds, the probability $\bar{w}_{k,s}^l$ depends also on x_{n_s} . If $n_s = n = \max_{s'} n_{s'}$, this gives x_n in (4.26). Since in (4.20) variables x_j are only used for $j = 0, \dots, n-1$, we need an extra equation to find x_n . To this end, observe that by summing $\bar{\sigma}_{n_s+1,s}^l \bar{w}_{n_s+1,s}^l g_{n_s+1,s}^l(0)$ for all s and l we obtain x_0 . Thus, x_n may be expressed in terms of $x_0, \dots, x_{n-1}$: $x_n = \sum_{j=0}^{n-1} \zeta_j x_j$ for a suitable choice of ζ_j , $j = 0, \dots, n-1$. Let $f_j(z) = \tilde{f}_j(z) + \zeta_j \tilde{f}_n(z)$, then $\sum_{j=0}^n x_j \tilde{f}_j(z) = \sum_{j=0}^{n-1} x_j f_j(z)$, which concludes the proof of (4.20).

Remark 4.9 (Probabilities x_j and extra delay P). The probabilities x_j can be found similar to the case of a lane with detectors, see [Remark 4.5](#). The coefficients $\bar{a}_{jk,s}^l$ are computed by considering the evolution of the queue length (see equations (4.15) - (4.19)) starting with j vehicles at the beginning of a cycle. Finally, the extra delay P for [delayed](#) departures, see [Remark 4.6](#), can be added to the model by considering

$$C(z) = \sum_{\xi} v_{\xi} \sum_{s=0}^{c-1} \kappa_s z^{n-n_{s,\xi}},$$

and

$$\tilde{f}_j(z) = \sum_{\xi} v_{\xi} \sum_{s=0}^{c-1} \kappa_s \sum_{k=j+1}^{n_{s,\xi}+1} \sum_{l \in \bar{\mathcal{L}}_{k,s}} \bar{\sigma}_{k,s}^l \bar{a}_{jk,s\xi}^l \left(g_{k,s}^l(z) z^{\mathbb{1}_{\{k \neq n_{s,\xi}+1\}}} - f_{k,s}^l(z) \right) z^{n+k-n_{s,\xi}-1},$$

where $n_{s,\xi} = \max\{m: s + \xi + d_m < e\}$, $\bar{a}_{jk,s\xi}^l$ is the probability of having an empty queue at second $s + d_{k-1} + \xi + 1$ (or s if $k = 1$) given $L_{s+d_{k-1}+\xi+1} = l$ (or $L_s = l$ for $k = 1$), $X_0 = j$, $B = s$ and $P = \xi$.

4.1.3. MAP for an internal lane

In this subsection, we construct the [MAP](#) for an [internal lane](#). As building blocks, we use the inputs generated by each of the upstream lanes. First, we consider arrivals from a lane with detectors. Then, we describe the differences compared to arrivals from a lane without detectors. In both cases, the considered upstream lanes are [external lanes](#), and, therefore, they have [free-flow arrivals](#). In [Chapter 5](#), we consider a general case, where the upstream lanes may be internal lanes as well. Finally, we show how to build the MAP for a downstream lane by combining the obtained arrival sub-processes. Below, we give the required notation.

Let the lanes at the upstream intersection be numbered by $u = 1, \dots, N$, according to their order in the green-time schedule of the upstream intersection. For lane u , let γ_u denote the fraction of vehicles from lane u that turn to the considered downstream lane. Lane $u + 1$ for $u = N$ means lane 1, lane $u - 1$ for $u = 1$ means lane N .

4.1.3.1. The input from a lane with detectors

In this subsection, we consider the input from lane u that has detectors. For each beginning of the green time, s , and each number of delayed departures, m , during the green time, we describe the corresponding part of MAP for the downstream lane.

Consider the departures of the delayed vehicles from lane u . The k^{th} delayed departure occurs at second $s + d_k$. Note that the delayed vehicle needs to accelerate to reach the [free-flow speed](#). Let $a_k - d_k$ be the time lost due to acceleration. Then, the k^{th} [delayed arrival](#) from lane u , i.e., arrival that corresponds to the k^{th} delayed

Figure 4.4: Parts of MAP corresponding to arrivals from upstream lane u with detectors. For each number $m = 0, 1, 2$ of delayed departures from upstream lane, the arrivals are different. The k^{th} delayed vehicle arrives at time a_k , $k = 1, \dots, m$. The states of the MAP are represented as circles, where dark orange circles are for the states with arrival rate γ_u , and white circles correspond to arrival rate 0. The arrows show the transitions.

departure, arrives at the downstream lane at second $s + t + a_k$ of the cycle at the upstream intersection, where t is the travel time between intersections at the free-flow speed. Note that the order of arriving vehicles corresponds to the order of departing vehicles, and, therefore, $a_1 < \dots < a_m$. Thus, for a fixed value of $m > 0$, the arrivals from lane u can be represented by a part of MAP consisting of $a_m + 1$ states with one state for each second. The arrival rate is equal to γ_u for seconds a_k , $k = 1, \dots, m$, and is 0 otherwise. The probability of such a part is $\kappa_s \cdot \bar{w}_{m,s}$. Note that if $m = 0$, there is no input for the downstream lane from lane u . In Figure 4.4, we give the inputs for $m = 0, 1, 2$.

4.1.3.2. The input from a lane without detectors

In this subsection we consider upstream lane u that has no detectors. In this case, the delayed arrivals to the downstream lane may be followed by *free arrivals*, which correspond to *free* departures of lane u . As before, we fix the beginning second of the green time, s , and the number of delayed departures, m .

Since lane u is an external lane, there is only state 0 for each second in the MAP of lane u , see Subsection 4.1.2.1. With probability $\bar{w}_{m+1,s}^0 - \bar{w}_{m,s}^0$ there is a delayed departure at second $s + d_m$ and the queue empties, where for convenience $\bar{w}_{0,s}^0 = 0$. Then, there are $e - s - d_m - 1$ possible free departures, where e is the second when the red time starts. Note that the free-departing vehicles can still encounter extra delay. Indeed, they can't be in the input earlier than the m^{th} delayed arrival. Thus, they lose $a_m - d_m$ seconds. Therefore, the corresponding part of MAP for the downstream lane consists of $a_m + e - s - d_m$ states with one state for each second. The arrival rate is γ_u for seconds a_k , $k = 1, \dots, m$, $\gamma_u \lambda_u$ for seconds $a_m + 1, \dots, a_m + e - s - d_m - 1$ and 0 otherwise, where λ_u is the arrival rate at lane u . If the queue at lane u does not empty during the cycle, which happens with probability $1 - \bar{w}_{n_s,s}^0$, then there are n_s delayed arrivals at the downstream lane without any free arrivals.

Figure 4.5: Parts of MAP corresponding to arrivals from an upstream lane without detectors. For each number of delayed departures $m = 0, 1, 2$, the arrivals are different. The k^{th} delayed arrival is given at time a_k . The free arrivals after a delayed arrival are also postponed. The input to the upstream lane is a MAP that generates free-flow arrivals. The colour of the states represents the arrival rate, with darker colour corresponding to higher rates. The arrows from the states represent the transitions with bold grey arrows for possible transitions to the states generated by other upstream lanes.

In this way, we get several possible inputs for the downstream lane that correspond to different values of s and m , where the length of each input depends on the number of delayed departures. However, comparing to the case of a lane with detectors, the green time of the next upstream lane, i.e., lane $u + 1$, is independent of m . In Figure 4.5, we give examples of possible inputs.

Note that, unlike other queueing models, a vehicle experiences an *acceleration delay* after leaving the queue. In the numerical results, we are interested in the average (*total*) *delay* of a vehicle, which we define as the average travel time minus the travel time under the free-flow speed. Thus, the average delay is a sum of this acceleration delay and the *waiting time*, which we obtain using the average queue length and Little's law, see Little (1961).

4.1.3.3. Combining the inputs from upstream lanes

In this subsection, we construct the input to a downstream lane, combining the sub-processes described in Subsections 4.1.3.1 and 4.1.3.2, see an example in Figure 4.6. We give definitions of the states, transitions and arrival rates of the resulting MAP.

Let s_u be the beginning of the green time at lane u of an upstream intersection. For convenience, $s_{N+1} = s_1 + c$, where c is the cycle time. Depending on the length of the input generated by lane u , the arrivals from lane u are followed either directly by arrivals from lane $u + 1$, or by *end states*, which mean that the arrivals from

Figure 4.6: An example of combining the inputs from upstream lanes in one input. The upstream intersection has two lanes: lane 1 with detectors and lane 2 without detectors. The green-time limit of 2 seconds for lane 1 allows zero (left side) or one departure (right side). In the first case, lane 2 has 6 seconds of green time with at most 2 delayed departures. In the second case, lane 2 has 4 seconds of green time with not more than one delayed departure, possibly followed by free departures. The free arrivals from lane 2 can lose 4 extra seconds due to acceleration of the departing from lane 1 vehicle. The acquired combinations of the inputs are further combined in one input chain, using values of the travel time, t , and the offset, o . The transitions from the last second to second 0 becomes independent of the state. The colour of the states represents the arrival rate. The darker the colour the higher the rate. The arrows from the states represent possible transitions. The grey boxes embrace the end states that represent the event that the arrivals from the corresponding lanes have ended.

lane u finished and the arrivals from lane $u + 1$ have not yet started. Observe that the inputs from different lanes can be dependent: arrivals from lane $u + 1$ may be delayed due to an accelerating vehicle from lane u .

Each state of the MAP is a pair of second i and description l . For end states, $l = (u, s_u)$, and, for other states, $l = (u, m, \delta, s_u)$, where u is the index of an upstream lane, m is the number of delayed departures from lane u , δ is a parameter of shift caused by the acceleration of the vehicles departing from lane $u - 1$. Parameter δ is equal to zero if $m > 0$, as delayed arrivals from lane u cannot be further delayed by accelerating vehicles from another lane.

Let o denote the *offset* and let $\beta_{m,s,u}$ be the probability of having m delayed departures from upstream lane u in case of the green-time beginning s for lane u . For a lane without detectors, this probability is given by $\beta_{m,s,u} = \bar{w}_{m+1,s,u}^0 - \bar{w}_{m,s,u}^0$ if $s + d_m + 1 < e_u$ and by $\beta_{m,s,u} = 1 - \bar{w}_{m,s,u}^0$ if $s + d_m + 1 \geq e_u$, where e_u is the beginning of the red time at lane u . For the lane with detectors, $\beta_{m,s,u} = \bar{w}_{m,s,u}$

if $m < n_{s_u, u}$, and $\beta_{m, s_u, u} = 1 - \sum_{m'=0}^{m-1} \bar{w}_{m', s_u, u}$ if $m = n_{s_u, u}$. Note that if there are m delayed departures from lane u , then the free departures from lane $u + 1$ can be delayed by $\max\{a_m - d_m - y_u, 0\}$ seconds (due to acceleration of departures from lane u), where y_u is the switch time between lanes u and $u + 1$. Let $\theta_{\delta, s_u, u}$ be the conditional probability that the free departures from lane u are delayed by δ seconds given $B_u = s$. This probability is calculated by considering all cases when $B_u = s$, all possible beginnings $B_{u-1} = s'$ and numbers of delayed departures m from lane $u - 1$ such that $\max\{a_m - d_m - y_{u-1}, 0\} = \delta$.

The MAP of the downstream lane is constructed as follows. For simplicity, we consider the case that the offset coincides with the travel time, i.e., $o = t$. Otherwise, the value of $o - t$, should be added at each second considered. For each possible combination $l = (u, m, \delta, s_u)$, state (i, l) is added at seconds $i = s_u, \dots, e_u + (a_m - d_m)$ if $m > 0$ and lane u has no detectors, at seconds $i = s_u, \dots, s_u + a_m$ if $m > 0$ and lane u has detectors and at seconds $i = s_u + \delta, \dots, e_u + \delta$ if $m = 0$. The probability of state (i, l) depends on second i :

$$\begin{aligned} \sigma_i^l &= \kappa_{s_u} \beta_{m, s_u, u} \left(1 - \sum_{\delta'=i-s_u+1}^{\infty} \theta_{\delta', s_u, u} \right), & \text{if } m > 0, \\ \sigma_i^l &= \kappa_{s_u} \beta_{m, s_u, u} \theta_{\delta, s_u, u}, & \text{if } m = 0. \end{aligned}$$

Here, $\kappa_{s_u} \beta_{m, s_u, u}$ is the probability that the green time at lane u starts at second s_u , and there are m departures. If $m > 0$, then the state is added if the arrivals were not postponed more than $i - s_u$ seconds, i.e., with probability $1 - \sum_{\delta'=i-s_u+1}^{\infty} \theta_{\delta', s_u, u}$. Otherwise, there are still arrivals from lane $u - 1$. Note that the summation in the first equation is actually finite since the extra delay is small, e.g., in our numerical examples, $\theta_{\delta', s_u, u} = 0$ for values $\delta' > 4$. If $m = 0$, then we consider only the case when the arrivals were postponed by δ seconds. The arrival rates are given by the following formulas:

$$\begin{aligned} \lambda_i^l &= 0, & \text{if } m > 0, i < s_u + a_m, i \neq s_u + a_k, \forall k < m, \\ \lambda_i^l &= \gamma_u, & \text{if } m > 0, i = s_u + a_k, k \leq m, \\ \lambda_i^l &= \gamma_u \lambda_u, & \text{if } m > 0, i > s_u + a_m \text{ or } m = 0, \end{aligned}$$

where $l = (u, m, \delta, s_u)$.

Finally, we give the transition probabilities. In most of the cases, the only transition from state (i, l) is to state $(i + 1, l)$ with probability 1. At the last second of arrival from lane u , i.e., $e_u + (a_m - d_m)$, $s_u + a_m$ or $e_u + \delta$, the transition depends on whether the arrivals from the next lane ($u + 1$) can happen during the next second. If not, i.e., $i < s_{u+1} - 1$, there is only one transition to the end state (i, l') , where $l' = (u, s_u)$, see grey boxes in Figure 4.6. The probability of the end state is calculated as κ_{s_u} minus the probability of all other states added to this second in case $B_u = s_u$. The arrival rate of all end states is equal to 0. If the arrivals from the next lane have already started, i.e., $i \geq s_{u+1} - 1$, then the transitions are as

follows:

$$p_i^{ll'} = \beta_{m', s_{u+1}, u+1}, \quad (4.27)$$

where $l' = (u+1, m', 0, s_{u+1})$ and $m' > 0$ or $l' = (u+1, 0, \delta', s_{u+1})$ and $\delta' = \max\{a_m - d_m - y_u, 0\}$. The transition of the end state are constructed in the same way. It is either $p_i^{ll} = 1$ if $i < s_{u+1} - 1$ or defined as in (4.27) if $i = s_{u+1} - 1$ with only difference that $\delta' = 0$. The only transitions that are not given by the above formulas are the transitions in the end of the cycle. These transitions are independent of the state, i.e., $p_{c-1}^{ll'} = \sigma_0^{l'}$ for $l' \in \mathcal{L}_0$, see [Assumption 4.7](#).

The above construction is simple, but contains many states with the same arrival rate. It is clear that some states can be merged. For example, states (i, l) and (i, l') , where $l = (u, m, 0, s_u)$ and $l' = (u, m', 0, s_u)$, generate the same arrivals up to second $s_u + a_m$, where $0 < m < m'$. These states can be merged, thus reducing the total number of states but also changing the meaning of the state.

4.2. Numerical results

In this section, we present several numerical results. In [Subsection 4.2.1](#), we measure the impact of our independence [Assumptions 4.2](#) and [4.7](#). The departure and arrival moments are determined in [Subsection 4.2.2](#), using SUMO. In [Subsection 4.2.3](#), we compare results of our model and SUMO. Finally, in [Subsection 4.2.4](#), we analyse the performance of [fixed](#) and [semi-actuated](#) control.

4.2.1. Independence assumptions

In this subsection, we discuss our independence assumptions. In [Assumption 4.2](#), we assume that the beginning moments of the green times are independent of each other. In reality, they depend on the queue lengths at the minor lanes. In [Assumption 4.7](#), the arrivals during different cycles are assumed to be independent of each other. The last assumption is justified in the case of low load, where [load](#) is defined as a ratio of the average number of arrivals per cycle and the [capacity](#) of the lane under [fixed](#) control. Consider a lane without detectors. If the load is low, the queue is empty at the end of the green time during most of the cycles. Therefore, the number of departures during the next cycle is determined only by the arrivals during the red time and, hence, is independent of the departures in the current cycle. For a higher load, there is a stronger dependency between the arrivals during subsequent cycles. We also assume that arrivals from different lanes are independent from each other. In [Subsection 4.2.1.1](#), we compare simulations of our model with and without independence assumptions. In [Subsection 4.2.1.2](#), we evaluate the autocorrelation of the number of delayed departures and the beginning of the green time.

Figure 4.7: The structure of the simulated model. Queues 0 and 1 are upstream lanes and queue 2 is a downstream lane. Queues 1 and 2 are on the main road and queue 0 is a minor road. For a given load ρ , the arrival rates to queues 0 and 1 are $4\rho/60$ and $8\rho/60$ vehicles per second, respectively. The cycle length is 60 seconds. The capacities of the queues 0, 1 and 2 are 4, 8 and 8 (delayed) vehicles per cycle, respectively. With probabilities 0.8 and 0.6 the departing vehicles from queues 0 and 1 join queue 2.

4.2.1.1. Simulation of dependent and independent models

To test our assumptions, we have simulated our model with and without [Assumptions 4.2](#) and [4.7](#). For simplicity, we consider 3 lanes: two upstream lanes (minor 0 and major 1) and one downstream lane (2), see [Figure 4.7](#). For an upstream intersection, we use both semi-actuated and fixed controls and, for a downstream lane, only fixed control. First, we have simulated the joint behaviour of all three lanes represented by queues 0, 1 and 2, respectively. During this simulation, we record the distribution of [platoons](#) from the upstream lanes and, in the case of semi-actuated control, the probability of terminating the green time early for queue 0. Second, this information is used to generate the arrivals to queue 2 and the beginning of the green time for queue 1 for each cycle, independently of the previous cycle. In this way, we simulate our model with independence assumptions.

We have considered several scenarios for green times and offsets between intersections. The results of different scenarios are similar. Therefore, we restrict our description to the scenario given below. The green time for minor and major lanes is 12 and 22 seconds, with resulting capacities of 4 and 8 vehicles per cycle, respectively, in correspondence with the departure moments d_k , $k = 1, 2, \dots$, found in [Subsection 4.2.2](#). We set the arrival rates and routing probabilities such that the load, ρ , for each lane is the same, see [Figure 4.7](#). In this way, the load is the only parameter of the system, and the lanes have a similar impact on the performance of the entire system. The offset between green times for queue 1 and 2 is 10 seconds and the travel time between the queues is 6 seconds.

As can be seen from [Figure 4.8](#), both under fixed control and semi-actuated control, for loads up to 0.8, the independence assumptions on the arrivals, [Assump-](#)

a) fixed control

b) semi-actuated control

Figure 4.8: The comparison between dependent and independent queues under a) fixed and b) semi-actuated control (for queues 0 and 1). The (maximum) green time for queue 0 is 12 seconds; the (minimum) green time for queue 1 is 22 seconds; the green time for queue 2 is 22 seconds.

tion 4.7, are justified, see the graphs for queue 2. For $\rho = 0.9$, Assumption 4.7 seems to result in underestimation of the average waiting time. Figure 4.8 shows the 95% confidence intervals, which are rather wide under high load. This is in agreement with observations in queueing theory that randomness of arrival and service processes has a higher impact on the variability of the waiting time under high load, see, e.g., Kingman (1961) for the Kingman approximation of the mean waiting time in the $G/G/1$ queue. The graphs for queue 1 in Figure 4.8b show

that [Assumption 4.2](#) seems justified for all loads ρ considered in our simulation.

In [Figure 4.8a](#), the waiting time for queue 2 is a bit decreasing between loads 0.3 and 0.5. This happens because the offset between green times of queues 1 and 2 is a bit longer than the time needed by a [free](#) departing vehicle to arrive to queue 2, but shorter than the time needed by a delayed vehicle. Therefore, for a low load, there are more free departures from queue 1, and these departures need to wait in queue 2, while for a little higher load, the first departure is delayed and during this delay some vehicles from queue 0 can depart, which makes the average waiting time less. In the case of semi-actuated control, the decrease is stronger because we use fixed control for queue 2, see [Figure 4.8b](#). This effect means that the optimal offset depends on the arrival rates of the upstream intersection.

We see that independence between green times gives a very good approximation for all loads, while assuming independence between arrivals gives reasonable results only for loads up to 0.8. Hence, the correlation between service times has less impact than the correlation between arrival times. Similar results were obtained for an $M/M/1$ queue with autocorrelation, see [Livny et al. \(1993\)](#). In the following subsection, we analyse these correlations.

4.2.1.2. Autocorrelation

For both independence assumptions, we study the autocorrelation of the involved random variables, namely the length G of a green time and the number D of [delayed](#) departures in a cycle. In [Figure 4.9](#), for $X \in \{G, D\}$, we plot the autocorrelation

$$R_X(\tau) = \frac{\mathbb{E}[(X_l - \mu_X)(X_{l+\tau} - \mu_X)]}{\sigma_X^2}$$

depending on τ and $R_X(1)$ as a function of the load, where index l is the cycle number; μ_X and σ_X^2 are the expectation and the variance of X , respectively.

Observe that the quantities are mostly positively autocorrelated. However, for a low load, the number of delayed departures from queue 0 and, therefore, the green time beginning for queue 1 have a negative coefficient of autocorrelation. This result can be explained as follows. Under semi-actuated control, for any load, the shorter the green time for the minor lane in the current cycle, the higher the probability that there will be many vehicles in the queue at the beginning of the next green time. Thus, the probability that there will be a longer green time is also higher. For a high load, if the time is fully used in one cycle, it is most probably also fully used during the next cycle. Hence, the green time lengths are negatively-correlated for a low load, but positively-correlated for a high load.

Concerning the size of the platoon, we observe that autocorrelation is almost 0 for low load, which is the basis of our [Assumption 4.7](#). For a higher load, if the green time was fully used by delayed vehicles, the probability that there will be many departures during the next cycle is higher than on average. This leads to long series of many arrivals to the downstream lane, which increases the queue length, see [Figure 4.8](#).

Figure 4.9: The autocorrelation of a), b), c), d) the number of delayed departures and e), f) the beginning of the green time. Subfigures a) and b) correspond to fixed control, and subfigures c) - f) correspond to semi-actuated control. Subfigures a), c) and e) show the autocorrelation for load 0.9 as a function of the time lag (τ). Subfigures b), d) and f) show the autocorrelation for time lag $\tau = 1$ as a function of load.

Departure moments														
d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
1	5	8	11	14	16	19	21	23	25	27	29	31	33	35
Arrival moments														
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
11	13	15	17	19	21	23	24	26	28	30	32	34	35	37

Table 4.1: The departure and arrival moments.

4.2.2. Determining the arrival and departure moments

Our numerical results require values for the departure moments d_k and the arrival moments a_k . For this purpose, we use SUMO with default values for all parameters except the acceleration that we set to 0.8 m/sec² similar to the examples in SUMO documentation. We simulate a long queue and a long cycle with 38 seconds of green time. For each cycle, we record the trajectories of the departing vehicles depending on their departure number, see Figure 4.10. Note that numbering of departures restarts from 1 at the beginning of each green period. We see that the distribution of departure moments is very close to deterministic. Since we consider a discrete model, we round each departure moment to an integer second. Then, the supports of the distributions consist of 2-4 values, see Figure 4.11. The only exception is the last departure (departure 16, see Figure 4.10). The bifurcating nature of the trajectories of this last departure is caused by vehicles that try but cannot stop after the traffic light for this lane switches to yellow. These vehicles depart from the lane with a lower speed. This behaviour is ignored in our model since it is restricted to a small fraction of arriving vehicles and occurs only for long queues.

We approximate the extra delay of vehicles depicted in Figure 4.11 by a common random variable P and add this value to the departure moments d_k of delayed vehicles in our model. As a result, the delayed vehicles depart at seconds $B + P + d_k$ instead of $B + d_k$, where B is the random beginning of the green time, and P is the added randomness. The value of P is randomly chosen in the beginning of a cycle and used for each delayed departure, see also Remark 4.6.

We find values for d_k and a_k in the following way. For a fixed distribution of P , we consider such $d_k = d_{k,P}$ (resp., $a_k = a_{k,P}$) that the total variation distance δ_k^d (resp., δ_k^a) between $P + d_{k,P}$ (resp., $P + a_{k,P}$) and the random variable of the k^{th} departure (resp., arrival) is minimised. Let $\delta_P = \max_k \{\delta_k^d, \delta_k^a\}$. Then, we find an optimal distribution of P^* such that $\delta_{P^*} = \min_P \delta_P$. Finally, we let $d_k = d_{k,P^*}$ and $a_k = a_{k,P^*}$, see Table 4.1.

Note that the distribution of P and the moments d_k completely determine the *capacity* of the lane, i.e., the average number of vehicles that can depart during one cycle in *over-saturated conditions*. The acquired capacity can differ from the

Figure 4.10: The trajectories of the delayed vehicles departing from a lane during the green time of 38 seconds for a) odd- and b) even-numbered departures. Black dashed lines show the trajectory of a vehicle under the free-flow speed.

one determined by SUMO, see Figure 4.12, and, therefore, may lead to significant errors in the estimation of the average delay especially for high arrival rates. Thus, for each green-time length g , we consider its own distribution of delay P . For a fixed value of g , we choose the distribution of $P = P_g$ in such a way that the capacity of green times g and $g - 1$ are the same as in SUMO. This means that

Figure 4.11: The distributions of the moments of a) departure and b) arrival for odd-number departures. The arriving intersection is 100 meters away from the departing intersection.

we incorporate all the randomness of the departures in delay P of the first delayed departure, while leaving the inter-departure times deterministic.

4.2.3. Comparison with SUMO

In this subsection, we compare the results of our model and SUMO. We use the same default internal parameters of SUMO as in [Subsection 4.2.2](#). We choose basic values of [phase](#) lengths, [offsets](#), cycle length, arrival rates and turning probabilities

and then vary these parameters to show that our model gives a good prediction of SUMO results.

To implement *semi-actuated* control in SUMO, we use the information from the detectors that are placed before the stop line of each minor lane. Such a detector can only detect vehicles in its direct neighbourhood and, therefore, is unable to observe whether or not there is a queue. To determine the end of the green time, we use the following algorithm. The green time for a minor lane turns yellow if either the time limit is reached or there was no detection during the last 3 seconds. On the one hand, this threshold of 3 seconds gives enough time for the next queued vehicle to pass. On the other hand, the probability that a new vehicle arrives exactly at this time is very low, and, therefore, the green time terminates when the queue has emptied. This implementation of semi-actuated control has consequences for the green time of the next lane, i.e., if the queue empties at second i then the green time for the next phase starts at second $\min(i + 3, s + g) + y$, where $s + g$ is the moment when the green time should be terminated and y is the yellow (switch) time. In what follows, we suppose $y = 6$ seconds.

We consider a tandem of two intersections, as in [Figure 4.1](#). There are 4 phases, see [Figure 4.2](#). The basic parameters are as follows:

- ▶ the arrival rates from north and south are $\lambda_n = \lambda_s = 0.025$ veh/sec,
- ▶ the arrival rates from west and east are $\lambda_w = \lambda_e = 0.075$ veh/sec,
- ▶ the probabilities to turn north or south are $p_n = p_s = 0.1$,
- ▶ the probabilities to turn west or east are $p_w = p_e = 0.3$,
- ▶ the cycle length is 60 seconds and the yellow (switch) time is 6 seconds,
- ▶ the lengths of phases are $g_{min} = 9$, $g_{maj} = 16$ and $g_{turn} = 2$ seconds for phases 1 and 2 (phases for minor lanes), phase 3 (major lanes) and phase 4 (left turn for major lanes), respectively,
- ▶ the offset (of phase 1) of the second intersection from the first intersection is 10 seconds.

For these parameters, the loads on the straight-going major lanes and minor lanes are approximately the same (about 0.72), which means that the contributions of these lanes in the average delay are comparable. The capacities of minor and major lanes under *fixed* control, see [Figure 4.12](#), are 2.1 and 5.0 veh/cycle, respectively, which correspond to the arrival rates of 0.035 and 0.083 veh/sec. The turning probabilities are chosen so that the arrival rate from one intersection to another is the same as $\lambda_w = \lambda_e$. In this way, the major lane at the downstream intersection is in average as busy (in terms of arrivals) as at the upstream intersection. There are many possible ways to set the basic parameters, but for simplicity we use a symmetric case.

Figure 4.12: The capacity of a lane as a function of green-time length.

To find the difference between results of our model and SUMO, we consider the following changes in the basic parameters, while keeping others fixed:

- $\lambda_n = \lambda_s = 0.035 \cdot v/20$, where $v = 1, \dots, 19$,
- $\lambda_w = \lambda_e = 0.083 \cdot v/20$, where $v = 1, \dots, 19$,
- $p_e = p_w = 3p_n = 3p_s = v/20$, where $v = 0, \dots, 10$,
- offsets from 0 to 59 seconds,
- phase lengths $g_{min} = v$, $g_{maj} = 34 - 2v$ and $g_{turn} = 2$, where $v = 7, 8, 9, 10$,
- cycles of length 60, 64, 68, 72 and 76 seconds with corresponding phase lengths $g_{min} = 9 + v$, $g_{maj} = 16 + v$ and $g_{turn} = 2 + v$, where $v = 0, \dots, 4$.

In this chapter, for the sake of conciseness, we only plot and discuss the results for different values of $\lambda_n = \lambda_s$ and offsets. Afterwards, we give the average values of the difference in all cases.

In [Figure 4.13](#), we plot the effect of the load at a minor lane on the average delay. This load influences the delay on the minor lane and, under semi-actuated control, both capacity and delay on the major road. As can be seen from [Figure 4.13b](#), the drivers that use the minor lanes lose approximately 10 seconds due to actuated control for each load. However, the external major lanes (straight-going and right-turning vehicles) benefit from it up to 13 seconds for right-turning vehicles, see [Figure 4.13c](#). Using SUMO, we find delay for each vehicle. This information is insufficient to compute the average delay for an external straight-going major lane since the vehicles that use this lane encounter more delay later in the system. For loads up to 0.8, semi-actuated control gives a shorter delay for an arbitrary vehicle, see [Figure 4.13a](#). For higher loads on the minor lane, the results of both controls are indistinguishable. Note that our model gives good results both

Figure 4.13: The effect of the minor-lane load on the average delay a) of an arbitrary vehicle, b) for a minor lane, c) for an external right-turning major lane, d) for an external left-turning major lane. Note the small scale of the average delay in subfigure d).

in total and for each lane. As can be seen from Figure 4.13d, the fluctuation in SUMO results are of the same scale as the difference between SUMO results and model (2-3 seconds). Our model is underestimating the delay only for very high load on the minor lane. As we saw in Subsection 4.2.1, this may be attributed to the independence assumption.

In Figure 4.14, we plot the average delay as a function of the offset. Due to symmetric arrival rates, the results for offsets o and $60-o$ are the same. Note that the impact of the offset is essential. Indeed, the difference between the lowest and highest values of the average delay are 12 (resp., 5) seconds for fixed (resp., semi-

Figure 4.14: The effect of the offset between intersections on the average delay of an arbitrary vehicle.

actuated) control. Moreover, as not all vehicles experience this effect, it means that those which do (60% of vehicles) feel a difference of 8-20 seconds, depending on the control. The plots for both types of control have a W-shape. This happens because the average delay of a vehicle that goes from the west intersection to the east intersection has an asymmetric V-shape with the global minimum for offset of 10 seconds. The average delay of a vehicle that goes from the east intersection to the west intersection is a mirror copy of the first one. Together they cause a W-shape of the average delay of an arbitrary vehicle. Observe that, in this example, the offset 10 is almost optimal for both types of control. Note also that the travel time t between the intersections is equal to 6, and offset $o = t$ gives far-from-optimal results especially for fixed control. Later, in [Subsection 4.2.4](#), we show that fixed and semi-actuated control may have very different optimal offsets. We note that, as in the case of the minor arrival rate, our model accurately predicts the average delay. The maximum difference of about 3 seconds is achieved for offsets close to 0 and 60.

The results for all 117 scenarios are summarised in [Table 4.2](#). Note that the average difference between our model and SUMO is less than 2.4 seconds (3.2%) for all roads and for an arbitrary vehicle. Moreover, in 88% of the cases the average difference is less than 5 seconds. The relative difference is less than 5% in 82% of the cases. The reason that the difference is higher in the other cases is that some scenarios result in a high load for one or several lanes.

Delay	Average $ r_m - r_s $ (sec.)	Average $ r_m - r_s /r_s$ (%)	Cases with $ r_m - r_s $ < 5 sec. (%)	Cases with $ r_m - r_s /r_s$ < 5% (%)
Fixed control				
Average	2.36	3.17 %	88.03 %	82.05 %
Minor lane	1.11	1.38 %	99.15 %	99.15 %
Major lane (left turn)	1.31	1.73 %	97.44 %	96.58 %
Major lane (right turn)	0.93	2.65 %	97.44 %	95.73 %
Semi-actuated control				
Average	1.55	2.35 %	99.15 %	92.31 %
Minor lane	1.46	1.83 %	99.15 %	99.15 %
Major lane (left turn)	0.68	1.06 %	100.00 %	100.00 %
Major lane (right turn)	0.18	0.83 %	100.00 %	99.15 %

Table 4.2: The difference between model results (average delay per arbitrary vehicle or for a particular lane), r_m , and SUMO, r_s . The absolute difference, $|r_m - r_s|$, and the absolute value of the relative difference, $|r_m - r_s|/r_s$, are averaged over 117 scenarios we consider. Additionally, the percentages of the cases where the absolute values of differences are small (less than 5 seconds or 5%, respectively) are given.

4.2.4. Comparison of semi-actuated and fixed control

In this subsection, we compare semi-actuated and fixed controls in terms of capacity, average delay and optimal settings, see Subsections 4.2.4.1, 4.2.4.2 and 4.2.4.3, respectively. As before, we use the network of Figure 4.1, the arrival rates and turning probabilities are symmetric, i.e., $\lambda_n = \lambda_s$, $\lambda_w = \lambda_e$, $p_n = p_s$, $p_w = p_e$, and the phase sequence is given in Figure 4.2. In Subsection 4.2.4.4, we summarise the results.

4.2.4.1. Capacity of the lanes under fixed and semi-actuated controls

Consider the impact of semi-actuated control on the capacity. Due to semi-actuation, the major phase is longer and more vehicles can depart. The effect depends on the arrival rate to the minor roads. In Figure 4.15, we plot the stability regions for fixed and semi-actuated control for a cycle length of 60 seconds. The difference between these regions is shaded with dark grey. Note that the stability region depends not only on the type of control but also on the turning probabilities. Moreover, the probabilities to turn north or south, i.e., p_n and p_s , affect the region more than the probabilities to turn west or east, i.e., p_w and p_e ,

Figure 4.15: The stability regions in terms of λ_n and λ_w for different values of $p_n = p_s, p_w = p_e$.

compare Figures 4.15 b, c with d. Figures 4.15 b, d only differ for high values of $\lambda_n = \lambda_s$, namely more than 0.05 veh/sec. The difference arises from the fact that busy minor roads affect the stability of *internal lanes* (between intersections). In contrast, Figures 4.15 c, d are strikingly different. Observe that the stability region for fixed control is a stepwise function. Each step corresponds to a choice of green times. For high values of $\lambda_n = \lambda_s$, the steps may be not horizontal due to the aforementioned effect on the stability of internal lanes. Note that even if the arrival rates from north and south are higher than from west and east, it may be beneficial to use semi-actuated control since the capacity of the “minor” lanes can be increased, see Figures 4.15 a, d.

minor load	major load				
	0.50	0.60	0.70	0.80	0.90
0.50	10.14	10.53	11.86	15.58	29.15
	19.23%	19.18%	20.12%	23.37%	33.53%
0.60	6.65	7.04	8.14	11.61	24.51
	11.37%	11.62%	12.59%	16.00%	26.29%
0.70	3.28	3.86	5.04	8.20	20.23
	4.92%	5.65%	6.98%	10.18%	19.76%
0.80	0.39	1.00	2.25	4.92	16.05
	0.48%	1.22%	2.64%	5.23%	13.54%
0.90	-2.15	-1.64	-0.67	1.52	11.11
	-1.78%	-1.37%	-0.55%	1.17%	6.99%

Table 4.3: The difference of the average delay per vehicle between fixed, r_f , and semi-actuated, r_a , control. The percent values correspond to the relative difference $(r_f - r_a)/r_a$. The phase lengths are $g_{min} = 9$, $g_{maj} = 16$, $g_{turn} = 2$ seconds. The turning probabilities are $p_n = p_s = 0.143$, $p_w = p_e = 0.5$.

Figure 4.16: The stability regions in terms of λ_n and λ_w . The turning probabilities are $p_n = p_s = 0.143$, $p_w = p_e = 0.5$. The thick line shows the border of the stability region for $g_{min} = 9$, $g_{maj} = 16$, $g_{turn} = 2$ independently of the control type. Crosses represent cases where we found that fixed control performs better than semi-actuated control.

4.2.4.2. Impact of the major and minor loads

In this subsection, we fix the length of each phase and analyse how loads on minor and major roads influence the difference in the average delay per vehicle between fixed and semi-actuated control. We choose the best *offset* for each type of control. In Table 4.3, we give the average delay difference (in seconds and %) between the optimal results (in terms of offsets) for fixed and semi-actuated control. The green times for minor direction, g_{min} , major direction, g_{maj} , and left turn on the major road, g_{turn} , are 9, 16 and 2 seconds, respectively. Semi-actuated control performs better in almost all cases, except for a very high minor load and a medium major load. In the latter case, fixed control performs better. The reason is that semi-actuated control is disadvantageous for the minor lanes and favours the major lanes, see also Figure 4.13 b). When the load on the minor lanes is not high, semi-actuated control increases capacity for the major road and, therefore, significantly decreases the delays for the major lanes. Since the arrival rate for these lanes is high, delay there has a higher weight in the average delay per vehicle compared to the delay for the minor lanes. We plotted the points of better fixed control in the capacity region for the case $g_{min} = 9$, $g_{maj} = 16$, $g_{turn} = 2$, see Figure 4.16. We would also like to note that if the minor load is (sufficiently) higher than the major load, the difference between controls is small.

4.2.4.3. Optimal settings for fixed and semi-actuated control

In this subsection, we show the difference between fixed and semi-actuated controls in terms of optimal settings. It is widely suggested to choose the settings for semi-actuated control in the same way as for fixed control, i.e., using Webster proportional policy, see Webster (1958). We show that the best possible settings for semi-actuated control are not the same as for fixed control. We consider two sub-problems: the choice of the green times and the choice of the offset.

To see the difference in choice of settings, we consider very high arrival rates for minor and major roads, namely $\lambda_n = \lambda_s = 0.0405, 0.0452$ and $\lambda_w = \lambda_e = 0.0810, 0.0905$ vehicles per second, respectively. Then, the minimum lengths of phases 1, 2, 3 and 4 are, respectively, 10, 10, 13 and 2 seconds. There is one more second that can be added to any of these phases. For each choice of phase, we also need to consider the best offset. In Figure 4.17, we plot the results for each choice of phase, offset and control type. Due to symmetry, we use only offsets up to 30 seconds. We see that for semi-actuated control, it is more beneficial to add 1 second to one of the minor phases (1 or 2), while for fixed control, it is better to add this second to the major phase (3). This can be explained in the following way. For semi-actuated control, adding 1 second for a minor phase means lower delays for the corresponding minor lane with a small effect on the delay of the major lane since if the second is unused it will be added to the major phase. However, for fixed control, the major lane is suffering long delays and it is more beneficial to lengthen phase 3.

Figure 4.17: The difference between fixed and semi-actuated control in terms of optimisation. The average delay per vehicle is plotted as a function of the offset for each choice of phase to which 1 second will be added. The arrival rates are a) $\lambda_n = \lambda_s = 0.0405$, $\lambda_w = \lambda_e = 0.0810$, b) $\lambda_n = \lambda_s = 0.0452$, $\lambda_w = \lambda_e = 0.0905$. The minimal required times for the phases 1, 2, 3 and 4 are 10, 10, 13 and 2 seconds, respectively. The load of the system is a) 0.85 and b) 0.95. The turning probabilities are $p_n = p_s = 0.15$, $p_w = p_e = 0.3$.

The best offset also depends on the control type, see Figure 4.17. The effect of the offset on the average delay is less for semi-actuated control than for fixed control, as in Figure 4.14. In particular, the plot for semi-actuated control is almost flat for offsets 12 to 16, which gives a wide choice of almost optimal offsets, while for fixed control the value 11 is a distinct minimum. In Figure 4.18, we consider the effect of the load on these graphs. We consider settings $g_{min} = 9$, $g_{maj} = 16$, $g_{turn} = 2$. Note that if the minor load is low, semi-actuated control performs better with offsets of 2 - 7 seconds, while the optimal offset for fixed control is 11 seconds. If the minor load is high, the results of both controls are closer and the dependency on the offset is similar. The best possible offsets in this case are in the interval 9 - 11. The reason for it is that the major lanes experience almost the same green times for both types of control.

Figure 4.18: The average delay per vehicles as a function of offset for different loads ρ_m and ρ_M at the minor lanes and at the major straight-going lanes. The settings are $g_{min} = 9$, $g_{maj} = 16$, $g_{turn} = 2$.

4.2.4.4. Summary of results

In the previous subsections, we showed the differences between fixed and semi-actuated control. Semi-actuated control provide higher capacity, which leads to better performance in terms of the average delay if the minor lanes are not very busy. However, the average delay for the minor lanes is higher than under fixed control. The optimal traffic-light settings significantly depend on the type of control. In particular, fixed control may require longer green time for the major lanes compared to the optimal green-time allocation for semi-actuated control. Moreover, the best offset for semi-actuated control considerably changes with the load, while under fixed control, it remains almost the same for varying loads.

4.3. Conclusions

In this chapter, we developed an analytical model for a tandem of two intersections. We considered two types of the traffic control for an arterial road: fixed and semi-actuated. Our model predicts the average queue length and the average delay for each lane.

Compared to deterministic approaches, our model takes the randomness of the driver behaviour and the arrival process into account and, therefore, is more accurate. We showed that, compared to a traffic microsimulation, our model accurately predicts the expected delay of an arbitrary vehicle as well as the expected delay for each lane. The average absolute value of error is only 2.36 seconds and in the majority of the cases (88%) it is less than 5 seconds. Furthermore, this accuracy is achieved in a short time. For fixed control, one scenario is calculated in less than 0.5 seconds.

We have compared the performance of fixed and semi-actuated controls, and have found that even though semi-actuated control increases the capacity of the major road, it may perform worse for poor settings because it is always disadvantageous for the minor lanes. We have also concluded that the optimisation of the settings and offsets significantly depends on the type of control.

In [Chapter 5](#), we extend our model for larger traffic-light networks. In particular, we describe the arrival process at any internal lane. For the purpose of traffic-jam prediction, we study the transient behaviour of the system in [Chapter 6](#).

Chapter 5

Green-wave efficiency for a tandem of traffic-light intersections

‘If you are in a rush, the traffic lights are always red when you get to them.’

Murphy’s Law of Traffic Lights

Green waves on arterial roads increase coordination between intersections aiming to reduce the delay and the number of stops per vehicle. Coordination algorithms in literature usually assume deterministic flow of vehicles, whereas randomness due to, e.g., reaction speed of the drivers and their route choice may seriously affect the efficiency of green waves and significantly influence the delay of the vehicles, see [Calvert et al. \(2012\)](#).

This chapter introduces a stochastic green-wave efficiency measure. The common measure of efficiency, i.e., *bandwidth*, is deterministic, see [Gartner and Stamatidis \(2002\)](#). Bandwidth is defined as the (time) width of a hypothetical band in the space-time diagram such that vehicles travelling along this band with the *free-flow speed* encounter a green wave. This measure of green wave efficiency ignores the stochastic effects and, therefore, does not represent the real flow of vehicles in the system. The quality of a green wave for a particular vehicle can be measured by the number of intersections crossed without stopping. Therefore, we propose a new green-wave efficiency measure: the fraction of intersections on a route crossed without stopping by an arbitrary vehicle. This measure is between 0 and 1, where 0 means that all vehicles stop at each intersection, and 1 represents an ideal green wave.

To evaluate the green-wave efficiency, we extend the model of [Chapter 4](#) to

Figure 5.1: An example of a) a tree-like network and b) a network with a cyclic path.

any *tree-like network*, i.e., any network without cyclic paths, see Figure 5.1. Note that the results of Subsection 4.1.2 can be used for any arrival process. Therefore, it is sufficient to characterise the arrivals at every downstream lane and to find the green-wave efficiency measure in terms of model parameters. Characterisation of the arrival process at downstream intersections is lengthy and technical and is, therefore, deferred to Appendix 5.A.

In our numerical experiments, we consider the relation between green-wave efficiency and average delay per vehicle for a network with three intersections in tandem. Using a genetic algorithm with specialised subpopulations, we optimise for the maximum green-wave efficiency and the minimum average delay. We show that green waves may increase the average delay per vehicle. We also consider the Pareto front, which provides insights into the relation between these objective functions depending on the green times and offsets. Comparison of the results for our approach with two deterministic approaches for facilitating green waves shows the importance of including both stochastic and acceleration effects.

This chapter is structured as follows. In Section 5.1, we describe the model that we use to find green-wave efficiency and average delay per vehicle. Then, we explain the genetic algorithm and the deterministic coordination approaches in Section 5.2. In Section 5.3, we present numerical results and, in Section 5.4, we summarise our conclusions. In Appendix 5.A, we describe the underlying analysis of the arrival processes at the intersections.

5.1. Green-wave efficiency and average delay per vehicle

In this section, we apply the results of Chapter 4 to a tree-like network of *traffic-light intersections* under *fixed* control to find our main performance measures: green-wave efficiency and average delay per vehicle. For each lane, we use deterministic green time and a random delay of the first delayed vehicle, see Subsection 5.1.2.

5.1.1. Performance measures

We consider a traffic-light network operating under fixed control. Recall that we decompose the network into individual lanes and consider the queueing process at each lane individually. The arrival process at a lane is obtained from the departure processes of upstream lanes, taking into account acceleration time of vehicles, offsets and the vehicle routing probabilities, see [Appendix 5.A](#) for a detailed description.

For each lane, we are interested in the fraction of vehicles that passes this intersection without stopping, i.e., the probability of encountering green light and no queue upon arrival, and in the average delay per vehicle caused by the traffic light. Let Λ denote the arrival rate of vehicles to the system, λ_u the arrival rate at lane u , p_u^{ws} the probability of passing lane u without stopping, δ_u the average delay at lane u , I (resp. I^{ws}) the number of intersections passed (resp. passed without stopping) by an arbitrary vehicle. [Subsection 5.1.2](#) gives explicit expressions for p_u^{ws} and δ_u . We are now ready to introduce our main performance measures.

Definition 5.1 (Green-wave efficiency). The *green-wave efficiency* Γ is the fraction of intersections passed without stopping:

$$\Gamma = \frac{\mathbb{E}I^{ws}}{\mathbb{E}I}.$$

Observe that λ_u/Λ is the visit ratio for lane u , i.e., the expected number of times an arbitrary vehicle visits lane u . Since we consider tree-like networks, each lane is visited at most once, so λ_u/Λ is the probability that an arbitrary route contains lane u . The green-wave efficiency can be evaluated as follows:

$$\Gamma = \frac{\mathbb{E}I^{ws}}{\mathbb{E}I} = \frac{\sum_u p_u^{ws} \lambda_u / \Lambda}{\sum_u \lambda_u / \Lambda} = \frac{\sum_u p_u^{ws} \lambda_u}{\sum_u \lambda_u}, \quad (5.1)$$

i.e., the green-wave efficiency can be interpreted as the probability that a vehicle arriving at a lane proceeds without stopping. By analogy, the *average delay per vehicle* Δ can be obtained as follows:

$$\Delta = \frac{\sum_u \delta_u \lambda_u / \Lambda}{\sum_u \lambda_u / \Lambda} = \frac{\sum_u \delta_u \lambda_u}{\sum_u \lambda_u}. \quad (5.2)$$

Remark 5.2 (Tree-like network assumption). Arrivals to a lane are caused by departures from all upstream lanes. In a cyclic network (recall [Figure 5.1b](#)), arrivals to a lane may depend on the departures from the *same* lane, which considerably complicates the analysis. Therefore, we consider only tree-like networks, which allows us to explicitly characterise the arrival process to each lane.

Remark 5.3 (Difference with the two-intersections case of [Chapter 4](#)). For two intersections, [Chapter 4](#) characterises the arrivals from an upstream lane using the joint distribution of the number of delayed vehicles and vehicles that experienced

a green wave. For larger networks, in addition, it is important to take into account whether the vehicles that proceeded without stopping at the upstream intersection were delayed at any further upstream intersections, which significantly complicates the characterisation of the arrival process, as can be seen from the results in [Appendix 5.A](#).

5.1.2. Model for a single lane

In this subsection, we recall the model of [Subsection 4.1.2](#) and find values of p_u^{ws} and δ_u in terms of model parameters. In what follows, we omit the subscript u as we consider a fixed lane.

First, we state small differences due to fixed control. Second, we give the notation and the pgf of the queue length at the beginning of a cycle, see [Subsections 5.1.2.1](#) and [5.1.2.2](#). Then, we find the average queue length at the lane, the probability of leaving the lane without stopping and the average delay, see [Subsections 5.1.2.3](#), [5.1.2.4](#) and [5.1.2.5](#), respectively.

Consider a cycle of length c seconds that starts at second 0. Let s denote the beginning of the green time and e be the first second of the consecutive red time. Note that compared to [Chapter 4](#), the green time beginning is a deterministic variable. During the green time, there are free and delayed departures. A free departure may occur at any second of the green time when a vehicle arrives at the lane, finds no queue and, hence, leaves without stopping in the same second in which it arrived. Thus, a free-departing vehicle experiences a green wave. A delayed departure occurs at second $s + P + d_k < e$ if there is a non-empty queue, where P is the (random) delay due to drivers' distraction, see [Remark 4.6](#), and d_k are deterministic departure moments, see [Subsection 4.2.2](#). We make the following assumption on the values of P for different cycles.

Assumption 5.4. The distraction variable P is independent of other random variables and of its realisations in the previous cycles.

Arrivals to the queue form a MAP. We assume that the arrivals at different cycles are independent and identically distributed. Moreover, there is not more than one arrival per second. Each second a vehicle arrives with probability λ_i^l depending on the current state $L_i = l \in \mathcal{L}_i$ of the underlying Markov chain, where $i = 0, \dots, c-1$ is the second of the cycle, l represents additional information that defines the arrivals, e.g., the number of delayed departures at an upstream lane, and $\mathcal{L}_i$ is the space of possible states at second i . Transition probabilities $p_i^{ll'} = \mathbb{P}(L_{i+1} = l' | L_i = l)$ of the Markov chain are independent of the previous states and arrivals. The external arrival process to a lane at the edge of the network is modelled as a Bernoulli process.

Remark 5.5 (Cycle length and multiple green times). Our model can readily be extended to include multiple green times during a cycle, which allows us to include networks where some intersections have, for example, a cycle of double length.

5.1.2.1. Notation

The notation in this chapter is almost the same as in [Chapter 4](#). Since the green time is deterministic, and we include a random delay P , the conditional probabilities and pgfs are given with regard to event $P = \xi$ instead of $B = s$, see [Chapter 4](#).

Let X_i denote the steady-state queue length at second i of the cycle, with pgf $X_i(z) = \mathbb{E}(z^{X_i})$, and mean $Q_i = \mathbb{E}(X_i)$. We also use the following notation:

$$\begin{aligned} Q_{i,\xi} &= \mathbb{E}(X_i | P = \xi), & Q_{i,\xi}^l &= \mathbb{E}(X_i | P = \xi, L_i = l), \\ \nu_\xi &= \mathbb{P}(P = \xi), & \sigma_i^l &= \mathbb{P}(L_i = l), \\ w_{i,\xi}^l &= \mathbb{P}(X_i = 0 | P = \xi, L_i = l), & \alpha_{ji,\xi}^l &= \mathbb{P}(X_i = 0 | X_0 = j, P = \xi, L_i = l), \\ x_j &= \mathbb{P}(X_0 = j). \end{aligned}$$

The law of total probability gives

$$w_{i,\xi}^l = \sum_{j=0}^k \alpha_{ji,\xi}^l x_j, \text{ for } i < s + \xi + d_{k+1}, \quad (5.3)$$

and

$$Q_i = \sum_{\xi} \nu_\xi Q_{i,\xi}, \quad Q_{i,\xi} = \sum_{l \in \mathcal{L}_i} \sigma_i^l Q_{i,\xi}^l. \quad (5.4)$$

[Assumptions 4.7](#) and [5.4](#) on independence of arrivals and values of P for different cycles imply that the queue length at the beginning of a cycle is independent of state L_0 and P , so $Q_{0,\xi}^l = Q_{0,\xi} = Q_0$.

The number of possible delayed departures in a cycle depends on the value of the distraction variable P . Let n_ξ be the maximum number of delayed departures during a cycle given that $P = \xi$, i.e., $n_\xi = \max\{m : s + \xi + d_m < e\}$, and $n = \max_\xi n_\xi$.

It is inconvenient to use the second of the beginning of a departure as an index. Therefore, we use the following notation:

$$\bar{\mathcal{L}}_{k,\xi} = \mathcal{L}_i, \quad \bar{\sigma}_{k,\xi}^l = \sigma_i^l, \quad \bar{\alpha}_{jk,\xi}^l = \alpha_{ji,\xi}^l, \quad \bar{w}_{k,\xi}^l = w_{i,\xi}^l,$$

where $i = s + \xi + d_k + 1$ for $k > 0$ and $i = s$ for $k = 0$.

Finally, we introduce notation for the pgfs of the arrivals. Let $A(z)$ denote the pgf of all arrivals during a cycle, $f_{k,\xi}^l(z)$ (resp., $g_{k,\xi}^l(z)$) the pgf of the arrivals starting from second i (resp., $s + \xi + d_{k+1} + 1$ for $k < n_\xi$ or e for $k = n_\xi$) till the end of the cycle given that $L_i = l$, where, for convenience, $i = s + \xi + d_k + 1$ for $k > 0$ and $i = s$ for $k = 0$.

5.1.2.2. Pgf of the queue length at a lane at the beginning of a cycle

To obtain the pgf of the queue length at the beginning of a cycle, we relate the pgf of the steady-state queue length at the beginning and the end of a cycle, and

use that $X_0(z) = X_c(z)$, which gives an equation on $X_0(z)$, see Subsection 4.1.2.3 and Remark 4.9 therein:

$$X_0(z) = \frac{\sum_{j=0}^n x_j f_j(z)}{z^n - A(z)C(z)}, \quad (5.5)$$

where

$$C(z) = \sum_{\xi} v_{\xi} z^{n-n_{\xi}},$$

and

$$f_j(z) = \sum_{\xi} \sum_{k=j}^{n_{\xi}} \sum_{l \in \mathcal{L}_{k,\xi}} v_{\xi} \bar{\sigma}_{k,\xi}^l \bar{\alpha}_{jk,\xi}^l \left(g_{k,\xi}^l(z) z^{\mathbb{1}\{k \neq n_{\xi}\}} - f_{k,\xi}^l(z) \right) z^{n-n_{\xi}+k}. \quad (5.6)$$

The unknown probabilities x_j can be found similar to the way described in Remark 4.5.

5.1.2.3. Average queue length

The average queue length at the beginning of a cycle is obtained from (5.5) as $Q_{0,\xi}^l = Q_{0,\xi} = Q_0 = X_0'(1)$. Consider the average queue length at state l in second i with $P = \xi$. If second i is red, then the average queue length increases by λ_i^l during second i . If second i is green, the queue does not change if it is empty. Thus, the average queue length will increase by $\lambda_i^{l'}(1 - w_{i,\xi}^l)$ for second $i \neq s + \xi + d_k$ or by $(\lambda_i^{l'} - 1)(1 - w_{i,\xi}^l)$ if $i = s + \xi + d_k$. Therefore, the average queue length at seconds $i = 1, \dots, c-1$ follows from the following equations:

$$\begin{aligned} Q_{i+1,\xi}^l &= \sum_{l' \in \mathcal{L}_i} \frac{p_i^{l'} \sigma_i^{l'}}{\sigma_{i+1}^l} (Q_{i,\xi}^{l'} + \lambda_i^{l'}), & \text{for } i < s \text{ or } i \geq e, \\ Q_{i+1,\xi}^l &= \sum_{l' \in \mathcal{L}_i} \frac{p_i^{l'} \sigma_i^{l'}}{\sigma_{i+1}^l} (Q_{i,\xi}^{l'} + \lambda_i^{l'}(1 - w_{i,\xi}^l)), & \text{for } s \leq i < e \text{ and } i \neq s + \xi + d_k, \\ Q_{i+1,\xi}^l &= \sum_{l' \in \mathcal{L}_i} \frac{p_i^{l'} \sigma_i^{l'}}{\sigma_{i+1}^l} (Q_{i,\xi}^{l'} + (\lambda_i^{l'} - 1)(1 - w_{i,\xi}^l)), & \text{for } s \leq i < e \text{ and } i = s + \xi + d_k, \end{aligned}$$

where $p_i^{l'} \sigma_i^{l'} / \sigma_{i+1}^l = \mathbb{P}(L_i = l' | L_{i+1} = l)$ is the backward transition probability of the MAP. The average queue length at the lane is

$$Q = \frac{1}{c} \sum_{i=0}^{c-1} Q_i = \frac{1}{c} \sum_{i=0}^{c-1} \sum_{\xi} \sum_{l \in \mathcal{L}_i} v_{\xi} \sigma_i^l Q_{i,\xi}^l. \quad (5.7)$$

5.1.2.4. Probability of leaving the lane without stopping

For the green-wave efficiency, we need to obtain the probability p^{ws} that an arriving vehicle finds a green light and no queue at the lane, see (5.1). Dividing the expected number of vehicles per cycle that experience a **green wave** by the expected number of arrivals per cycle, we find

$$p^{ws} = \frac{\sum_{i=s}^{e-1} \sum_{\xi} \sum_{l \in \mathcal{L}_i} v_{\xi} \sigma_i^l w_{i,\xi}^l \lambda_i^l}{A'(1)}.$$

5.1.2.5. Average delay

Using Little's law, see [Little \(1961\)](#), we find the average **waiting time** $Q/A'(1)$ at a lane from the average queue length Q , defined in (5.7). As before, the vehicles can depart slowly from the lane, causing (additionally to the waiting time) the **acceleration delay** for itself and the following vehicles. The formula for the average acceleration delay $\mathbb{E}AD$ requires additional notation and is given in [Appendix 5.A.4](#). Together with the waiting time, the acceleration delay gives the **(total) delay** caused by the traffic light at the lane:

$$\delta = Q/A'(1) + \mathbb{E}AD.$$

5.2. Design of green waves

In this section, we describe several ways of facilitating **green waves** for a tandem of **traffic-light intersections**. We consider a network of three intersections in tandem, see [Subsection 5.2.1](#). In [Subsection 5.2.2](#), we suggest a genetic algorithm that uses predictions of our model, see [Section 5.1](#), to find non-dominated settings in terms of the average delay and green-wave efficiency. We also describe two deterministic ways of coordinating the intersections: the build-in optimisation tool of the traffic microsimulation suite SUMO and MAXBAND, see [Subsection 5.2.3](#). For simplicity, we fix the cycle length, **phase schedule** and the speed limits for all algorithms. In this way, we can focus on the differences between stochastic and deterministic approaches. Moreover, fixing the cycle length allows us to compare the algorithms for different **loads**. When the cycle length is fixed, the traffic load of the network equals ρ if multiplying the arrival rates to the network by at least $1/\rho$ renders an unstable system, which means that no possible traffic-light settings (with the given cycle length) exist to accommodate all arriving vehicles.

5.2.1. Network

We consider the following network. There are three intersections at regular distances of 100 meters that form an arterial in west-east direction, see [Figure 5.2](#). There are 24 lanes in total.

Figure 5.2: A tandem of three traffic-light intersections.

The arrival rate for the major roads (from west and east) is higher than the arrival rate for the minor roads (from north and south). We assume that most of the vehicles want to use the arterial road, i.e., the turning probability for the major road is low and for the minor road is high. We consider the same fixed phase schedule as in Chapter 4, see Figure 4.3. The minor roads do not have lanes for the turns, and, therefore, has one phase for all movements. For the major roads, it is better to use a special phase for the left turns because of the low turning probability. In this way, the major directions have higher capacity. The algorithms described below are easy to adjust for other phase schedules or number of intersections.

5.2.2. Genetic algorithm

The space of possible settings, i.e., green times and offsets, grows exponentially with the number of intersections. Our example in Section 5.3 below, has more than 10^{12} feasible solutions. Although we have explicit expressions, further properties of our performance measures (green-wave efficiency (5.1) and average delay (5.2)) are unknown. Therefore, we develop a genetic algorithm. This approach is widely used in optimization of traffic settings, see, e.g., Ceylan (2006), Di Gangi et al. (2016). Compared to local search, this approach considers a wide selection of solutions and, therefore, generally avoids bad local optima.

Our genetic algorithm pursues three goals: maximising the green-wave efficiency, minimising the average delay per vehicle and finding the *Pareto front* i.e., all *non-dominated solutions*. In our case, setting x dominates setting y , which we write as $x > y$, if $\Gamma(x) \geq \Gamma(y)$, $\Delta(x) \leq \Delta(y)$, and at least one of the inequalities is strict, where $\Gamma(\cdot)$ is the function of the green-wave efficiency and $\Delta(\cdot)$ is the function of the average delay.

We consider a set of 180 traffic-light settings, which is called a *population*, with three specialised subpopulations (of size 60), each with its own goal (minimal delay, maximal green-wave efficiency and Pareto optimality). For each setting

in the population, we compute the average delay and green-wave efficiency. By comparing all considered settings, we find and update (if necessary) the rank, $r(x)$, and neighbour number, $n(x)$, of each setting:

$$r(x) = 1 + |\{y: y > x\}|, \quad n(x) = |\{y: |\Gamma(x) - \Gamma(y)| < \varepsilon_\Gamma, |\Delta(x) - \Delta(y)| < \varepsilon_\Delta\}|,$$

where $|A|$ denotes the cardinality of set A . To find the *Pareto-optimal settings*, i.e., non-dominated settings, we minimise the following objective:

$$\Pi(x) = r(x) + 0.2n(x).$$

The neighbour number in the objective improves the diversity of the Pareto front.

After computing all three objective functions ($\Delta(\cdot)$, $\Gamma(\cdot)$ and $\Pi(\cdot)$) for one population, we generate a new population of the same size by combining settings of the previous generation as described below. For each objective, we produce a subpopulation of 60 new settings, where settings with better values of the objective function are more likely to be selected to produce the new generation. The combination of two settings is obtained by the operations *crossover* and *mutation*. Using our three objectives in one algorithm improves genetic diversity of the population, thus enhancing all three optimisations.

Let $g_i^{\min}$ denote the minimal green time for phase i , $i = 1, 2, 3, 4$, and g the rest of the available green time. Let s_1, s_2, s_3 determine the allocation of the extra green time over the phases as follows: the first phase receives s_1 extra seconds of green time, the second phase $s_2 - s_1$, the third $s_3 - s_2$, and the last phase $g - s_3$. We will refer to s_1, s_2, s_3 as *split moments* as they determine how the extra green time is split between phases:

$$\begin{aligned} s_1 &= g_1 - g_1^{\min}, & g_1 &= s_1 + g_1^{\min}, \\ s_2 &= s_1 + g_2 - g_2^{\min}, & g_2 &= s_2 - s_1 + g_2^{\min}, \\ s_3 &= s_2 + g_3 - g_3^{\min}, & g_3 &= s_3 - s_2 + g_3^{\min}, \\ & & g_4 &= g - s_3 + g_4^{\min}, \end{aligned}$$

where g_i is the green time of phase i . The equations for s_i determine the split moments. If the split moments are given, then the actual green times are obtained from the equations on the right-hand side.

For the crossover operation, we produce the ‘*chromosome*’ for each setting. Each chromosome is an array of the offset and split moments for each intersection. Crossover operation combines the chromosomes of two ‘parents’ to produce two ‘children’ as follows. The first part of the chromosome of one ‘parent’ until a random switching point is combined with the second part (after the switching point) of the chromosome of the second parent. The other parts of both chromosomes are used to produce the second child. Then, with a certain probability (0.4) a mutation occurs, which alters a randomly selected parameter. If an offset parameter is chosen, then it is changed to a random offset. If a split moment is

Figure 5.3: An illustration of the crossover and mutation operations. In the example, the randomly selected switching point occurs at the part of the chromosome that corresponds to the split moments of the middle intersection, see the dashed line. A, B, C, D represent offsets and split moments for other intersections. As the result of crossover operation, parameters 10, 17, D of the second parent is switched with 9, 16, B of the first parent. Subsequently, a random parameter is selected for mutation. In the example, the split moment 5 of the first child mutates to 6.

chosen, then we add randomly 1 or -1 only if the resulting split moment is feasible, see Figure 5.3 for an illustration. We run the algorithm for 30 generations followed by local search to improve the Pareto front, i.e., we compare the average delay and the green-wave efficiency for all neighbours of the current Pareto front and update the front accordingly until the front cannot be improved anymore.

Remark 5.6 (Runtime). We reduce runtime by saving the results for [external lanes](#) as their arrival processes are independent of the settings and offsets. Hence, the contribution of an external lane in the average delay per vehicle or the green-wave efficiency is identical for all settings that provide the same green time for this lane.

5.2.3. Deterministic approaches

In this subsection, we describe two deterministic approaches of facilitating the green waves. The first approach is the combination of the build-in tools of [SUMO](#). The second approach is the classical MAXBAND algorithm.

5.2.3.1. SUMO

Traffic microsimulation suite SUMO has two tools for optimising traffic-light control settings: *tlsCycleAdaptation* and *tlsCoordinator*. The first tool chooses the green times based on Webster allocation formula, see [Webster \(1958\)](#), given a fixed cycle length and switch (yellow) times. Based on these green times and the cycle length, the second tool tries to coordinate the intersections using the offsets. To distinguish the results of microsimulation from these tools, this approach for coordination of the intersections, following [García-Nieto et al. \(2013\)](#), will be called SUMO cycle program generator (SCPG).

Figure 5.4: Space-time diagram for the MAXBAND algorithm. The bold lines represent the red times for the major directions.

5.2.3.2. MAXBAND algorithm

In its simplest form, the MAXBAND algorithm, see Gartner and Stamatiadis (2002), finds optimal offsets between consecutive intersections for the given phase lengths of the traffic lights. We choose the same lengths of phases for each intersection. For our network and the chosen phase schedule (see Subsection 5.2.1), the problem of maximising the bandwidth can be formulated as follows:

$$\max b + k \bar{b}$$

$$w_i + b \leq g, \quad (5.8)$$

$$\bar{w}_i + \bar{b} \leq g, \quad (5.9)$$

$$(w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (\bar{\tau}_i + \tau_{i+1}) = 2t + m_i c, \quad (5.10)$$

$$m_i \in \mathbb{Z},$$

$$b, \bar{b}, w_i, \bar{w}_i \geq 0. \quad (5.11)$$

Here b ($\bar{b}$) is the west-east (east-west) bandwidth, k is the so called target ratio of the bandwidths, w_i ($\bar{w}_i$) is the time-gap between the west-east (east-west) green band and the following (previous) red time at intersection i , t is the travel-time from one intersection to another, g is the green time of the main phase, c is the cycle time, τ_i ($\bar{\tau}_i$) is the queue clearance time, and m_i is the so called loop integer. In Figure 5.4, we show the interrelation between these variables. All times are given in seconds. The target ratio between the bandwidths b and $\bar{b}$ is usually chosen to be equal to the ratio of the traffic flows.

For the optimal values of $b, \bar{b}, w_i, \bar{w}_i$, the offset of intersection $i + 1$ from intersection i is equal to $w_{i+1} - \tau_{i+1} + t - w_i$.

5.3. Optimal settings for an arterial road

This section studies the optimal settings for an arterial road. First, [Subsection 5.3.1](#) describes the parameters of the network. Then, [Subsection 5.3.2](#) invokes our genetic algorithm to obtain the settings that maximise green-wave efficiency and minimise the average delay per vehicle. For these settings, the analytical results for green-wave efficiency and average delay per vehicle are compared to the results of SUMO. [Subsection 5.3.3](#) compares our optimisation results with two deterministic approaches to optimise settings in traffic networks, described in [Subsection 5.2.3](#). Finally, [Subsection 5.3.4](#) analyses the Pareto fronts for different traffic loads.

5.3.1. Network parameters

Consider the network in [Figure 5.2](#). The arrival rates to the minor roads (from north and south) and from east are 20% and 50% of the arrival rate from west, respectively. The fractions of vehicles turning north or south from the main road at an intersection are both equal to 0.08, and the fractions turning east (0.4) and west (0.2) from the minor roads are chosen to compensate for the turning vehicles from the major road. The cycle length of traffic lights is fixed at 60 seconds and speed limits at 60 km/hour. The switch (yellow) time is 6 seconds. In our experiments, we have considered 5 arrival rates from west, with corresponding loads of the system ranging from 0.5 to 0.9. For light load, delay is mostly experienced by vehicles arriving during the red time, and it may be beneficial to shorten the cycle length. Therefore, we have not considered loads less than 0.5.

5.3.2. Optimal settings for green-wave efficiency and average delay

We have optimised the settings for maximum green-wave efficiency and for minimum average delay per vehicle using our genetic algorithm. In [Figure 5.5](#), the graphs labelled ‘Optimal delay, model’ and ‘Optimal green wave, model’ present the average delay and the green-wave efficiency under these optimisation criteria. The graphs labelled ‘Optimal delay, SUMO’ and ‘Optimal green wave, SUMO’ present the average delay computed by SUMO for the same settings. Note that SUMO is a simulation environment, so these graphs provide the 95% confidence intervals. Observe that [Figure 5.5](#) also presents graphs for the deterministic methods MAXBAND and SCPG discussed in [Subsection 5.2.3](#), see comparison below in [Subsection 5.3.3](#).

First, observe that for traffic loads in the range 0.5 – 0.8 our model accurately captures the average delay per vehicle, see [Figure 5.5a](#). For load 0.9 the discrepancy of 10% is explained by the effect of our [Assumption 4.7](#) on the independence of arrivals during different cycles, see also [Subsection 4.2.1](#). Note that operating under high load is undesirable, and system load may be reduced by lengthening

Figure 5.5: Performance of the green-wave design algorithms (optimisation of average delay or green-wave efficiency, SCPG or MAXBAND) in terms of a) average delay per vehicle and b) green-wave efficiency. The delay results of the model are compared with traffic microsimulation SUMO.

the cycle length as this reduces the idle time due to yellow times. It is worth mentioning that our model yields results over 240 times faster than SUMO, which allows our approach to be used for optimisation, whereas SUMO may only be used to compare a small number of settings.

Observe that optimal green-wave efficiency settings are disadvantageous for the average delay, except for very high load (0.9), see Figure 5.5a. For medium load (0.5, 0.6), green-wave efficiency favours the major directions and gives the minimal possible green time for one or both minor roads at each intersection leading to extremely long delays at the affected minor roads that counterbalance the gain in delay per vehicle on the arterial road, despite the relatively low traffic density at the minor roads. For higher load (0.7, 0.8), the optimal green time at intersection 1 for Phase 3 is shorter, see Table 5.1, resulting in a shorter platoon in the main direction. As a consequence, the platoon easier catches the green wave at subsequent intersections, but the reduced green time leads to longer delays at the first intersection for the main direction. Observe from Figure 5.5b that optimising the average delay also yields good results for green-wave efficiency for loads 0.7 - 0.9. For lower loads, these settings provide sufficient green times for all phases without discriminating other roads resulting in a considerable degradation of green-wave efficiency. It is interesting to observe that the settings for minimal delay for load 0.8 reported in Table 5.1 remain optimal for loads in the entire range

Green-wave design	Intersection 1				Intersection 2					Intersection 3				
	phases				offset	phases				offset	phases			
	1	2	3	4		1	2	3	4		1	2	3	4
opt. delay	7	7	20	2	10	7	7	20	2	18	7	7	20	2
opt. green wave	7	8	18	3	10	7	7	20	2	17	7	7	20	2
MAXBAND	7	7	20	2	6	7	7	20	2	12	7	7	20	2
SCPG	7	7	20	2	55	6	7	20	3	9	7	8	18	3

Table 5.1: Phases and offsets (in sec.) for load 0.8 for different green-wave designs.

0.6 – 0.9, which makes optimisation of delay very robust for networks settings. In summary, balancing green-wave efficiency and delay is tedious as providing a green wave to the arterial road may considerably increase delay for vehicles at minor roads or even for vehicles at the main road. In [Subsection 5.3.4](#), we continue comparison of our performance measures by analysing the Pareto fronts for different traffic loads.

5.3.3. Comparison with deterministic approaches

In this subsection, we compare our approach with two deterministic approaches of optimising traffic-light settings described in [Subsection 5.2.3](#). [Figure 5.5](#) indicates that the settings from our model outperform those of SCPG and MAXBAND. SCPG cannot handle loads exceeding 0.7 due to Webster allocation policy and rounding errors: for loads 0.8 and 0.9 this leads to unstable queues and a huge number of vehicles that are still present in the system at the end of the simulation in SUMO. In addition, for loads 0.5, 0.6 and 0.7 the average delay using the settings of SCPG is about 25% longer than optimal, while the green-wave efficiency is up to 2 times lower than optimal.

The input for the MAXBAND algorithm consists of the green times and the queue-clearance times (needed to clear the queues formed by arrivals from the minor lanes). We have chosen green times that maintain stability for high arrival rates, in agreement with the optimal-delay green times from our model. This choice can be seen as a discrete-time alternative to Webster’s proportional allocation since each phase has sufficient time that is independent of the load, see [Webster \(1958\)](#). The MAXBAND algorithm was further calibrated by choosing queue-clearance times (between 0 and 5 seconds) that minimise the average delay, which reduced the delays in MAXBAND by 10% compared to a deterministically computed queue-clearance time. Despite the optimisation of MAXBAND, the resulting settings give up to 10% longer average delay and less efficient green waves than our model. In particular, compared to MAXBAND the optimal delay settings from our model give up to 20% reduction of the average delay for the main (west-east) direction without disadvantages for other directions. In summary, our

Figure 5.6: The Pareto front for load $\rho = 0.6, 0.7, 0.8, 0.9$.

model clearly outperforms the deterministic approaches by taking into account the stochastic behaviour of traffic.

5.3.4. Pareto front

Section 5.2 introduced the Pareto front for joint optimisation of delay and green waves. Figure 5.6 plots all settings found by our genetic algorithm. The Pareto front is distinguished by the green colour and shape of the markers in Figure 5.6. Figure 5.7 gives the settings corresponding to the rectangle in Figure 5.6b.

Observe that the length of the Pareto front decreases with increasing load, resulting in a single point for load 0.9. The diversity of the Pareto front for lower loads implies that our objectives do not reinforce each other: as observed in Subsection 5.3.2 reducing green time for some lanes results in better green waves and longer average delays. Observe from Figure 5.7 that the points in the Pareto front are near neighbours in the settings space. For all plotted non-

Figure 5.7: A part of the Pareto front for load 0.7, see the dashed rectangle in Figure 5.6b. The shape of the points depends on the phase lengths, which are given in a set of three arrays corresponding to the intersections from west to east, and the phases in each array are ordered as in the phase schedule.

dominated solutions the offsets of the middle and east-most intersections from the west-most intersection are 10 and 17-19 seconds, respectively. Most of the differences between the settings are in the phases of the west-most intersection. Shortening the (minor) phase 1 or 2 by 1 second for the middle or east-most intersection leads to an abrupt increase in the delay by about 12 seconds, but may improve the green-wave efficiency, which explains the big gaps in the Pareto fronts. Changing the offsets for nearly-optimal solutions gives the line structure of the points in the figure.

5.4. Conclusions

In this chapter, we generalised our model for two intersections in tandem described in Chapter 4. We have proposed a measure of green-wave efficiency that represents the drivers' point of view and takes the stochastic nature of traffic into account. We have introduced a multi-objective genetic algorithm to find optimal settings for a tandem of intersections in terms of green-wave efficiency and average delay per vehicle. We have shown that green-wave efficiency and average delay may be conflicting performance measures. Therefore, we have also considered the Pareto front combining these two objectives.

We have shown that settings that maximise the green-wave efficiency can be achieved in one of two ways:

- a large share of the available green time is allocated to the main phase,

- the green time for the main phase is shorter for the first intersection in a tandem and longer for other intersections.

In the first case, the average delays are very long due to delay of traffic on the minor roads. The main-direction traffic has short delays and encounters a [green wave](#). This approach gives high values of green-wave efficiency for medium loads. The second case leads to a formation of a short and dense [platoon](#) that easily passes through the system. However, due to the shortened green time at the first intersection, the main traffic may experience long delays at the first intersection. Thus, optimal green waves may lead to long average delays, which is in contrast with the common belief that green waves result in lower delays, see [Tomescu et al. \(2011\)](#). In our experiments, the optimal settings for average delay seem to have the same green times for all loads and, thus, are very robust compared to the optimal green-wave efficiency settings. This green time allocation policy can be seen as a discrete-time alternative to Webster's proportional allocation, see [Webster \(1958\)](#).

In [Chapter 6](#), we apply our methods to large networks. This requires simple approximations of the arrival processes to increase the computation speed of our model.

5.A. Arrival process

This appendix presents the construction of the arrival process to a lane. [Subsection 5.A.1](#) describes the departures from a lane and its conversion to arrivals to a downstream lane. The general MAP to a downstream lane is constructed in [Subsection 5.A.2](#), and its transition probabilities in [Subsection 5.A.3](#). The acceleration delay is characterised in [Subsection 5.A.4](#).

5.A.1. Arrivals from an upstream lane

This subsection describes the departures from a lane and its conversion to arrivals to a downstream lane for a tandem of 3 lanes to provide the basis for the general description in [Subsection 5.A.2](#), see [Figure 5.8](#) for a graphical illustration. In this subsection, we assume, for simplicity, that there is no turning to and from the main road. The analysis for lanes 1 and 2 in the tandem partially repeats the description in [Chapter 4](#). However, there are certain differences.

As before, arrivals to an [external lane](#) (lane 1) form a Bernoulli process, which is modelled using a MAP that for each second has a single state, say state 0, with $\lambda_i^0 = \lambda$, $\sigma_i^0 = 1$, $p_i^{00} = 1$, $i = 0, \dots, c - 1$, where λ is the arrival rate.

Departures from this lane 1 are converted to arrivals to the downstream lane 2. These arrivals depend on the number of [delayed](#) departures from lane 1 and the distraction variable. Let D_u be the (random) number of delayed departures from lane u if the queue has emptied, and ∞ otherwise. As before, t is the travel time between two intersections, and a_k is equal to d_k plus the time needed for acceleration of the k^{th} delayed departure. The traffic light for lane 1 is green

Figure 5.8: The possible inputs from a) external lane 1 to lane 2 or b) internal lane 2 to lane 3. The distraction value is set to $P_1 = P_2 = 0$, and $s_2 = 8$. The states of the arrival processes are indicated by circles. Darker states have higher arrival rates. For different number m of delayed departures, $m = 0, 1, 2$, the arrivals to a downstream lane are not the same. Arrows give the transitions from the states, where grey arrows represent possible transitions to the next states in the input. The squares represent green and red time of the considered upstream lane. The grey boxes emphasise the shifts that the arrivals do in order to catch up with the leading accelerating vehicle. A green second with a free departure represents the possibility of free departure in case there would be an arrival. Details are provided in the text.

during seconds $s_1, \dots, e_1 - 1$. Here and later, we use the notation of Section 5.1 with additional subscript corresponding to the lane number. Consider some fixed value of the distraction variable, say $P_1 = \xi$. The arrivals to lane 2 can be characterised as follows. If there are no delayed departures, i.e., $D_1 = 0$, then there are $e_1 - s_1$ possible free departures, see the row $m = 0$ in Figure 5.8a. If $D_1 = m$, $0 < m < \infty$,

then there are m delayed departures followed by $e_1 - s_1 - \xi - d_m - 1$ possible free departures. Note that delayed departures leave the queue slowly and need some time to accelerate. The k^{th} departure departs at time $s_1 + \xi + d_k$. The time lost due to acceleration by the k^{th} delayed departure is $a_k - d_k$. Therefore this departing vehicle arrives at the downstream lane at time $s_1 + \xi + a_k + t$. Thus, if $0 < m < \infty$, the free departures are postponed by $a_m - d_m$ seconds, see Figure 5.8a for $m = 1, 2$. In the case $D_1 = \infty$, there are no free departures, only n_ξ delayed departures.

Consider the *internal lane 2*. With only one upstream lane 1, the MAP for this lane is a combination of all possible inputs from lane 1 complemented with *end states*, see input to lane 2 in Figure 5.8b. The offset between the intersections is such that all departures from lane 1 in case of $D_1 = 0$ are queued at lane 2. Thus, only vehicles delayed at lane 1 can pass lane 2 without stopping. The departures from lane 2 form arrivals to lane 3 and depend on the values of $D_2 = m$ and $P_2 = \xi$. There are two essential differences in the construction of the inputs to a downstream lane compared to the case of an external lane 1.

The first difference is between the arrival processes to lanes 1 and 2. Namely, the MAP for lane 2 may have several states just after the last (m^{th}) delayed departure, i.e., at second $i = s_2 + \xi + d_m + 1$ for $m > 0$ and $i = s_2$ for $m = 0$, see seconds $s_2 = 8$ (marked with $*$) and $s_2 + d_1 + 1 = 10$ (marked with $\dagger$) in Figure 5.8b. In this case, the event $L_{i,2} = l$ may be not independent of the event $D_2 = m$. For example, consider $m = 0$ and second $i = s_2 = 8$ in Figure 5.8b. The queue for the state l in second i in the top row is longer than for the other state, so that $\mathbb{P}(D_2 = 0 | L_{i,2} = l) < \mathbb{P}(D_2 = 0)$, i.e., events $D_2 = 0$ and $L_{i,2} = l$ are not independent. Note that the seconds with free departures in Figure 5.8 represent the potential possibility of a free departure in case there would be an arrival at this second. Also observe that a *free arrival* at the lane not necessary results in a free departure at the lane. For example, in Figure 5.8b, the free arrivals for the case $D_1 = 0$ arrive during the red time and result in delayed departures.

The second difference is that there may be gaps in the arrivals allowing a free-departing vehicle to catch up with the delayed vehicles and thus be postponed by less than $a_m - d_m$ seconds. Compare the grey boxes in Figure 5.8b for $m = 0$ and $m = 1$. For $m = 0$, the structure of the arrivals to lane 3 is the same as to lane 2 only the probabilities of the states are different due to event $D_2 = 0$, see also the description of the previous difference. Suppose $m = 1$. The only delayed departure needs to accelerate and is at second a_1 (marked by $\star$) in the input to lane 3. After this departure, there are two states of MAP to lane 2 at second $s_2 + d_1 + 1$ (marked by $\dagger$). Depending on the state, there are either no more *non-empty state*, i.e., states with non-zero arrival rate, (top row state) during current green time, or up to three non-empty states (bottom row). The empty states represent gaps in the arrivals and are collapsed in the input to lane 3 to give place to the real arrivals. In particular, the two delayed departures from lane 1 have caught up with each other and the delayed departure from lane 2 and are now consecutive arrivals to lane 3. The collapsing empty states may give rise to transitions to the rest of the input, indicated by the grey arrows. Below we use these observations to describe

the general MAP to an internal lane.

5.A.2. Description of the MAP to an internal lane

This subsection builds on the insight provided above to construct the arrival process to a considered internal lane. We divide the states in three types depending on the origin of the corresponding arrival. Then, for each type, we give state descriptions, arrival rates and probabilities. Finally, we describe the transitions.

There are three types of arrival states for an internal lane, see [Subsection 5.A.1](#). The first two types represent delayed and free departures from the upstream lanes. As before, the corresponding arrivals we call *delayed arrivals* and *free arrivals*, respectively. The last type of states is an end state, which represents a second between the arrivals from one upstream lane and from the next upstream lane. Note that not all empty states are end states. For example, the input to lane 2 at second s_2 (marked by *) in [Figure 5.8b](#) contains two empty states, but only the top one is the end state as it represents the end of arrivals from lane 1 in this cycle, while the bottom state is due to delayed arrivals from lane 1.

We describe the states of the considered lane by second i of the cycle of the considered intersection and sequence l , referred to as pair (i, l) . The first element of sequence l , denoted by χ , indicates the type, with $\chi = 1$ for delayed arrivals, $\chi = 0$ for free arrivals, and $\chi = -1$ for end states. For $\chi = 1$, the sequence l has the following form: $l = (\chi, u, s_u, \xi_u)$, where u is the index of upstream lane from which the delayed arrivals come, s_u is the green-time beginning of lane u , and ξ_u is the value of the distraction variable of lane u . We emphasise that the index u refers to the upstream lane u . The variables without this extra index correspond to the considered downstream lane. If $\chi = 0$, then $l = (\chi, u, i', l')$, where (i', l') is the corresponding pair of the arrival state for lane u . In general, $i' \neq i$, because the seconds are labelled differently at the upstream intersection due to the offset, and, moreover, the free arrival may be delayed. Sequence l' may have a similar format as l , which means that the state dimension is growing along a directed path in the system. Finally, states with $\chi = -1$ do not require additional information, i.e., $l = (\chi)$.

The arrivals to a considered lane also depend on the routing probabilities and the offset. Recall that c is the cycle length, and t is the travel time between intersections under the [free-flow speed](#), and denote by

- γ_u : the probability of turning from lane u to the considered lane,
- o : the offset between intersections, i.e., the cycle at the considered intersection starts o seconds later than the cycle at the upstream intersection,
- $t_o = t - o \bmod c$,
- $s'_u = s_u + t_o \bmod c$,
- Q_u : the (random) delay of vehicles from upstream lane u due to acceleration of the vehicles from lane $u - 1$.

Note that the vehicle that departs with the free-flow speed from the upstream intersection at second i of an upstream cycle arrives at the considered intersection at second $i + t - o$ of the downstream cycle, which starts with offset o . If the travel time is long, the arrival occurs at second $i + t_o \pmod{c}$ in one of the next cycles. For the considered intersection, arrivals in the current cycle then originate from one of the previous cycles of the upstream lane. As we assume the system to be stationary, we will analyse the system mod c , and identify seconds i and $i + c$ for any $i \in \mathbb{Z}$. The delay Q_u can be illustrated by Figure 5.8. There, the vehicles from an upstream lane arrive during a period of time which is substantially longer than the green time of the upstream lane. Therefore, the free-departing vehicles from lane u cannot arrive earlier than second $s_u + t_o + Q_u$ of the considered cycle. Now we describe the states, i.e., possible pairs (i, l) , the arrival rates and probabilities of these states.

First, we consider delayed arrivals with $l = (1, u, s_u, \xi_u)$. Fix the index u of the upstream lane and the value ξ_u . Consider pairs (i, l) for seconds $i = s'_u, \dots, s'_u + \xi_u + a_{n_{\xi_u, u}}$. The arrival rates of these states depend on the second. Namely, the states at seconds $i = s'_u + \xi_u + a_k$, $k = 1, \dots, n_{\xi_u, u}$, correspond to delayed arrivals and have arrival rate γ_u . The other states are empty and have no arrivals. Hence, for $i = s'_u, \dots, s'_u + \xi_u + a_{n_{\xi_u, u}}$:

$$\begin{aligned} \lambda_i^l &= \gamma_u, & \text{if } i = s'_u + \xi_u + a_k \text{ for some } k, \\ \lambda_i^l &= 0, & \text{otherwise.} \end{aligned}$$

The state probability also depends on the second since the queue may become empty. Consider second i between two arrivals, i.e., $s'_u + \xi_u + a_{k-1} < i \leq s'_u + \xi_u + a_k$ (or $s'_u \leq i \leq s'_u + \xi_u + a_1$ if $k = 1$). Then, the probability of state $l = (1, u, s_u, \xi_u)$ at second i is $\mathbb{P}(Q_u \leq i - s'_u, D_u \geq k, P_u = \xi_u)$, where $Q_u \leq i - s'_u$ reflects the event that the state has not collapsed due to arrivals from lane $u - 1$, which happens only to empty states with i close to s'_u , as values of Q_u are not more than 4 seconds. Event $D_u \geq k$ coincides with event that there are still delayed arrivals. Note that the random variable Q_u is independent of D_u and P_u because it is defined by departures from lanes $u - 1$ and, therefore, independent of what happens at lane u . Thus,

$$\sigma_i^l = \mathbb{P}(Q_u \leq i - s'_u) \mathbb{P}(D_u \geq k, P_u = \xi_u).$$

The main transition from delayed pair (i, l) is to pair $(i + 1, l)$. If $i \neq s'_u + \xi_u + a_k$ for all k , then the probability of the transition is 1. If $i = s'_u + \xi_u + a_k$, $k \neq n_{\xi}$, then this probability is $\mathbb{P}(D_u > k | D_u \geq k, P_u = \xi_u)$, i.e., the transition happens if there are more delayed departures. If there are exactly k delayed departures, then there are three possibilities. The first possibility is that there are some free departures from the same lane u , then there are multiple transitions to the possible non-empty states. The probabilities of these transitions depend on probability of being in a particular state given that $D_u = k$ and $P_u = \xi_u$. Moreover, as we discussed, several

empty states may collapse. The second (resp., third) possibility is that there are no more arrivals from this lane and $i + 1 < s'_{u+1}$ (resp., $i + 1 \geq s'_{u+1}$). In the former case, there is a transition to an end state with the probability that there are no more arrivals. In the latter case, the transitions go to the possible states that correspond to arrivals from lane $u + 1$. The probabilities of such transitions we give later in Table 5.2, after defining the required notation.

Second, we consider free arrivals. They are caused by arrivals to an upstream lane and may be delayed due to acceleration of the preceding vehicles. Let $R'_{i',u}$ denote the random variable of the delay caused to state l' at second i' of upstream lane u by the accelerating vehicles. We set $R'_{i',u} = \infty$ if this state does not generate a free departure, i.e., $\mathbb{P}(R'_{i',u} = \infty) = \mathbb{P}(X_{i',u} > 0 | L_{i',u} = l')$. In $R'_{i',u}$ we include delay Q_u , for more details see Subsection 5.A.3. For a non-empty state l' , i.e., $\lambda'_{i',u} > 0$, state (i, l) with $l = (0, u, i', l')$ is added to seconds $i = i' + t_o + r$, where r is a finite value of $R'_{i',u}$. Since an empty state l' , i.e., $\lambda'_{i',u} = 0$, is collapsed when delayed, we consider only $i = i' + t_o + r$ for $r = 0$. The probability of state (i, l) and its arrival rate is given by

$$\sigma_i^l = \mathbb{P}(R'_{i',u} = r) \sigma_{i',u}^{l'} \quad \lambda_i^l = \lambda'_{i',u} \gamma_u, \quad \text{for } i = i' + t_o + r,$$

where $\sigma_{i',u}^{l'}$ (resp., $\lambda'_{i',u}$) is the probability (resp., arrival rate) of state l' at second i' , $l' \in \mathcal{L}_{i',u}$, of upstream lane u . The possible transitions from pair (i, l) are transitions to the free-arrival states that correspond to the following non-empty or non-shifted states in the MAP of lane u . If there are no more arrivals from lane u , then, as before, there is a transition either to an end state (if $i + 1 < s'_{u+1}$) or to the possible states corresponding to arrivals from lane $u + 1$ (if $i + 1 \geq s'_{u+1}$).

Third, we consider end states. They should be added at each second in case the total probability of other states is less than 1:

$$\sigma_i^{(-1)} = 1 - \sum_{l \neq (-1)} \sigma_i^l.$$

The arrival rate is $\lambda_i^{(-1)} = 0$. The transitions are either to an end state (if $i + 1 \neq s'_u$ for all u) or to states that correspond to arrivals from lane u (if $i + 1 = s'_u$).

Finally, we need to determine the transition probabilities between the states. First of all, because we assume that arrivals in one cycle are independent of the previous, we set $p_{c-1}^{\hat{l}} = \sigma_0^{\hat{l}}$ for all states $l \in \mathcal{L}_{c-1}$ and $\hat{l} \in \mathcal{L}_0$. For seconds $i \neq c - 1$, the possible transitions are summarised in Table 5.2, where we use the following notation:

$$\begin{aligned} DP_{k,\xi_u,u}: & \quad D_u \geq k \text{ and } P_u = \xi_u, \\ E_{i,i',u}: & \quad \text{the event that there are only empty states during seconds} \\ & \quad i, \dots, i' - 1, \text{ i.e., } \lambda_{i,u}^{L_{i,u}} = \dots = \lambda_{i'-1,u}^{L_{i'-1,u}} = 0, \\ EL_{i,i',l,u}: & \quad E_{i,i',u} \text{ and } L_{i',u} = l, \\ d_{k,\xi_u,u} = & \quad s_u + \xi_u + d_k \pmod{c}, \end{aligned}$$

initial pair	transition probability	condition
transitions to $(i + 1, (1, u, s_u, \xi_u))$		
$(i, (1, u, s_u, \xi_u))$	$\mathbb{P}(D_u > k DP_{k, \xi_u, u}) \mathbb{1}\{k \neq n_\xi\}$	$i = a_{k, \xi_u, u}$
$(i, (-1))$	1	otherwise
$(i, (1, u', s_{u'}, \xi_{u'}))$	$\mathbb{P}(DP_{1, \xi_u, u})$	$i + 1 = s'_u$
$(i, (1, u', s_{u'}, \xi_{u'}))$	$\mathbb{P}(DP_{1, \xi_u, u}) \mathbb{P}NA_{k, \xi_{u'}, u'}^d$	$i = a_{k, \xi_u, u}, i + 1 \geq s'_u$
$(i, (0, u', i_{u'}, l_{u'}))$	$\mathbb{P}(DP_{1, \xi_u, u}) \mathbb{P}NA_{i_{u'}, l_{u'}, u'}^f$	$i + 1 \geq s'_u$
transitions to $(i + 1, (0, u, i'_u, l'_u))$		
$(i, (1, u, s_u, \xi_u))$	$\mathbb{P}(D_u = k, EL_{d_{k, \xi_u, u} + 1, i'_u, l'_u, u} DP_{k, \xi_u, u})$	$i = a_{k, \xi_u, u}, (5.12)$
$(i, (0, u, i_u, l_u))$	$\mathbb{P}(EL_{i_u + 1, i'_u, l'_u, u} L_{i_u, u} = l_u)$	(5.12)
$(i, (-1))$	$\mathbb{P}(D_u = 0, L_{s_u, u} = l'_u)$	$i + 1 = s'_u, i'_u = s_u$
$(i, (1, u', s_{u'}, \xi_{u'}))$	$\mathbb{P}(D_u = 0, EL_{s_u, i'_u, l'_u, u}) \mathbb{P}NA_{k, \xi_{u'}, u'}^d$	$i = a_{k, \xi_{u'}, u'} \geq s'_u - 1, (5.12)$
$(i, (0, u', i_{u'}, l_{u'}))$	$\mathbb{P}(D_u = 0, EL_{s_u, i'_u, l'_u, u}) \mathbb{P}NA_{i_{u'}, l_{u'}, u'}^f$	$i + 1 \geq s'_u, (5.12)$
transitions to $(i + 1, (-1))$		
$(i, (1, u, s_u, \xi_u))$	$\mathbb{P}NA_{k, \xi_u, u}^d$	$i = a_{k, \xi_u, u}, i + 1 < s'_{u+1}$
$(i, (0, u, i_u, l_u))$	$\mathbb{P}NA_{i_u, l_u, u}^f$	$i + 1 < s'_{u+1}$
$(i, (-1))$	1	$i + 1 \neq s'_u \forall u$

Table 5.2: Possible transitions $p_i^{ll'}$ of the Markov chain of arrivals for $i \neq c - 1$. For simplicity, $u' = u - 1$, for $u > 1$, and $u' = N$, for $u = 1$, where N is the number of upstream lanes.

$$\begin{aligned}
 a_{k, \xi_u, u} &= s'_u + \xi_u + a_k \bmod c, \\
 \mathbb{P}NA_{k, \xi_u, u}^d: & \text{ the probability that there are no departures} \\
 & \text{ from lane } u \text{ after the } k^{\text{th}} \text{ delayed departure, i.e.,} \\
 & \mathbb{P}(D_u = k, E_{s_u + \xi_u + d_{k+1, e_u, u}} | DP_{k, \xi_u, u}) + \mathbb{1}\{k = n_\xi\} \mathbb{P}(D_u = \infty | \\
 & DP_{k, \xi_u, u}), \\
 \mathbb{P}NA_{i, l, u}^f: & \text{ the probability that there are only empty states after pair} \\
 & (i, l) \text{ of lane } u, \text{ i.e., } \mathbb{P}(E_{i+1, e_u, u} | L_{i, u} = l).
 \end{aligned}$$

For transitions to a free-arrival state that corresponds to the pair (i'_u, l'_u) of lane u , we use the condition that state l'_u at second i'_u is either non-empty or non-shifted, i.e.,

$$i'_u + t_o = i + 1 \text{ or } \lambda_{i'_u, u}^{l'_u} > 0, i'_u + t_o \leq i. \quad (5.12)$$

5.A.3. Delayed and free departures

In the previous subsections, we used several events and probabilities that are not directly defined in the analysis of the model. In this subsection, we give their distributions.

First, we consider events $D_u = k$ and $P_u = \xi_u$. These events are not independent. Their joint distribution can be found as follows:

$$\begin{aligned} \mathbb{P}(D > k, P = \xi) &= \mathbb{P}(X_i \neq 0, P = \xi) = \\ &= \sum_{l \in \mathcal{L}_i} \mathbb{P}(X_i \neq 0 | L_i = l, P = \xi) \mathbb{P}(L_i = l, P = \xi) = \\ &= \sum_{l \in \mathcal{L}_i} (1 - \bar{w}_{k,\xi}^l) \bar{\sigma}_{k,\xi}^l \mathbb{P}(P = \xi), \end{aligned}$$

where $i = d_{k,\xi} + 1$, and we omitted index u .

Second, we focus on the probability $\mathbb{P}(EL_{i+1,i',l',u} | L_{i,u} = l)$. This probability only depends on the arrival process of the upstream lane, and can be found recursively. Indeed, event $EL_{i+1,i',l',u}$ means that the states at seconds $i+1, \dots, i'-1$ are empty, which also means that event $EL_{i+1,i'-1,l'',u}$ happens for some empty state $(i' - 1, l'')$. Thus, we get

$$\mathbb{P}(EL_{i+1,i',l',u} | L_{i,u} = l) = \sum_{\substack{l'' \in \mathcal{L}_{i'-1,u}: \\ \lambda_{i'-1,u}^{l''} = 0}} p_{l'-1,u}^{l''l'} \mathbb{P}(EL_{i+1,i'-1,l'',u} | L_{i,u} = l).$$

If $i' = i + 1$, then simply $\mathbb{P}(EL_{i+1,i',l',u} | L_{i,u} = l) = \mathbb{P}(L_{i+1,u} = l' | L_{i,u} = l) = p_i^{l''}$. Probabilities $\mathbb{P}NA_{i,l,u}^f$ can be obtained as a sum for $i' = e_u - 1$ and all empty states l' of probabilities $\mathbb{P}(EL_{i+1,i',l',u} | L_{i,u} = l)$. From that we find $\mathbb{P}NA_{k,\xi_u,u}^d$ as follows:

$$\begin{aligned} \mathbb{P}NA_{k,\xi_u,u}^d &= \sum_{\substack{l \in \mathcal{L}_i: \\ \lambda_{i,u}^l = 0}} \mathbb{P}NA_{i,l,u}^f \mathbb{P}(L_{i,u} = l, D_u = k | DP_{k,\xi_u,u}) + \\ &\quad + \mathbb{1}\{k = n_\xi\} \mathbb{P}(D_u = \infty | DP_{k,\xi_u,u}), \end{aligned}$$

where $i = d_{k,\xi_u,u}$. The probability $\mathbb{P}(L_{i,u} = l, D_u = k | DP_{k,\xi_u,u})$ can be found using the values of $x_j = \mathbb{P}(X_0 = j)$ and the transition probabilities of the arrival process to lane u .

Finally, we consider the distributions of the delays Q_u and $R_{i,u}^l$. To find them we use the auxiliary random variable $\tilde{R}_{i,u}^l$ of the delay caused to pair (i, l) by acceleration of the vehicles of lane u only, compare with definition of $R_{i,u}^l$ on page 154. The state l at second i is postponed by r seconds either if the previous state is not empty and postponed by r seconds, or if the previous state is empty and postponed by $r + 1$ seconds, or if the considered state is the first state after the k^{th} delayed departure, where $r = a_k - d_k$. Thus, we can evaluate probabilities recursively:

$$\begin{aligned}
\mathbb{P}(\tilde{R}_{i+1,u}^l = r) = & \sum_{\substack{l' \in \mathcal{L}_{i,u}: \\ \lambda_{i,u}^{l'} > 0}} p_{i,u}^{l'l} \mathbb{P}(\tilde{R}_{i,u}^{l'} = r) + \sum_{\substack{l' \in \mathcal{L}_{i,u}: \\ \lambda_{i,u}^{l'} = 0}} p_{i,u}^{l'l} \mathbb{P}(\tilde{R}_{i,u}^{l'} = r+1) + \\
& + \sum_{\substack{k, \xi_u: \\ i=d_k, \xi_u, u \\ r=a_k-d_k}} \mathbb{P}(\tilde{D}_u = k, P_u = \xi_u | L_{i+1,u} = l), \quad (5.13)
\end{aligned}$$

starting with

$$\begin{aligned}
\mathbb{P}(\tilde{R}_{s_u,u}^l = 0) &= \mathbb{P}(X_{s_u,u} = 0 | L_{s_u,u} = l), \\
\mathbb{P}(\tilde{R}_{s_u,u}^l = \infty) &= 1 - \mathbb{P}(X_{s_u,u} = 0 | L_{s_u,u} = l).
\end{aligned}$$

Note that delay Q_{u+1} is caused only by vehicles of lane u . Thus, for $\delta > 0$,

$$\mathbb{P}(Q_{u+1} = \delta) = \sum_{i,l:\lambda_{i,u}^l > 0} \mathbb{P}NA_{i,l,u}^f \mathbb{P}(\tilde{R}_{i,u}^l = s_{u+1} + \delta - 1 - i) + \sum_{\substack{\xi_u, k: \\ a_k, \xi_u, u+1 = \\ = s_{u+1} + \delta}} \mathbb{P}NA_{k, \xi_u, u}^d.$$

The distribution of $R_{i,u}^l$ can be found as in (5.13) starting with $\mathbb{P}(R_{s_u,u}^l = \delta) = \mathbb{P}(X_{s_u,u} = 0 | L_{s_u,u} = l) \mathbb{P}(Q_u = \delta)$ and $\mathbb{P}(R_{s_u,u}^l = \infty) = 1 - \mathbb{P}(X_{s_u,u} = 0 | L_{s_u,u} = l)$.

5.A.4. Acceleration delay

The expected acceleration delay at lane u is given by the following formula:

$$\mathbb{E}A_u \cdot \mathbb{E}AD_u = \sum_{\substack{i,l,r: \\ s_u \leq i < e_u \\ r < \infty}} r \sigma_{i,u}^l \lambda_{i,u}^l \mathbb{P}(R_{i,u}^l = r) + \sum_k \mathbb{P}(\tilde{D}_u \geq k) (a_k - d_k),$$

where $\mathbb{E}A_u$ is the expected number of arrivals per cycle, $\mathbb{E}AD_u$ is the expected acceleration delay per vehicle, and $\tilde{D}_u = \min\{D_u, n_{P_u, u}\}$. The first summation gives the acceleration delay of **free** departures. Each term of this summation is a multiplication of a possible shift, r (in seconds), the probability of arrival for pair (i, l) , i.e., $\sigma_{i,u}^l \lambda_{i,u}^l$, and the probability of having this departure postponed by r seconds, i.e., $\mathbb{P}(R_{i,u}^l = r)$. The second summation gives the acceleration delay of the **delayed** departures. If there are at least k delayed departures, then the k^{th} delayed departure loses $a_k - d_k$ seconds.

Chapter 6

An approximation model for large urban traffic networks

‘Why do they call it “rush hour” when nothing moves?’

Robin Williams

During morning and evening rush hours, traffic networks experience high arrival rates. Such a period can be too short to reach a stationary state if such a state exists, see [Son et al. \(1995\)](#). Thus, steady-state traffic models may give inaccurate predictions of the system performance. If the traffic control settings are optimised using such steady-state models, it may lead to non-optimal results and, consequently, to poor utilisation of the capacity and long delays. Typically, the network may temporarily operate under *over-saturated conditions*, which means that the vehicle arrival rate exceeds the network capacity, preventing the existence of steady-state solutions. Therefore, we develop a model that captures the time-dependent behaviour of traffic networks and can be used for model-based online traffic control as in [Hoogendoorn et al. \(2007\)](#) and [Kutadinata et al. \(2016\)](#).

There are different approaches for short-time traffic predictions. For example, in [Wang et al. \(2014\)](#) control theory is used, while in [Polson and Sokolov \(2017\)](#) neural networks are applied. In these papers, the predictions of traffic conditions are computed at a *point* in the network based on historical data and current detector information. For urban areas, it is important to provide network-wide estimations. Due to traffic lights in urban networks and a large scale of the problem, these estimations are often based on traffic simulations, see [van Lint and van Hinsbergen \(2012\)](#) and [Laña et al. \(2018\)](#). In this chapter, we develop an analytical model for the transient behaviour of the system by focusing on the evolution of the queues at the intersections. The input of the model consists of the arrival rates, the turning probabilities and the traffic-light settings. We assume

Figure 6.1: A part of the urban traffic network of Enschede with 12 traffic-light and 15 priority intersections. The entrances and the exits of the network with high traffic intensities are marked by circles. The map was obtained using Open-StreetMap data.

that the traffic-light settings are fixed but possibly acyclic, which means that the input consists of the sequences of [phases](#) and their lengths at each intersection for the entire prediction horizon. Note that for acyclic control, the notion of the [offsets](#) is not applicable. The arrival rates and turning probabilities can be estimated using local short-time forecasting approaches. Our model predicts the distribution of the queue length for each lane at each second, while a traffic simulation provides only one realisation per run. The information on the distributions can be used, for example, to obtain the average number of the vehicles in the network or to estimate the blocking probabilities.

In the previous chapters, we analysed small networks with [traffic-light intersections](#). In this chapter, we consider realistic traffic networks consisting of many [traffic-light](#) and [priority intersections](#), see an example in [Figure 6.1](#). Our objective is to achieve a high accuracy but at the same time preserve the model tractability. To increase the accuracy, we add several realistic features to the arrival and service processes compared to [Chapters 4 and 5](#). In particular, we consider random inter-departure and travel times allowing more uncertainty in drivers behaviour. We also provide a safe [time-headway](#) between vehicles by limiting the arrivals at (and departures from) a lane to at most one vehicle per 2 seconds.

In [Chapter 5](#), we characterised the arrivals using a detailed [MAP](#). The size of the MAP primarily depends on the total number of upstream lanes, which leads to

long computation times for large networks. In this chapter, we ensure the model tractability by characterising the arrivals using a small underlying Markov chain that captures the essentials of the arrival process. As a result, the computation time for our model is linear in the number of intersections, which is crucial for network models, see [Osorio and Yamani \(2017\)](#). This allows us to study the influence of rush hours on the system for a real network, see [Figure 6.1](#).

This chapter is structured as follows. In [Section 6.1](#), we describe the model. In [Section 6.2](#), we compare the results of our models with a traffic microsimulation and study the effect of rush hours on the system. In [Section 6.3](#), we conclude the chapter.

6.1. Model

In this section, we describe our modelling approach for an urban traffic network that consists of traffic-light and priority intersections, see, e.g., [Figure 6.1](#). We represent such a network as a queueing network with three types of bottlenecks: traffic-light intersections, priority intersections and lane-changing movements.

As before, we consider a discrete-time model where time is split in seconds. The input of the model consists of the computation horizon, the arrival rates from outside the network in each second, the routing probabilities inside the network and the traffic-light settings for each intersection. In the present model, compared to the steady-state model of the previous chapters, we do not require a cyclic traffic-light control. The output is the evolution of the number of vehicles at each lane, which also yields the total number of vehicles in the entire network.

In [Subsection 6.1.1](#), we present our queueing network model and decomposition approach. Then, we describe the general model for one queue in [Subsection 6.1.2](#). In [Subsection 6.1.3](#), we characterise the arrivals using the MAP. Finally, we give details of the service process for different types of queues in [Subsection 6.1.4](#).

6.1.1. Network model

Based on the traffic network, we construct a queueing network as follows. Each queue corresponds to a certain part of the road with a given travel time. There are four types of queues: traffic-light, low-priority, lane-changing and auxiliary queues.

Each lane at the traffic-light intersection corresponds to a *traffic-light queue*, see [Figure 6.2a, b](#). For a priority intersection, we assume that the vehicles at the major road have priority and pass the intersection without stopping. For these vehicles, the priority intersection is an intermediate destination point. Therefore, we represent each major lane by an *auxiliary queue* that has arrivals, but always remains empty as the arriving vehicles continue without stopping. Note that these vehicles can be delayed later at the next lane by slow vehicles from the minor road. We use a separate auxiliary queue for each direction at the major road,

Figure 6.2: Parts of the traffic network and the corresponding queues: a) and b) a split of a road before an intersection, c) and d) a priority intersection, e) and f) a multi-lane road where lane-changing is allowed.

as only straight-going vehicles interact with the vehicles from the minor road. Each lane at the minor road is represented by a *low-priority queue*, which has a common server with the auxiliary queue of straight-going vehicles at the major road, see Figure 6.2c, d. We use an auxiliary queue also for each lane that exits the network, see Figure 6.2d, where we added a queue for the vehicles that turn from the major road to the minor road and leave the system. Finally, we consider a road with multiple lanes upstream of an intersection, where lane changing is allowed. Compared to the previous cases, there is no physical queue of completely stopped vehicles. The lane-changing vehicles slow down and wait for a gap in the arrivals at the destination lane while moving along the original lane. We combine these vehicles in a (virtual) *lane-changing queue*. In this way, each lane is represented by a pair of a lane-changing and an auxiliary queue with a common server. In this case, the auxiliary queue corresponds to the stream of the vehicles that do

not change the lane, see Figure 6.2e, f.

The resulting network consists of many inter-dependent queues. To avoid the *curse of dimensionality*, we decompose the network into individual queues and consider the evolution of these queues. The dependency between the queues is taken into account in the arrival and service processes of the queues.

For each queue, the arrivals are modelled by a discrete-time MAP. The arrival process for an *external queue*, i.e., a queue without upstream queues, is computed based on given (time-dependent) arrival rates. An *internal queue* is a queue inside the network, and it has upstream queues. Its arrival process depends on the output of the upstream queues, the travel time, the routing probabilities and the acceleration of the vehicles, see Subsection 6.1.3.

Each queue in the network has a ‘server’, the state of which defines the departures and is governed by another Markovian process. For a traffic-light queue, the state depends on whether the traffic light for this lane is red or green and, in the latter case, also on how long the light has already been green. For low-priority and lane-changing queues, the server state depends on the arrivals at the corresponding auxiliary queues, see Subsection 6.1.4.

The total number of vehicles in the system consists of the number of travelling and queued vehicles. Therefore, we keep track not only of the lengths of the queues and the states of the arrival and service processes in any particular second but also of the vehicles travelling towards each queue. The travelling vehicles are the vehicles that departed from the upstream queue but have not yet arrived at the downstream queue. The arrival process at the queue is fully known for a time period of length t ahead, where t is the travel time of the corresponding road under the *free-flow speed*. Due to acceleration of the vehicles, they leave the intersections with different speeds, which means that some vehicles depart slowly and have longer travel times. As the results, there may be some partial information of the arrivals after the following t seconds. For each second of the computation horizon, we update the queues and the arrival and service processes, see the detailed description below.

6.1.2. General model for one queue

In this subsection, we describe the queue evolution based on the arrival and service processes. Let X_i be the queue length at the beginning of second i , and let L_i (resp., R_i) be the state of the arrival (resp., service) process at the beginning of second i . In the sequel, whenever we write ‘second i ’ we mean the beginning of second i , unless stated otherwise.

The states of arrival and service processes at second i determine the number of arrivals, Y_i , and *possible* departures, Z_i , from the queue. In particular, an arrival state either generates an arrival with probability 1 or does not generate an arrival, which means that $\mathbb{P}(Y_i = 1|L_i = l) = 1 - \mathbb{P}(Y_i = 0|L_i = l) \in \{0, 1\}$ for each possible l . Similarly, $\mathbb{P}(Z_i = 1|R_i = r) = 1 - \mathbb{P}(Z_i = 0|R_i = r) \in \{0, 1\}$ for each possible r .

We assume that the arrivals in a second occur before the departures in that same second. Thus, given values of Y_i and Z_i , the queue length at second $i + 1$ is determined by the following Lindley-type equation:

$$X_{i+1} = [X_i + Y_i - Z_i]^+, \quad (6.1)$$

where $x^+ = \max\{x, 0\}$. Note that random variable Z_i refers to the possibility of a departure and not to the actual number of departures. Thus, it is possible that $X_i + Y_i - Z_i = -1$, which occurs when $X_i = Y_i = 0$ and $Z_i = 1$.

For computations, we use the joint distribution of X_i , L_i and R_i . Due to the relation between Y_i , Z_i and L_i , R_i and equation (6.1), X_{i+1} is a function of X_i , L_i and R_i . The arrival state L_{i+1} depends only on L_i while the service state R_{i+1} depends both on R_i and on whether the event $X_i + Y_i - Z_i = 0$ occurred. The event $X_i + Y_i - Z_i = 0$ occurs in three different cases. The first case is that the queue clears, which corresponds to $X_i = 1$, $Y_i = 0$ and $Z_i = 1$. This case relates to the transition from *delayed* to *free* departures similar to the model of [Chapters 4 and 5](#), see [Subsection 6.1.4.1](#) below. The second case, i.e., $X_i = 0$, $Y_i = 1$ and $Z_i = 1$ corresponds to an arrival and immediate departure of a vehicle. In these two cases, a departure occurs, which means that there will be no departure next second. The last possibility is that the queue remained empty during this second and there was no possibility for departure, which means that $X_i = Y_i = Z_i = 0$. The latter event is of importance in the service processes of low-priority and lane-changing queues, see [Subsections 6.1.4.2 and 6.1.4.3](#) below. We use the following notation:

$$\begin{aligned} \mathcal{L}_i: & \quad \text{the state space of the arrival process at second } i, \\ A_i^l(z) &= \mathbb{E}(z^{Y_i} | L_i = l), \\ p_i^{ll'} &= \mathbb{P}(L_{i+1} = l' | L_i = l), \\ \mathcal{R}_i: & \quad \text{the state space of the service process at second } i, \\ B_i^{r'}(z) &= \mathbb{E}(z^{-Z_i} | R_i = r), \\ q_i^{r'r'} &= \mathbb{P}(R_{i+1} = r' | R_i = r, X_i + Y_i - Z_i \neq 0), \\ \dot{q}_i^{r'r'} &= \mathbb{P}(R_{i+1} = r' | R_i = r, X_i + Y_i - Z_i = 0), \\ X_i^{l,r}(z) &= \mathbb{E}(z^{X_i} \mathbb{1}\{L_i = l, R_i = r\}), \\ w_i^{l,r} &= \mathbb{P}(X_i + Y_i - Z_i = -1, L_i = l, R_i = r), \\ v_i^{l,r} &= \mathbb{P}(X_i + Y_i - Z_i = 0, L_i = l, R_i = r). \end{aligned}$$

In terms of the functions $X_i^{l,r}(z)$, (6.1) can be written as follows:

$$\begin{aligned} X_{i+1}^{l,r}(z) = & \sum_{l' \in \mathcal{L}_i} p_i^{ll'} \sum_{r' \in \mathcal{R}_i} q_i^{r'r'} \left[X_i^{l',r'}(z) A_i^{l'}(z) B_i^{r'}(z) - v_i^{l,r} + w_i^{l,r} \left(1 - \frac{1}{z} \right) \right] + \\ & + \sum_{l' \in \mathcal{L}_i} p_i^{ll'} \sum_{r' \in \mathcal{R}_i} \dot{q}_i^{r'r'} v_i^{l,r}. \end{aligned}$$

This equation is similar to equations (2.21), (4.18) and (4.19). Here, the term $X_i^{l',r'}(z) A_i^{l'}(z) B_i^{r'}(z)$ is equal to $z X_i^{l',r'}(z)$ if there is an arrival and no departures,

Figure 6.3: A part of the MAP for an **external lane**. The circles represent the states of the Markov chain, and the arrows show the transitions. Each column of states corresponds to one second.

or to $X_i^{l',r'}(z)/z$ if there is a departure but no arrivals, or to $X_i^{l',r'}(z)$ in the other cases. The term $w_i^{l,r}(1 - 1/z)$ corresponds to the event of an empty queue with no arrivals but with a possibility for a departure. The probability $v_i^{l,r}$ is used to distinguish the cases $X_i + Y_i - Z_i = 0$ and $X_i + Y_i - Z_i \neq 0$.

Thus, we compute the evolution of the queue length based on the initial queue X_0 and the processes L_i and R_i , $i = 0, 1, \dots$, which are constructed dynamically based on the state of the other queues.

6.1.3. Arrival process

In this subsection, we describe the arrival process of a queue. As in [Chapters 4 and 5](#), we use a MAP. However, to enhance the computation speed comparing to the model of [Chapters 4 and 5](#), we restrict the state space of the underlying Markov chain to at most 4 states. To make the arrival process more realistic, we do not allow more than one arrival per two seconds, which corresponds to the saturation rate of $1800 \text{ veh/hour} = 0.5 \text{ veh/sec}$ used in the literature, see, e.g., [Lo \(1999\)](#).

For an **external queue**, we construct the arrival process as follows. We use two states: a waiting state w and an arrival state α with $\mathbb{P}(Y_i = 1 | L_i = w) = 0$ and $\mathbb{P}(Y_i = 1 | L_i = \alpha) = 1$. The allowed transitions are $w \rightarrow w$, $w \rightarrow \alpha$ and $\alpha \rightarrow w$, see [Figure 6.3](#). Transition $\alpha \rightarrow \alpha$ would violate our assumption that there is at most one arrival per two seconds. This Markov chain is completely determined by the probabilities of the arrival state at second i , which is equal to the arrival rate λ_{i-t} at second $i-t$, where t is the travel time of the corresponding lane under the **free-flow speed**. If the travel time (in seconds) is not an integer, we assume that it is a random variable with support $[t]$ and $[t]$ and average t . In this case, the probability of an arrival at the queue at second i is equal to $([t] - t)\lambda_{i-[t]} + (t - [t])\lambda_{i-[t]}$. For the case of heterogeneous traffic, one can choose a different distribution of the travel time with a bigger support.

For an **internal queue**, the arrivals depend on the departures from the upstream queues. If the upstream queues correspond to lanes at a **traffic-light intersection**, the arrivals form **platoons**, which we model using states b , w , α and e . Here, states b and e correspond to the time before and after the arrivals from one lane. In addition to previous transitions, only transitions $b \rightarrow b$, $b \rightarrow \alpha$, $\alpha \rightarrow e$ and $e \rightarrow e$ are allowed for one platoon, see [Figure 6.4](#). The first arrival of the next platoon

Figure 6.4: A part of the MAP for one platoon. The circles represent the states of the Markov chain, and the arrows show the transitions. Each column of states corresponds to one second.

may occur at second $s + \lfloor t \rfloor$, where s is the beginning of the green time at the corresponding upstream lane. To switch between platoons, the other transitions are also allowed at second $s + \lfloor t \rfloor - 1$ with the exception of transition $a \rightarrow a$. If the upstream queues do not correspond to lanes at a traffic-light intersection, states b and e are not used as the arrivals can be at any time. For all cases, the transition probabilities are computed using the output of the upstream queues.

In the remainder of this subsection, we construct the arrival process for an internal queue. For each departure from an upstream queue, we update the arrival process at the considered downstream queue. In what follows, we fix a pair of upstream and downstream queues.

For each pair (l, r) of arrival and service state that generates a departure at the upstream queue, there is a distribution of the [acceleration delay](#) and information about previous departures from the lane. For a given acceleration delay δ , the travel time of the departed vehicle to the downstream queue is equal to $\lfloor t \rfloor + \delta$ or $\lceil t \rceil + \delta$, where t is the travel time under the free-flow speed. The actual travel time may be longer if the last arrived vehicle was slower. Based on the information of the last arrival, we find the arrival moments corresponding to the pair (l, r) and update the arrival process accordingly. In this way, we store the (partial) information of the future arrivals at the queue.

Remark 6.1 (Acceleration delay). Acceleration delays depend on the maximum allowed speed at the downstream lane. Using the approach of [Chapter 4](#), we find these delays for the speed of 60 km/hour and use these values to compute the acceleration delays for other speeds. Note that in this chapter, the acceleration delay is included in the arrival process at the downstream lane, while in [Chapters 4](#) and [5](#), we found the acceleration delay during the analysis of the upstream lane.

Consider a pair $(\bar{l}, \bar{r})$ of the arrival state and the service state at second $\bar{t}$ at the upstream queue with $\mathbb{P}(Z_{\bar{t}} = 1 | R_{\bar{t}} = \bar{r}) = 1$. We find the corresponding arrivals at the downstream queue using information of the last departure from the upstream lane. We use the following notation, in which, for convenience, we do not include indices for $\bar{l}, \bar{r}, \bar{t}$:

γ :	the probability that vehicles departing from the upstream queue arrive to the considered downstream queue,
$\beta_{t'}$:	the probability that the travel time is equal to t' ,
α	$= \mathbb{P}\{X_{\bar{t}} + Y_{\bar{t}} > 0, L_{\bar{t}} = \bar{l}, R_{\bar{t}} = \bar{r}\},$
$\theta_{i,l,r}$:	the probability that the last departure from the upstream queue was at second i for $R_i = r$ and $L_i = l$,
Θ :	the set of all triples (i, l, r) such that $\theta_{i,l,r} > 0$,
$\omega_{i,l,r}^{i',l'}$:	the probability that departure at second i with $R_i = r$ and $L_i = l$ corresponds to an arrival state l' at second i' at the downstream queue,
$\Omega_{i,l,r}$:	the set of pairs (i', l') such that $\omega_{i,l,r}^{i',l'} > 0$,
η_i :	the probability that the last arrival from another upstream lane was at second i ,
H :	the set of all i with $\eta_i > 0$.

Probability α is the probability of a departure for $L_{\bar{t}} = \bar{l}, R_{\bar{t}} = \bar{r}$. For non-traffic-light queues, we set $H = \emptyset$. For traffic-light queues, by last departure in the definition of $\theta_{i,l,r}$ we mean the last departure during the current green time. Note that with probability $\bar{\theta} = 1 - \sum_{(i,l,r) \in \Theta} \theta_{i,l,r}$, the departure generated by pair of states $L_{\bar{t}} = \bar{l}, R_{\bar{t}} = \bar{r}$ is the first departure.

Fix one last departure and the corresponding arrival state, i.e., fix $(i, l, r) \in \Theta$, $(i', l') \in \Omega_{i,l,r}$. Consider some t' with $\beta_{t'} > 0$. Note that $t' = \lfloor t \rfloor + \delta$ or $t' = \lceil t \rceil + \delta$, where δ is a possible value of the acceleration delay for service state $\bar{r}$. The arrival generated by pair $L_{\bar{t}} = \bar{l}, R_{\bar{t}} = \bar{r}$ occurs at second $\max\{i' + 2, i + t' + \delta\}$ with probability $\theta_{i,l,r} \omega_{i,l,r}^{i',l'} \beta_{t'}$. In the case that the considered departure is the first one, the arrival occurs at second $\max\{i' + 2, i + t' + \delta\}$ with probability $\bar{\theta} \eta_{i'} \beta_{t'}$ and at second $i + t' + \delta$ with probability $\bar{\theta}(1 - \sum_{i'} \eta_{i'}) \beta_{t'}$. The probability that the arrival is generated is equal to $\alpha \gamma$. With probability $\alpha(1 - \gamma)$, a gap in the arrivals is generated.

Given the second i'' of the arrival, the arrival process is updated as follows. We increase the flow of probabilities on the path $L_{i'} = l', L_{i'+1} = \tilde{l}, \dots, L_{i''-1} = \tilde{l}, L_{i''} = \alpha$ by the probability that there is an arrival and on path $L_{i'} = l', L_{i'+1} = \tilde{l}, \dots, L_{i''-1} = \tilde{l}, L_{i''} = \tilde{l}$ by the probability that there is no arrival. Here i'' is the second of the last arrival and l' is the corresponding state of the arrival process, state $\tilde{l}$ is an intermediate state, which coincides with l' if $l' \neq \alpha$ and equal to α if $l' = \alpha$. In the case that the departure is the first one, we set $i' = i + \lfloor t \rfloor$ and $l' = \mathfrak{b}$ for a traffic-light upstream queue and $l' = \mathfrak{w}$ otherwise.

In the case that there is no full information at second i about arrivals at second $i + \lfloor t \rfloor$, we add transitions $\mathfrak{b} \rightarrow \mathfrak{b}$, $\mathfrak{w} \rightarrow \mathfrak{w}$, $\alpha \rightarrow \mathfrak{w}$ and $\mathfrak{e} \rightarrow \mathfrak{e}$ at second $i + \lfloor t \rfloor - 1$ with the rest of transition probability. Transitions to state $\mathfrak{e}$ are added at the first second of yellow time at the upstream queue.

Remark 6.2 (On network topology). In the model of Chapters 4 and 5, we were interested in the steady state. There, we computed the arrival process during a cycle for each queue. This arrival process depends on the steady state of all

upstream queues. For this reason, we considered only acyclic networks. In this chapter, we focus on the transient behaviour. Thus, though the arrival process still depends on all upstream queues, it depends only on the past state of the system. Therefore, there are no restrictions on the topology of the network.

Remark 6.3 (On arrival rate). In the model of Chapters 4 and 5, the states of MAP may have an arrival rate which is not equal to 0 or 1. There, we modelled the effect of the vehicles that catch up with the leading vehicle by collapsing states with arrival rate 0. In the model of this chapter, such an approach would lead to very dense arrivals. Therefore, we added the assumption that there is not more than one arrival per two seconds and an arrival occurs not earlier than second $i' + 2$, where i' is the second of the last arrival.

6.1.4. Service process

The departures are governed by the state R_i of the server. There are two types of states: departure, i.e., states r with $P(Z_i = 1 | R_i = r) = 1$, and non-departure states, i.e., $P(Z_i = 1 | R_i = r) = 0$. In the following subsections, we consider the service processes for [traffic-light](#), [low-priority](#) and [lane-changing](#) queues. For these queues, we describe possible states and their transitions. The service process for an [auxiliary queue](#) is generated by one departure state for each second.

6.1.4.1. Traffic-light intersections

In this subsection, we consider a lane at a traffic-light intersection. The departures from this lane depend on the traffic light and are only possible during the green time and the first yellow second.

Consider one green time of a fixed length. Let s (resp., e) be the first second of this green time (resp., the following yellow time). We distinguish [delayed](#) and [free departures](#). The delayed departures are departures of the vehicles that have joined the queue. We assume that the k^{th} delayed departure during this green time occurs at second $s + P_k$, where P_k is a random departure time of the k^{th} delayed vehicle, and variables P_k form a Markov chain, i.e., P_{k+1} depends only on P_k . This means that the behaviour of a vehicle in the queue is only influenced by its position in the queue and the vehicle directly in front of it, which is similar to the car-following models. The distributions of P_1 and of P_{k+1} given P_k are estimated using a traffic microsimulation, similar to [Chapter 4](#). A free departure occurs when a vehicle arrives at a lane during the green time, finds no queue and proceeds without stopping. This vehicle can be delayed by one second in case there was a departure just before its arrival.

To model such departures, we use states (j, k) , $\mathbf{d}$, $\mathbf{n}$ and $\mathbf{r}$, where $k = 1, 2, \dots$, $j = s, \dots, e$, see an example in [Figure 6.5](#). State (j, k) corresponds to the event that the next departure is the k^{th} delayed departure at second j , i.e., $s + P_k = j$. In each second, there are only a few possible states, because each departure moment

Figure 6.5: An example of the service-state process for a traffic-light queue for $s = 2$, $e = 8$ and seconds $i = 0, \dots, 9$. The dotted arrows show transitions in the case of the event $X_i + Y_i - Z_i = 0$, while solid arrows give transitions when $X_i + Y_i - Z_i \neq 0$. The departure states are green.

P_k has a support of 3 – 4 seconds according to the data of [Chapter 4](#), see, e.g., [Figure 4.11](#). Clearly, the departure at state $R_i = (j, k)$ is only possible if $j = i$. For these states, the transition probabilities $q_i^{rr'}$ can be found from the conditional distributions of P_{k+1} given P_k . For the case that there is no queue during a green time period, there are two states: **d** and **n**, which correspond to departure and non-departure states, respectively. The non-departure state **n** provides a gap between departures after the queue has cleared. Thus, $\hat{q}_i^{rn} = 1$ for all departure states r . The only transition from state **n** is to state **d**. For state **d**, the event $X_i + Y_i - Z_i \neq 0$ corresponds to the event that the queue was empty and there were no arrivals, in which case the only transition is to state **d**. The last state is a non-departure state **r** that is the only state during the red time.

The correct [capacity](#) of the lane is modelled by reducing the probability of departure states at the first yellow second to fit the capacity curve found in [Chapter 4](#), see [Figure 4.12](#) there. For this second, we merge all departure states in state **d** and all non-departure states in state **n**, see second $i = 8$ in [Figure 6.5](#).

Remark 6.4 (Service process in the model of [Chapters 4](#) and [5](#)). Note that the service process of the model analysed in [Chapters 4](#) and [5](#) can be also described using server states. During the green time, there would be one state for each realisation ξ of P and a separate state for free departures. In this chapter, we consider a more realistic service process with random inter-departure times. In this case, the transitions from state **r** in the end of the red time are independent of the length of the following green time, which is not the case for the model in

Chapters 4 and 5.

6.1.4.2. Priority intersections

In this subsection, we consider a [priority intersection](#). In what follows, we use as an example a T-intersection where a minor road joins the major road with higher priority, see [Figure 6.2c](#). For such an intersection, we focus on the queue at the minor road and construct its service process using the arrival process at the major road. For simplicity, we suppose that the major road is a one-way road. The model for a road with both directions is similar. In that case, the departures from a [low-priority](#) queue would be influenced by the arrivals at multiple [auxiliary queues](#).

We suppose that the priority intersection has a ‘yield’ sign. As the result of a ‘yield’ sign, the vehicles at the minor road do not need to stop, but instead they slow down to check the conditions and leave if there are no arrivals from the major road. Due to deceleration, the slowed-down vehicles arrive at the intersection later than they would arrive at the [free-flow speed](#). We model it by adding an extra 2 seconds delay before vehicles can join the queue, which is done by increasing the travel time at the minor lane. In what follows, by the departure second of a vehicle from the minor lane we mean such second i that an arrival at the major road at second i would make the vehicle stop and wait, but an arrival at second $i + 1$ or later would not.

Remark 6.5 (Model for ‘stop’ sign and unprotected turns). A priority intersection with a ‘stop’ sign can be modelled in the same way. In this case, one would need fewer service states than described below. Moreover, a similar model can be used in the case of a traffic-light intersection with an [unprotected](#) left or right turn, i.e., when turning vehicles need to yield to the conflicting traffic that has green time at the same time.

To model the different speeds of departing vehicles, we use three departure states: $\mathbf{d}_q$, $\mathbf{d}_s$ and $\mathbf{d}_c$ for quick, slow and completely stopped vehicles, respectively. Here, a quick vehicle is a vehicle that only slows down a little bit and, thus, leaves the queue with relatively high speed (≈ 8 m/sec). A slow vehicle slows down one second longer and, thus, leaves with low speed (≈ 4 m/sec). The stopped vehicle needed to stop due to an arrival at the major lane and departs at a very low speed (< 0.5 m/sec). After departure these vehicles have extra [acceleration delay](#) of 1–2, 5–6 or 8–9 seconds, respectively.

If there is an arrival at the major road, the state of the server changes to one of three arrival states: $\mathbf{a}_e$, $\mathbf{a}_i$ and $\mathbf{a}_d$. State $\mathbf{a}_e$ corresponds to an arrival when there is no queue at the minor lane and the last departure was a long time ago. State $\mathbf{a}_i$ represents the case when the last gap between arrivals was small for a departure from the minor road and the waiting drivers got impatient. State $\mathbf{a}_d$ is used when the gap was sufficient for a departure. In this case, the arrival is delayed. Except departure and arrival states, we use auxiliary states to model the time gap

The probability that there is an arrival at the major road at the next second is computed as follows. We store conditional probabilities $\mathbb{P}(\bar{L}_i = l | R_i = r)$, where $\bar{L}_i$ is the state of the arrival process at the auxiliary queue. Then, we compute probability $\mathbb{P}(\bar{L}_{i+1} = a | R_i = r)$ as follows:

$$\begin{aligned} \mathbb{P}(\bar{L}_{i+1} = a | R_i = r) &= \sum_l \mathbb{P}(\bar{L}_{i+1} = a | R_i = r, \bar{L}_i = l) \mathbb{P}(\bar{L}_i = l | R_i = r) \\ &= \sum_l \mathbb{P}(\bar{L}_{i+1} = a | \bar{L}_i = l) \mathbb{P}(\bar{L}_i = l | R_i = r), \end{aligned} \quad (6.2)$$

where we use the fact that the state of the arrival process only depends on the state at the previous second. Using this probability, we find the transition probabilities $q_i^{rr'}$ and $\dot{q}_i^{rr'}$ of the service process in the following way. We know the transition probabilities in the cases that there is an arrival or that there is no arrival at the auxiliary queue during the following second, see Figure 6.6. Let $q_{i,a}^{rr'}$, $q_{i,n}^{rr'}$, $\dot{q}_{i,a}^{rr'}$ and $\dot{q}_{i,n}^{rr'}$ denote these probabilities, where indexes a and n correspond to the events of an arrival and no arrival, respectively, and let C_i , $\bar{C}_i$ denote the events $X_i + Y_i - Z_i \neq 0$ and $X_i + Y_i - Z_i = 0$, respectively. We find the probability $q_i^{rr'}$ by conditioning on the events $\bar{L}_{i+1} = a$ and $\bar{L}_{i+1} \neq a$:

$$\begin{aligned} q_i^{rr'} &= \mathbb{P}(R_{i+1} = r' | R_i = r, C_i) \\ &= q_{i,a}^{rr'} \mathbb{P}(\bar{L}_{i+1} = a | R_i = r, C_i) + q_{i,n}^{rr'} \mathbb{P}(\bar{L}_{i+1} \neq a | R_i = r, C_i) \\ &\approx q_{i,a}^{rr'} \mathbb{P}(\bar{L}_{i+1} = a | R_i = r) + q_{i,n}^{rr'} (1 - \mathbb{P}(\bar{L}_{i+1} = a | R_i = r)), \end{aligned} \quad (6.3)$$

where we estimated the probability $\mathbb{P}(L_{i+1} = a | R_i = r, C_i)$ as $\mathbb{P}(L_{i+1} = a | R_i = r)$ since, for many service states, the event $\bar{C}_i$ is impossible as $Z_i = 0$ and $X_i > 0$, see Figure 6.6, meaning that $\mathbb{P}(\bar{L}_{i+1} = a | R_i = r, C_i) = \mathbb{P}(\bar{L}_{i+1} = a | R_i = r)$. When both events C_i and $\bar{C}_i$ are possible, we suppose that the dependency of $\bar{L}_{i+1}$ given R_i on them (through service process $R_0, \dots, R_{i-1}$) is negligible. The probabilities $\dot{q}_i^{rr'}$ we find in the same way as $q_i^{rr'}$ by changing C_i to $\bar{C}_i$ and $q_{i,a}^{rr'}$, $q_{i,n}^{rr'}$ to $\dot{q}_{i,a}^{rr'}$, $\dot{q}_{i,n}^{rr'}$ in (6.3).

The remaining step is to estimate the probabilities $\mathbb{P}(\bar{L}_{i+1} = l' | R_{i+1} = r')$ using the probabilities $\mathbb{P}(\bar{L}_{i+1} = l' | R_i = r)$, which can found as in (6.2). Let us first consider the case $l' = a$:

$$\begin{aligned} \mathbb{P}(\bar{L}_{i+1} = a | R_{i+1} = r') &= \mathbb{P}(R_{i+1} = r', \bar{L}_{i+1} = a) / \mathbb{P}(R_{i+1} = r') \\ &= \sum_r q_i^{rr'} \mathbb{P}(\bar{L}_{i+1} = a | R_i = r, C_i) \mathbb{P}(R_i = r, C_i) / \mathbb{P}(R_{i+1} = r') + \\ &\quad + \sum_r \dot{q}_i^{rr'} \mathbb{P}(\bar{L}_{i+1} = a | R_i = r, \bar{C}_i) \mathbb{P}(R_i = r, \bar{C}_i) / \mathbb{P}(R_{i+1} = r') \\ &\approx \sum_r q_{i,a}^{rr'} \mathbb{P}(\bar{L}_{i+1} = a | R_i = r) \mathbb{P}(R_i = r, C_i) / \mathbb{P}(R_{i+1} = r') + \end{aligned}$$

Figure 6.7: Lane-changing movement. A vehicle from lane 1 wants to change to lane 2. It decelerates and find the first gap in the arrivals. During this time it is in the virtual queue.

$$+ \sum_r \dot{q}_{i,a}^{rr'} \mathbb{P}(\bar{L}_{i+1} = a | R_i = r) \mathbb{P}(R_i = r, \bar{C}_i) / \mathbb{P}(R_{i+1} = r'),$$

where, as before, we assumed that $\bar{L}_{i+1}$ given R_i is independent of the events C_i and $\bar{C}_i$. When $l' \neq a$, we can compute the probabilities in the same way by changing $q_{i,a}^{rr'}$ and $\dot{q}_{i,a}^{rr'}$ to $q_{i,n}^{rr'}$ and $\dot{q}_{i,n}^{rr'}$, respectively.

6.1.4.3. Lane changes

In this subsection, we consider the virtual queue formed due to lane changes. Consider two lanes and vehicles that want to change from lane 1 to lane 2, see Figure 6.7. The vehicles from lane 1 can change immediately if there are no arrivals at lane 2 at the same moment. If there are arrivals, a vehicle slows down and picks the first gap of at least 2 seconds between arrivals. During this time, the vehicle is in the virtual queue. This time is equal to the delay of the vehicle. Instead of considering the actual movement of the vehicle from lane 1, we assume that the virtual queue is at a certain place of the lane. The vehicles are waiting in it and when there is a sufficient gap at lane 2, i.e., at least 2 seconds without arrivals, they leave the queue with the *free-flow speed*. The vehicles at lane 2 may be delayed by one second if a vehicle from lane 1 changes the lane just in front of them, see Figure 6.7.

We model this service process similar to the case of low-priority queue but with fewer states, see Figure 6.8. There are two arrival and departure states. Arrival states $\mathbf{a}_d$ and $\mathbf{a}_n$ correspond to delayed and non-delayed arrivals. An arrival at the *auxiliary queue* is delayed when it arrives just after the departure from the *lane-changing queue*. Departure states $\mathbf{d}_e$ and $\mathbf{d}_q$ correspond to an empty and non-empty queue at the beginning of the second. For state $\mathbf{d}_e$, if there are no arrivals at the queue, $X_i + Y_i - Z_i = -1$, and there is a transition $\mathbf{d}_e \rightarrow \mathbf{d}_e$. Otherwise, the only transition is to state $\mathbf{n}$ which represents the gap between arrivals and departures. State $\mathbf{b}$ provides a bigger gap before the next departure if the previous arrival at the destination lane was delayed.

Figure 6.8: Service states for a lane-changing queue and its transitions between arrivals at the destination lane. The dotted and solid arrows show transitions for the events $X_i + Y_i - Z_i = 0$ and $X_i + Y_i - Z_i \neq 0$, respectively. If there is an arrival at the next second, then there is a transition to one of the arrival states a_d, a_n . In this case, states d_e, d_q, b have a transition to state a_d , and state n to state a_n . These transitions are excluded from the figure for the sake of clarity.

6.2. Numerical results

In this section, we present our numerical results. The network and input parameters are described in Subsection 6.2.1. For this network, we check the accuracy of our model in Subsection 6.2.2 and study the impact of the length and intensity of a rush hour on the system in Subsection 6.2.3.

6.2.1. Network

In this subsection, we describe the network and the input of the model. We use a part of the traffic network of Enschede with 12 traffic-light and 15 priority intersections, see Figure 6.1. For each pair of points at the edge of the network, we find the shortest route between these points. The number of vehicles that choose a particular route is proportional to the weight of the route which, in turn, depends on the origin and destination of the route. We distinguish major and minor entry points, see Figure 6.1. The weight of a route is 1, 12 or 144 if it connects two minor points, one minor and one major or two major points, respectively.

We decompose the network into 153 queues according to Subsection 6.1.1. The routing probabilities between the queues are calculated based on the routes and their weights. The arrival rate $\lambda_{i,j}$ at an external queue j at second i is proportional to the total weight w_j of the routes starting at this queue and a time-dependent parameter ρ_i called *arrival-rate intensity*, i.e., $\lambda_{i,j} = \rho_i w_j / w_{\text{crit}}$. The value of w_{crit} is chosen such that the arrival-rate intensity ρ_i is equal to the load at the busiest intersection in the network.

Figure 6.9: The average number of vehicles in the system as a function of time for a) $\rho = 0.33$ and b) $\rho = 1.66$.

The traffic lights operate under fixed control. The [cycle](#) is equal to 90 seconds, the total all-red time is equal to 10 seconds, and the green times are chosen according to the discrete-time alternative of the Webster formula, see [Chapter 5](#). The traffic-light settings are the same for all results below.

6.2.2. Comparison to a microsimulation

In this subsection, we compare the results of our model with traffic microsimulation [SUMO](#). We compute the evolution of the number of vehicles in the system for 20 minutes ahead, see [Figure 6.9](#). The arrival rates are positive and constant for the first 400 seconds, i.e., $\rho_i = \rho$ for $i = 0, \dots, 399$ and $\rho_i = 0$ for $i = 400, \dots, 1199$. SUMO results were obtained using 200 simulations.

Our approximate model quite accurately predicts the number of vehicles in the system. For $\rho = 0.33$, the relative difference is around 8% when the system is busy. When the system is almost empty the absolute difference is not more than 1 vehicle. For $\rho = 1.66$, the average relative difference for a busy system grows to 13%. The underestimation of our model is the result of the approximation. In particular, the time of an arrival depends only on the previous arrival and not on the history before. Thus, as in the model of [Chapters 4 and 5](#), there is a certain independence between cycles, which leads to an underestimation for high loads. Also our model does not include the effect of blocking, which often leads to increased number of vehicles in the system. Despite these effects, our model gives a fair estimation of the system behaviour.

6.2.3. Rush hour impact

In this subsection, we study the impact of rush hours on the system behaviour. We consider piecewise-constant arrival rates with a period of high arrival rate.

In particular, there is a warm-up period of 10 minutes with $\rho_i = 0.1$, a period of length t minutes with $\rho_i = 0.1 + \rho_{extra}$, after which the arrival rate intensity changes back to $\rho_i = 0.1$, see Figure 6.10. In this way, we model a rush hour when a large number of vehicles arrives in the system during a short time period.

We focus on the effect of time period t for the fixed number of extra vehicles during the rush hour, i.e., the product $\rho_{extra}t$ is fixed. In this way, we model the effect of spreading the traffic intensity during the rush hour. We use a 100-minute horizon and let $\rho_{extra}t = 63$, which gives an average of around 2200 arrivals in the system during the entire prediction horizon. For $t = 20, 40, 60, 80$ minutes, we consider the number of vehicles in the system, the departure rate and the *congestion time*, i.e., time before the system returns to the steady state for the arrival intensity $\rho_i = 0.1$.

In Figure 6.11, we plot the total number of vehicles in the system as a function of time. Here and later, we use the values that are averaged over a cycle to remove the oscillation due to cyclic traffic-light control. Note that only for $t = 80$, the arrival-rate intensity is lower than 1. Thus, it is the only case where the system reaches steady-state during the period of high arrival rate. In the other cases, the system is over-saturated during this period. In Figure 6.11, we also plot the total number of travelling, i.e. not-queued, vehicles. For $t = 20$, only 10% of the vehicles in the system are travelling, while for $t = 80$, it is about 40%. The absolute values show the opposite trend, i.e., about 120 vehicles are travelling for $t = 20$ and only 60 for $t = 80$. Note that due to differences between *vertical* and *horizontal* queues, the real number of vehicles that are currently travelling between the queues is smaller.

In Figure 6.12, we plot the arrival and departure rates as functions of time. At the end of the warm-up period, the departure rate equals the arrival rate which means that the system is in a steady-state. During the period of high arrival rate, the departure rate converges to a certain value which is smaller than the arrival rate for $t = 20, 40, 60$. This limiting rate is higher for smaller values of t . The reason for this is that, for smaller values of t , more intersections are working in oversaturated conditions and have a higher departure rate. After the period of high arrival rate has stopped, the departure rate returns to the steady-state values for $\rho_i = 0.1$. Note that for $t = 20, 40, 60$, the systems returns to the steady state at the same time, i.e., at around 90 minutes (5400 seconds). This means, that the *congestion time* is independent of t for oversaturated conditions. The reason for this is that the service process works at full capacity as long as there is a sufficient number of vehicles in the system. For $t = 80$, the congestion time is almost equal to the time horizon. Even though the number of arrivals and the congestion time are the same for $t = 20, 40, 60$, the total number of vehicles in the system depends on t with the most severe conditions for $t = 20$, see Figure 6.13.

Figure 6.10: The arrival-rate intensity ρ_i during a rush hour as a function of time.

Figure 6.11: The evolution of the total number of vehicles in the system for $t = 20, 40, 60, 80$. The dashed lines represent the number of travelling vehicles.

Figure 6.12: The arrival and departure rates as functions of time for $t = 20, 40, 60, 80$ minutes.

Figure 6.13: The total number of vehicles in the system through the prediction horizon as a function of t .

6.3. Conclusions

In this chapter, we developed a transient model for large traffic networks that include traffic-light and priority intersections. Our model predicts the number of vehicles in the system for a certain time horizon. In particular, we find the distribution of the number of vehicles for each lane and each second. This information can be used to find the average number of vehicles in the system, but also, for example, to estimate the probability that a lane is empty. Compared to a traffic microsimulation, our model is quite accurate and fast, which means it can be used for model-based online control.

In the numerical results, we studied the effect of rush hours on the system. We found that after a change in the arrival rate the departure rates quickly converge to a certain value, which for undersaturated conditions leads to a steady state. We also found that for oversaturated conditions, the length of the period of the high arrival rate does not influence the time during which the system is congested.

The current model does not include the effect of blocking, which is an interesting topic for future research. However, we still can estimate the probability of blocking by the probability that a queue has more than a certain number of vehicles. Thus, our model should be especially useful if one of the objectives of the control is to reduce the probability of blocking. In this case, our model is able to provide the estimate of the probability and should give more accurate results.

Chapter 7

Conclusions

‘The important thing is not to stop questioning.’

Albert Einstein

In urban areas, growing population and increased individual mobility result in traffic congestion, which, in turn, leads to long average delays and uncertain travel times. Traffic jams usually form at bottlenecks such as intersections. Proper traffic-light control at the intersections provide a way to decrease the delays and improve the traffic flow. In this thesis, we modelled the effect of traffic-light settings on the behaviour of the system using queueing theory. In [Part I](#), we introduce the problem and review the existing literature on (queueing) traffic models. In [Part II](#), we analyse a certain class of queueing systems and propose a new way of computing the expectation and distribution of the queue length in these systems. In [Part III](#), we develop accurate queueing models for urban traffic networks and provide insights in the behaviour of the traffic systems. The developed models can be analysed using the methods of [Part II](#). Below, we give an overview of our results and suggest possible directions for further research.

In [Chapter 2](#), we consider a class of rational [probability-generating functions \(pgfs\)](#). In this class, the numerator of the pgf is a linear combination of functions with unknown coefficients, where the functions form a special geometric sequence. For such a pgf, we find the unknowns and the mean of the corresponding random variable in terms of contour integrals without any implicitly-defined variables as in the [root-finding](#) and [matrix-analytic](#) approaches. We apply this method to the [bulk-service queue](#) and several extensions of the [fixed-cycle traffic-light](#) model. To obtain our contour integral solution, we represent the pgf in a product form and the mean as an [additive](#) function of the roots of the characteristic equation. The latter result allows us to find the mean using one contour integral.

When using the classical root-finding approach for the systems considered in

Chapter 2, one obtains a linear system of equations on the unknowns. This system depends on the functions in the numerator, the zeroes of the denominator and a normalisation constant. In Chapter 3, we study this system of equations as a function of the zeroes for general functions in the numerator. We give an explicit matrix representation of the pgf, where the entries of the matrix are symmetric functions of the roots. This representation allows one to use the roots even if they are close to each other without encountering the corresponding numerical sensitivity problems. Moreover, it is possible to find the pgf using contour integrals instead of computing the roots, which can further improve the accuracy. This approach can be applied to the models developed in Part III. We also prove that the pgf has the product form as in Chapter 2 and the corresponding random variable has a non-trivial additive mean if and only if the functions in the numerator of the pgf form (up to a linear transformation) a geometric sequence. Under a certain condition, the mean is independent of the roots.

One of the directions for further research is to generalise our results to other queueing systems that involve roots of a certain equation but do not have a rational pgf studied in Chapters 2 and 3. For example, this may happen when a system is analysed using Laplace-Stieltjes transforms, see, e.g., Cohen (1969), or matrices, see, e.g., Chaudhry et al. (2012) and Perel and Yechiali (2008). Since the numeration of the roots is usually arbitrary, the dependency of the unknowns on these roots should be symmetric. For this reason, we believe that the ideas of Chapters 2 and 3 can be used in the analysis of such systems, and, therefore, it should be possible to determine the unknowns in terms of contour integrals. Another research direction would be to describe all symmetric functions that can be represented as contour integrals. Finally, our closed-form integral results might be used to obtain structural results for the considered queueing systems.

In Part III we generalise the fixed-cycle traffic-light model considered in Part II for traffic networks. In Chapters 4 and 5, we develop an analytical steady-state model for any tree-like network of traffic-light intersections. We consider two types of the traffic control: fixed and semi-actuated. For each lane, our model predicts the average queue length and the average delay. In our model, we take the randomness of the driver behaviour and the arrival process into account, which makes it as accurate as a traffic microsimulation. Furthermore, our model is fast and, in case of small traffic networks, can be used for optimisation of the traffic-light settings. Using our model, we outline the differences between fixed and semi-actuated controls and study the relation between average delays and green waves on an arterial road.

The detailed description of the arrival process to a lane in Chapters 4 and 5 leads to low computation speed for large networks. For this reason, we consider a simple approximation of the arrival processes in Chapter 6. There, we develop a transient model for large traffic networks that consist of traffic-light and priority intersections. The model of Chapter 6 predicts the number of vehicles in the system for a certain time horizon. In particular, we find the distribution of the number of vehicles for each lane and each second. This information can be used

to find the average number of the vehicles in the system, but also, for example, to estimate the probability that a lane is empty or blocked. Compared to a traffic microsimulation, our model is quite accurate. Moreover, it provides rich information about queue-length distributions and their evolution while a microsimulation gives only one realisation for each run. Using our transient model, it is possible to analyse the behaviour of the system under rapidly changing conditions such as during rush hours. As an example, we considered a part of the traffic network of Enschede, a city in the Netherlands.

One of the possible applications of our models is to study the effect of uncertainty in the drivers' behaviour on the delays. In this way, one can compare traditional traffic and future situation with autonomous vehicles, the behaviour of which is more deterministic. Another interesting topic for future research would be to incorporate the effect of blocking in our steady-state and transient models. One way to do so is to consider queues with finite capacity. However, it may be more important to look at the [horizontal](#) queues since due to [shock](#) waves the capacity of the lane may vary with time. An even bigger challenge would be to include the impact of navigational systems, the use of which makes the drivers avoid the most congested areas. This means that the arrival rate at a queue is not independent of the queue length or the expected lost time. For routing, the variance of the travel time for different routes also plays an important role. For this reason, it is interesting to find the distribution of the delay instead of the queue length. Another research direction would be to integrate detector information in our transient model. For example, we can update the queue-length distribution when detectors report an empty queue, or an arrival. In such a model, one needs to incorporate the probability of detector failure in order to construct a robust system. In this way, it would be possible to build an online model-based traffic control, which predicts the behaviour of the system for several minutes ahead and chooses the best traffic-light settings that, for example, reduce delays and blocking probabilities.

Glossary

This glossary specifies the terms defined in this thesis and some terms commonly used in queueing theory, traffic theory, optimisation theory and algebra.

(k, m) -transformation of a function $f(z)$ is a function of variables $z_1, \dots, z_m$ defined as

$$F_k^m = F_k^m(z_1, \dots, z_m) = \sum_{i=k}^{\infty} \alpha_i \zeta_{i-k}(z_1, \dots, z_m),$$

where α_i are the coefficients in the Taylor expansion of the function $f(z)$ at 0, i.e., $f(z) = \sum_{i=0}^{\infty} \alpha_i z^i$, and ζ_{i-k} are the complete homogeneous symmetric polynomials. pp. 69, 86

acceleration delay is the delay of a vehicle experienced after leaving the lane and caused by the slow speed of this or the leading vehicle. pp. 19, 110, 139, 157, 166, 170

additive function is a function of several variables such that its value at a point is equal to a sum of the values of a one-variable function for each variable, see Definition 3.3. pp. 66, 72, 74, 81, 181

additive-mean property is the property that $X'(1)$, where $X(z)$ is a pgf of type (3.1), is an additive function of the zeroes of the denominator in (3.1). pp. 74

alternant matrix is a matrix that consists of entries $f_j(z_k)$, see Definition 3.4. pp. 66, 68, 72, 78, 85, 89

auxiliary queue is an always-empty queue that represents an intermediate destination point. pp. 161, 168, 170, 173

background state is the second coordinate in the two-dimensional Markov chain of $M/G/1$ -type queue; usually it has a finite state space. pp. 36, 37

bandwidth is the (time) width of a band in the time-space diagram that goes through the green times of the intersections along an arterial road according to the free-flow speed, see [Gartner and Stamatiadis \(2002\)](#). pp. 133, 143

bulk-service queue is a discrete-time queueing model, where the customers are served in batches of the fixed size, see [Subsection 2.2.1](#). pp. 10, 14, 25, 33, 46, 58, 63, 76, 77, 181

capacity of a lane is the average number of vehicles that can depart during one cycle in over-saturated conditions. pp. 4, 53, 98, 106, 113, 118, 125, 169

chromosome is the representation of a solution in a genetic algorithm. pp. 141

common ratio is the number r for the geometric sequence $a, ar, ar^2, \dots$ pp. 63

complete homogeneous symmetric polynomials are symmetric polynomials ζ_m of variables $z_1, \dots, z_n$ defined as

$$\zeta_m = \zeta_m(z_1, \dots, z_n) = \sum_{1 \leq i_1 \leq \dots \leq i_m \leq n} z_{i_1} \cdots z_{i_m},$$

for $m = 1, 2, \dots$, with $\zeta_0 = 1$, and $\zeta_m = 0$ if $m < 0$. pp. 68

congestion time is the time before the system returns to the steady state after a temporary increase in the arrival rates, see [Subsection 6.2.3](#). pp. 176

crossover operation is an operation in genetic algorithms that produces new solutions using a pair of ‘parent’ solutions. pp. 141

curse of dimensionality refers to several phenomena in analysis of multi-dimensional systems; in particular, it refers to the exponential number of states in discrete multi-dimensional systems. pp. 20, 163

cycle is a period of time that is equal to one green and one red time. pp. 4, 17, 42, 51, 96, 136, 152, 167, 175

delayed arrival is an arrival of a vehicle that joined the queue at the upstream lane. pp. 108, 152

delayed departure is the departure of a vehicle that joined the queue. pp. 97, 101, 103, 108, 116, 136, 149, 157, 164, 168

demand is the average number of arrivals per certain time period, e.g., per cycle. pp. 4

effective green time is an idealised time period during which the vehicles are assumed to depart with constant inter-departure times. pp. 8, 17, 42

effective red time is the rest of the cycle time after the effective green time. pp. 8, 18, 42

elementary symmetric polynomials are symmetric polynomials σ_m of variables $z_1, \dots, z_n$ defined as

$$\sigma_m = \sigma_m(z_1, \dots, z_n) = \sum_{1 \leq i_1 < \dots < i_m \leq n} z_{i_1} \cdots z_{i_m},$$

for $m = 1, \dots, n$, with $\sigma_0 = 1$, and $\sigma_m = 0$ if either $m > n$, or $m < 0$. pp. 28, 67

empty state is a state in MAP that has arrival rate 0. pp. 151

end state is a state in MAP that represents a gap between arrivals from different upstream lanes and has arrival rate 0. pp. 110, 151

exhaustive service is the service type for which a queue is served until it empties. pp. 12

external arrivals are free-flow arrivals from outside the system. pp. 103

external lane is an entrance lane from outside the system. pp. 103, 108, 142, 149, 165

external queue is a queue that has arrivals from outside the system; it corresponds to an external lane. pp. 163, 165, 174

factorisation property is the property that the pgf of type (3.1) can be represented as a product where each of the terms depends on not more than one zero of the denominator in (3.1), see (3.19). pp. 74

FCTL model see fixed-cycle traffic-light model. pp. 8, 14, 25, 32, 41, 63, 93, 95, 97

fixed control is a type of traffic control under which each lane has fixed lengths of green and red time independently of the actual traffic. pp. 6, 19, 93, 102, 113, 121, 134, 182, 205

fixed-cycle traffic-light model (FCTL) is a discrete-time queueing model that describes the evolution of the queue length at a lane operated under fixed control. pp. 8, 181, 205

free arrival is an arrival of a vehicle that proceeded without stopping at the upstream lane. pp. 109, 151, 152

free departure is the departure of a vehicle that arrived during green time, found an empty queue and proceeded without stopping. pp. 97, 103, 109, 116, 136, 150, 157, 164

- free-flow arrivals** are such arrivals that the numbers of arrivals at different time intervals are independent and identically distributed. pp. 98, 103, 108
- free-flow speed** is the normal speed in a non-congested situation. pp. 8, 17, 42, 108, 119, 133, 152, 163, 165, 170, 173
- fundamental diagram** is the relation between the density and flow of vehicles. pp. 8
- gated service** is a service type for which the server serves all the customers that were present in the queue upon the arrival of the server. pp. 12
- green wave** is such an (ideal) situation that the vehicles travelling along an arterial road arrive at the intersections during green time. pp. 19, 133, 139, 145, 149
- green-end lag** is the part of the yellow time during which there are still departures; this time is included in the effective green time. pp. 8
- green-wave efficiency** is a measure of the green-wave performance, see [Section 5.1](#). pp. 20, 133, 135
- horizontal-queue model** is a traffic model that focuses on the spatial distribution of the vehicles at the lane, as opposed to the vertical-queue models, see [Motie and Savla \(2015\)](#). pp. 8, 19, 176, 183
- internal arrivals** are arrivals from an upstream intersection. pp. 103
- internal lane** is a lane between two intersections in the system. pp. 103, 108, 126, 151
- internal queue** is a queue that has arrivals from other queues but not from outside the system; it corresponds to an internal lane. pp. 163, 165
- Kronecker delta** is the function of two variables that is equal to 1 if the variables coincide and to 0 otherwise. pp. 79
- lane** is a part of the road that is intended for one line of traffic. pp. 4, 41, 93
- lane-changing queue** is a virtual queue that consists of the vehicles that want to change to a particular lane. pp. 162, 168, 173
- level state** is the first coordinate in the two-dimensional Markov chain of $M/G/1$ -type queue. pp. 36
- load** is a measure of system utilisation. In particular,

- *load of a lane* is a ratio of the average number of arrivals per cycle and the capacity of the lane under fixed control;
- *load of the intersection/system for a given fixed control* is the maximum load at the lanes;
- *load of the intersection/system for a given cycle length* is equal to the maximum value ρ such that multiplying the arrival rates to the network by at least $1/\rho$ renders an unstable system, which means that no possible traffic-light settings (with the given cycle length) exist to accommodate all arriving vehicles.

pp. 4, 38, 50, 55, 113, 139, 144, 174

low-priority queue is a queue that corresponds to a lane with lower priority at a priority intersection. pp. 162, 168, 170

macroscopic traffic model is a traffic model which describes the density, speed and flow of the vehicles. pp. 7

MAP see Markovian arrival process. pp. 20, 93, 103, 108, 136, 149, 160, 163

Markovian arrival process is an arrival process that is generated by an underlying Markov chain. pp. 20

matrix-analytic approach is a numerical approach for $M/G/1$ - and $G/M/1$ -type queueing systems that involves finding the roots of a characteristic equation, see Subsection 2.2.4. pp. 15, 25, 36, 38, 58, 181

MAXBAND is a deterministic algorithm for facilitating green waves, see Gartner and Stamatidis (2002). pp. 139, 143

mesoscopic traffic model is a traffic model that provides more details than macroscopic models but less than microscopic models; it usually describes the traffic behaviour in terms of distributions. pp. 7

microscopic traffic model is a traffic model which describes the movements of individual vehicles. pp. 6

minimal polynomial of $z_1, \dots, z_n$ is the polynomial of the smallest power that has $z_1, \dots, z_n$ as zeroes, i.e., $\prod_{k=1}^n (z - z_k)$. pp. 26, 71

mutation operation is an operation in genetic algorithms that alters a solution in order to increase the diversity of the population. pp. 141

non-dominated solution is a solution of multi-objective optimisation such that any other solution is worse in at least one of the objectives. pp. 140

offset is the time between the beginning of a cycle at one intersection and the beginning of a cycle at the other intersection. pp. 13, 94, 111, 120, 128, 134, 140, 160

overflow queue is the queue at the end of service and before the arrivals for the bulk-service queue or at the end of green time for the FCTL queue. pp. 9, 26, 43, 53

over-saturated conditions refer to a situation when the number of arrivals exceeds the capacity of the system, i.e., when load is more than 1. pp. 118, 159

Pareto front is the set of all non-dominated solutions in a multi-objective optimisation. pp. 140

Pareto-optimal solution is a non-dominated solution. pp. 141

pgf see probability-generating function. pp. 10, 25, 31, 40, 58, 63, 72, 81, 98, 136, 181

phase is a combination of traffic lights that have green time simultaneously. pp. 11, 51, 95, 120, 140, 160

phase schedule is the sequence of phases (and their intended lengths) in a cycle. pp. 95, 139, 148

platoon is a group of vehicles that departed from the same upstream lane during the same green period. pp. 7, 13, 20, 42, 103, 114, 145, 149, 165

polling system is a system, where a number of queues are served by a single server according to a certain service policy, see [Takagi \(2000\)](#). pp. 12, 45

power-sum symmetric polynomials are symmetric polynomials η_m of variables $z_1, \dots, z_n$ defined as

$$\eta_m = \eta_m(z_1, \dots, z_n) = \sum_{l=1}^n z_l^m,$$

for $m \geq 0$, with $\eta_m = 0$ if $m < 0$. pp. 28, 70

priority intersection is an intersection where vehicles at one road have priority over the other vehicles. pp. 11, 20, 160, 170, 182

probability-generating function is a function that is equal to $\mathbb{E}(z^X)$ for some non-negative discrete random variable X . pp. 10, 181

root-finding approach is a numerical approach for certain queueing systems, that involves finding the roots of a characteristic equation, see [Subsection 2.2.3](#). pp. 15, 25, 35, 38, 58, 63, 71, 101, 181

- roundabout** is a circular intersection, which is used as an alternative for uncontrolled intersections or traffic-light intersections when two or more roads (with similar traffic densities) intersect, see [Pochowski et al. \(2016\)](#). pp. 12
- SCPG** is SUMO cycle program generator, see [Subsection 5.2.3.1](#). pp. 142
- semi-actuated control** is a type of traffic control under which lengths of the green and red times of a lane depend on the actual traffic at some of the lanes. pp. 11, 19, 93, 113, 121, 182, 205
- shock wave** is an abrupt change in the density and speed of vehicles which propagates through the traffic. pp. 7, 14, 19, 183
- skew-symmetric function** is a function of several variables the value of which changes the sign for any transposition of the variables, see [Definition 3.2](#). pp. 65, 73
- split moment** is an auxiliary variable that determines the allocation of the green time, see [Subsection 5.2.2](#). pp. 141
- start-up lag** is the time in the beginning of the green time which is lost due to acceleration of the vehicles; it is equal to the time between the beginnings of real and of effective green time. pp. 8, 17
- SUMO** (Simulation of Urban MObility) is an open-source traffic microsimulation suite, see [Lopez et al. \(2018\)](#). pp. 7, 94, 102, 113, 118, 139, 142, 144, 175
- symmetric function** is a function of several variables the value of which does not change for any permutation of the variables, see [Definition 3.1](#). pp. 65, 68, 72, 81, 182
- time-headway** is the distance in time between two consecutive vehicles. pp. 7, 10, 160
- time-limited service** is a service type for which the server visits a queue for a given time period; under exhaustive time-limited service, the server leaves the queue before the end of the time period if the queue empties. pp. 13
- total delay of a vehicle** is the actual travel time minus the travel time under the free-flow speed. pp. 9, 110, 139
- traffic-light intersection** is an intersection where the flow of vehicles is controlled by traffic lights. pp. 4, 12, 19, 42, 49, 93, 134, 139, 160, 165, 182, 205
- traffic-light queue** is a queue that corresponds to a lane at a traffic-light intersection. pp. 65, 160, 161, 167, 168

transposition is a permutation that interchanges two elements. pp. 65

tree-like network is a network without cyclic paths, see [Figure 5.1](#). pp. 134, 182, 205

uncontrolled intersection is an intersection for which there are no priority signs and the vehicles yield to the vehicles on their right (or on their left for left-hand traffic as in the United Kingdom or Japan). pp. 12

unprotected turn is a turning movement such that the turning vehicles need to yield to the conflicting traffic that has green time at the same time. pp. 170

Vandermonde determinant is the determinant of the Vandermonde matrix. pp. 68, 69, 74, 78, 86

Vandermonde matrix is an alternant matrix for which $f_j(z) = z^{j-1}$. pp. 66, 78

vertical-queue model is a traffic model that disregards the spatial distribution of the vehicles at the lane, see [Motie and Savla \(2015\)](#). pp. 8, 14, 19, 176

waiting time is the delay due to waiting in the queue. pp. 13, 19, 33, 98, 110, 115, 139

Wronskian of the functions $f_1(z), \dots, f_n(z)$ is the determinant of the matrix

$$\begin{pmatrix} f_1(z) & \dots & f_n(z) \\ \vdots & \ddots & \vdots \\ f_1^{(n-1)}(z) & \dots & f_n^{(n-1)}(z) \end{pmatrix}.$$

pp. 67, 70, 84

Bibliography

- Abhishek, Stochastic models for unsignalized road traffic intersections. Ph.D. thesis, 2019.
- Adan, I. J. B. F., van Leeuwaarden, J. S. H., Winands, E. M. M., On the application of Rouché's theorem in queueing theory. *Operations Research Letters*, 34 (3), 355–360, 2006.
- Akçelik, R., Time-dependent expressions for delay, stop rate and queue length at traffic signals. *Australian Road Research Board*, 1980.
- Akçelik, R., Roupail, N. M., Overflow queues and delays with random and platooned arrivals at signalized intersections. *Journal of Advanced Transportation*, 28 (3), 227–251, 1994.
- Al Hanbali, A., Busy period analysis of the level dependent PH/PH/1/K queue. *Queueing Systems*, 67 (3), 221–249, 2011.
- Al Hanbali, A., de Haan, R., Boucherie, R. J., van Ommeren, J. C. W., Time-limited polling systems with batch arrivals and phase-type service times. *Annals of Operations Research*, 198 (1), 57–82, 2012.
- Baer, N., Boucherie, R. J., van Ommeren, J. C. W., Threshold queueing to describe the fundamental diagram of uninterrupted traffic. *Transportation Science*, 53 (2), 585–596, 2019.
- Bailey, N. T. J., On queueing processes with bulk service. *Journal of the Royal Statistical Society: Series B (Methodological)*, 16 (1), 80–87, 1954.
- Beckmann, M., McGuire, C. B., Winsten, C. B., Studies in the economics of transportation. *Yale University Press*, 1956.
- Bekker, B. M., Ivanov, O. A., Merkurjev, A. S., An algebraic identity and the Jacobi–Trudi formula. *Vestnik St. Petersburg University: Mathematics*, 49 (1), 1–4, 2016.

- Ben-Elia, E., Ettema, D., Rewarding rush-hour avoidance: A study of commuters' travel behavior. *Transportation Research Part A: Policy and Practice*, 45 (7), 567–582, 2011.
- Berends-Ballast, M., et al., Transport and mobility 2016, 2016. <https://www.cbs.nl/en-gb/publication/2016/25/transport-and-mobility-2016> (accessed on 3 June 2019).
- Bôcher, M., The theory of linear dependence. *Annals of Mathematics*, 2 (1/4), 81–96, 1900.
- Boffey, D., Luxembourg to become first country to make all public transport free. *The Guardian*, 2018. <https://www.theguardian.com/world/2018/dec/05/luxembourg-to-become-first-country-to-make-all-public-transport-free> (accessed on 10 December 2018).
- Boon, M. A. A., Polling models: from theory to traffic intersections. Ph.D. thesis, 2011.
- Boon, M. A. A., Adan, I. J. B. F., Winands, E. M. M., Down, D. G., Delays at signalized intersections with exhaustive traffic control. *Probability in the Engineering and Informational Sciences*, 26 (3), 337–373, 2012.
- Boon, M. A. A., Janssen, A. J. E. M., van Leeuwen, J. S. H., Timmerman, R. W., Pollaczek contour integrals for the fixed-cycle traffic-light queue. *Queueing Systems*, 91 (1-2), 89–111, 2019.
- Boon, M. A. A., van Leeuwen, J. S. H., Networks of fixed-cycle intersections. *Transportation Research Part B: Methodological*, 117, 254–271, 2018.
- Boxma, O. J., Groenendijk, W. P., Pseudo-conservation laws in cyclic-service systems. *Journal of Applied Probability*, 24 (4), 949–964, 1987.
- Brackstone, M., McDonald, M., Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2 (4), 181–196, 1999.
- Braess, D., Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12 (1), 258–268, 1968.
- Branston, D., van Zuylen, H., The estimation of saturation flow, effective green time and passenger car equivalents at traffic signals by multiple linear regression. *Transportation Research*, 12 (1), 47–53, 1978.
- Bruneel, H., Wittevrongel, S., Analysis of a discrete-time single-server queue with an occasional extra server. *Performance Evaluation*, 116, 119–142, 2017.
- Calvert, S. C., Taale, H., Snelder, M., Hoogendoorn, S. P., Probability in traffic: a challenge for modelling. In: *4th International Symposium on Dynamic Traffic Assignment*, 2012.

- Ceylan, H., Developing combined genetic algorithm—hill-climbing optimization method for area traffic control. *Journal of Transportation Engineering*, 132 (8), 663–671, 2006.
- Chaudhry, M. L., Kim, N. K., A complete and simple solution for a discrete-time multi-server queue with bulk arrivals and deterministic service times. *Operations Research Letters*, 31 (2), 101–107, 2003.
- Chaudhry, M. L., Samanta, S. K., Pacheco, A., Analytically explicit results for the GI/C-MSP/1/ ∞ queueing system using roots. *Probability in the Engineering and Informational Sciences*, 26 (2), 221–244, 2012.
- Chaudhry, M. L., Singh, G., Gupta, U. C., A simple and complete computational analysis of MAP/R/1 queue using roots. *Methodology and Computing in Applied Probability*, 15 (3), 563–582, 2013.
- Cohen, J. W., The single server queue. North-Holland Amsterdam, 1969.
- Daganzo, C. F., The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28 (4), 269–287, 1994.
- Darroch, J. N., On the traffic-light queue. *The Annals of Mathematical Statistics*, 35 (1), 380–388, 1964.
- Darroch, J. N., Newell, G. F., Morris, R. W. J., Queues for a vehicle-actuated traffic light. *Operations Research*, 12 (6), 882–895, 1964.
- Di Gangi, M., Cantarella, G. E., Di Pace, R., Memoli, S., Network traffic control based on a mesoscopic dynamic flow model. *Transportation Research Part C: Emerging Technologies*, 66, 3–26, 2016.
- Fellendorf, M., Vortisch, P., Microscopic traffic flow simulator VISSIM. In: *Fundamentals of traffic simulation*. Vol. 145 of International Series in Operations Research & Management Science. pp. 63–93, 2010.
- Frigui, I., Alfa, A., Analysis of a time-limited polling system. *Computer communications*, 21 (6), 558–571, 1998.
- Fu, L., Hellenga, B., Delay variability at signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 1710 (1), 215–221, 2000.
- García-Nieto, J., Olivera, A. C., Alba, E., Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 17 (6), 823–839, 2013.

- Gärling, T., Schuitema, G., Travel demand management targeting reduced private car use: effectiveness, public acceptability and political feasibility. *Journal of Social Issues*, 63 (1), 139–153, 3 2007.
- Gartner, N.H., Stamatiadis, C., Arterial-based control of traffic flow in urban grid networks. *Mathematical and Computer Modelling*, 35 (5-6), 657–671, 2002.
- Gawron, C., Simulation based traffic assignment: computing user equilibria in large street networks. Ph.D. thesis, 1999.
- Gazis, D.C., The origins of traffic theory. *Operations Research*, 50 (1), 69–77, 2002.
- Goodwin, P.B., Empirical evidence on induced traffic. *Transportation*, 23 (1), 35–54, 1996.
- Greenshields, B.D., Bibbins, J.R., Channing, W.S., Miller, H.H., A study of traffic capacity. In: *Highway Research Board Proceedings*. Vol. 14. pp. 448–477, 1935.
- Grether, D., Neumann, A., Nagel, K., Simulation of urban traffic control: a queue model approach. *Procedia Computer Science*, 10, 808–814, 2012.
- Gunning, R.C., Rossi, H., Analytic functions of several complex variables. Vol. 368. American Mathematical Society, 2009.
- He, Q., Fundamentals of matrix-analytic methods. Vol. 365. Springer, 2014.
- Heidemann, D., Wegmann, H., Queueing at unsignalized intersections. *Transportation research part B: Methodological*, 31 (3), 239–263, 1997.
- Helbing, D., Gas-kinetic derivation of Navier-Stokes-like traffic equations. *Physical Review E*, 53 (3), 253–282, 1996.
- Helbing, D., Modeling multi-lane traffic flow with queuing effects. *Physica A: Statistical Mechanics and its Applications*, 242 (1-2), 175–194, 1997.
- Hoogendoorn, S.P., Bovy, P.H.L., Modeling multiple user-class traffic. *Transportation Research Record: Journal of the Transportation Research Board*, 1644 (1), 57–69, 1998.
- Hoogendoorn, S.P., Bovy, P.H.L., State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215 (4), 283–303, 2001.
- Hoogendoorn, S.P., Knoop, V.L., van Zuylen, H.J., Model-based stochastic control of traffic networks. In: *The Third International Symposium on Transportation Network Reliability*, 2007.

- Janssen, A. J. E. M., van Leeuwen, J. S. H., Analytic computation schemes for the discrete-time bulk service queue. *Queueing Systems*, 50 (2-3), 141–163, 2005.
- Janssen, A. J. E. M., van Leeuwen, J. S. H., Back to the roots of the M/D/s queue and the works of Erlang, Crommelin and Pollaczek. *Statistica Neerlandica*, 62 (3), 299–313, 2008.
- Javid, R. J., Nejat, A., Hayhoe, K., Quantifying the environmental impacts of increasing high occupancy vehicle lanes in the United States. *Transportation Research Part D: Transport and Environment*, 56, 155–174, 2017.
- Kesting, A., Treiber, M., Helbing, D., General lane-changing model MOBIL for car-following models. *Transportation Research Record: Journal of the Transportation Research Board*, 1999 (1), 86–94, 2007.
- Kim, N. K., Chaudhry, M. L., Equivalences of batch-service queues and multi-server queues and their complete simple solutions in terms of roots. *Stochastic Analysis and Applications*, 24 (4), 753–766, 2006.
- Kingman, J. F. C., The single server queue in heavy traffic. *Mathematical Proceedings of the Cambridge Philosophical Society*, 57 (4), 902–904, 1961.
- Kingman, J. F. C., The first Erlang century—and the next. *Queueing Systems*, 63, 3–12, 2009.
- Knoflach, H., A new way to organize parking: the key to a successful sustainable transport system for the future. *Environment and Urbanization*, 18 (2), 387–400, 2006.
- Knoop, V., van Zuylen, H., Hoogendoorn, S., The influence of spillback modelling when assessing consequences of blockings in a road network. *European Journal of Transport and Infrastructure Research*, 8 (4), 287–300, 2008.
- Knopp, K., Theory of Functions Parts I and II, Two Volumes Bound as One. Dover Publications, 1996.
- Koonce, P., Rodegerdts, L., Traffic signal timing manual. Tech. rep., United States. Federal Highway Administration, 2008.
- Krajzewicz, D., Brockfeld, E., Mikat, J., Ringel, J., Rössel, C., Tuchscheerer, W., Wagner, P., Wösl, R., Simulation of modern traffic lights control systems using the open source traffic simulation SUMO. In: *Proceedings of the 3rd Industrial Simulation Conference 2005*. pp. 299–302, 2005.
- Krauß, S., Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. Ph.D. thesis, 1998.

- Kutadinata, R., Moase, W., Manzie, C., Zhang, L., Garoni, T., Enhancing the performance of existing urban traffic light control through extremum-seeking. *Transportation Research Part C: Emerging Technologies*, 62, 1–20, 2016.
- Lämmer, S., Helbing, D., Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008 (04), P04019, 2008.
- Laña, I., Del Ser, J., Vélez, M., Vlahogianni, E. I., Road traffic forecasting: recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10 (2), 93–109, 2018.
- Langer, G., ABC News poll: traffic in the United States. *ABC News*, 2005.
- Lee, J., Phase force-off modes in coordinated-actuated signal operations. *Procedia Computer Science*, 83, 600–606, 2016.
- Lehoczky, J. P., Traffic intersection control and zero-switch queues under conditions of Markov chain dependence input. *Journal of Applied Probability*, 9 (2), 382–395, 1972.
- Liang, Z., Chen, H., Flötteröd, Y., Sohr, A., Bei, X., Bottazzi, M., Trumpold, J., Real-time microscopic traffic simulation and optimization at intersections with video traffic detection. *International Journal of Electrical and Electronic Engineering & Telecommunications*, 7 (1), 30–37, 2018.
- Lighthill, M. J., Whitham, G. B., On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 229 (1178), 317–345, 1955.
- Little, J. D. C., A proof for the queuing formula: $L = \lambda W$. *Operations Research*, 9 (3), 383–387, 1961.
- Litvak, N. V., Fedotkin, M. A., An adaptive control for conflicting flows: a quantitative and numerical study of its probabilistic model. *Automation and Remote Control*, 61 (6), 952–960, 2000.
- Livny, M., Melamed, B., Tsiolis, A. K., The impact of autocorrelation on queuing systems. *Management Science*, 39 (3), 322–339, 1993.
- Lo, H. K., A novel traffic signal control formulation. *Transportation Research Part A: Policy and Practice*, 33 (6), 433–448, 1999.
- Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., Microscopic traffic simulation using SUMO. In: *21st International Conference on Intelligent Transportation Systems*. pp. 2575–2582, 2018.

- Lu, J., Osorio, C., A probabilistic traffic-theoretic network loading model suitable for large-scale network analysis. *Transportation Science*, 52 (6), 1509–1530, 2018.
- Lu, T., Yang, Z., Ma, D., Jin, S., Bi-level programming model for dynamic reversible lane assignment. *IEEE Access*, 6, 71592–71601, 2018.
- Macedo, F., Milheiro-Oliveira, P., Pacheco, A., Simões, M. L., Application of a particular class of Markov chains in the assessment of semi-actuated signalized intersections. In: *International Conference on Analytical and Stochastic Modeling Techniques and Applications*. pp. 138–151, 2017.
- Maerivoet, S., De Moor, B., Cellular automata models of road traffic. *Physics Reports*, 419 (1), 1–64, 2005.
- Mahnke, R., Kühne, R., Probabilistic description of traffic breakdown. In: *Traffic and Granular Flow'05*. pp. 527–536, 2007.
- Marinică, N., Boel, R., Platoon based model for urban traffic control. In: *American Control Conference*. pp. 6563–6568, 2012.
- McNeil, D. R., A solution to the fixed-cycle traffic light problem for compound Poisson arrivals. *Journal of Applied Probability*, 5 (3), 624–635, 1968.
- McShane, C., The origins and globalization of traffic control signals. *Journal of Urban History*, 25 (3), 379–404, 1999.
- Mead, D. G., Newton's identities. *The American Mathematical Monthly*, 99 (8), 749–751, 1992.
- Miller, A. J., Settings for fixed-cycle traffic signals. *Journal of the Operational Research Society*, 14, 373–386, 1963.
- Motie, M., Savla, K., On dynamical analysis of a horizontal traffic queue. In: *54th Annual Conference on Decision and Control*. pp. 2181–2186, 2015.
- Mueller, E. A., Aspects of the history of traffic signals. *IEEE Transactions on Vehicular Technology*, 19 (1), 6–17, 1970.
- Nagel, K., Particle hopping models and traffic flow theory. *Physical Review E*, 53 (5), 4655–4672, 1996.
- Neuts, M. F., Stationary waiting-time distributions in the GI/PH/1 queue. *Journal of Applied Probability*, 18 (4), 901–912, 1981.
- Neuts, M. F., Matrix-analytic methods in queuing theory. *European Journal of Operational Research*, 15 (1), 2–12, 1984.

- Newell, G. F., Queues for a fixed-cycle traffic light. *The Annals of Mathematical Statistics*, 31 (3), 589–597, 1960.
- Newell, G. F., Approximation methods for queues with application to the fixed-cycle traffic light. *SIAM Review*, 7 (2), 223–240, 1965.
- Newell, G. F., Theory of highway traffic signals. Tech. rep., 1989.
- Newell, G. F., A simplified theory of kinematic waves in highway traffic, part I: general theory. *Transportation Research Part B: Methodological*, 27 (4), 281–287, 1993a.
- Newell, G. F., A simplified theory of kinematic waves in highway traffic, part II: queueing at freeway bottlenecks. *Transportation Research Part B: Methodological*, 27 (4), 289–303, 1993b.
- Nourinejad, M., Roorda, M. J., Impact of hourly parking pricing on travel demand. *Transportation Research Part A: Policy and Practice*, 98, 28–45, 2017.
- Oblakova, A., Al Hanbali, A., Boucherie, R. J., Ommeren, J. C. W., Zijm, W. H. M., Exact expected delay and distribution for the fixed-cycle traffic-light model and similar systems in explicit form. *Memorandum Faculty of Mathematical Sciences University of Twente*, 2056, 2016.
- Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Comparing semi-actuated and fixed control for a tandem of two intersections. *Memorandum Faculty of Mathematical Sciences University of Twente*, 2061, 2017.
- Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Green wave analysis in a tandem of traffic-light intersections. *Memorandum Faculty of Mathematical Sciences University of Twente*, 2062, 2017.
- Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., An exact root-free method for the expected queue length for a class of discrete-time queueing systems. *Queueing Systems*, 92 (3–4), 257–292, 2019.
- Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Roots, symmetry and contour integrals in queueing systems. *Memorandum Faculty of Mathematical Sciences University of Twente*, 2067, 2019.
- Oblakova, A., Al Hanbali, A., Boucherie, R. J., van Ommeren, J. C. W., Zijm, W. H. M., Stochastic approximation model for urban traffic networks, 2019 (in preparation).
- Olszewski, P. S., Modeling probability distribution of delay at signalized intersections. *Journal of Advanced Transportation*, 28 (3), 253–274, 1994.

- Osorio, C., Chong, L., A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transportation Science*, 49 (3), 623–636, 2015.
- Osorio, C., Wang, C., On the analytical approximation of joint aggregate queue-length distributions for traffic networks: a stationary finite capacity Markovian network approach. *Transportation Research Part B: Methodological*, 95, 305–339, 2017.
- Osorio, C., Yamani, J., Analytical and scalable analysis of transient tandem Markovian finite capacity queueing networks. *Transportation Science*, 51 (3), 823–840, 2017.
- Pacheco, A., Simões, M. L., Milheiro-Oliveira, P., Queues with server vacations as a model for pretimed signalized urban traffic. *Transportation Science*, 51 (3), 841–851, 2017.
- Paveri-Fontana, S. L., On Boltzmann-like treatments for traffic flow: a critical review of the basic model and an alternative proposal for dilute traffic analysis. *Transportation Research*, 9 (4), 225–235, 1975.
- Payne, H. J., Models of freeway traffic and control. *Mathematical Model of Public System*, 1, 51–61, 1971.
- Perel, E., Yechiali, U., Queues where customers of one queue act as servers of the other queue. *Queueing Systems*, 60 (3-4), 271–288, 2008.
- Pipes, L. A., An operational analysis of traffic dynamics. *Journal of Applied Physics*, 24 (3), 274–281, 1953.
- Pochowski, A., Paul, A., Rodegerdts, L. A., Roundabout Practices. Transportation Research Board, 2016.
- Pollaczek, F., Théorie analytique des problèmes stochastiques relatifs à un groupe de lignes téléphoniques avec dispositif d'attente. Gauthier-Villars, 1961.
- Polson, N. G., Sokolov, V. O., Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79, 1–17, 2017.
- Pradhan, S., Gupta, U. C., Samanta, S. K., Queue-length distribution of a batch service queue with random capacity and batch size dependent service: $M/G_r^Y/1$. *OPSEARCH*, 53 (2), 329–343, 2016.
- Prigogine, I., Herman, R., Kinetic theory of vehicular traffic, 1971.
- Quinn, D. J., A review of queue management strategies. *Traffic Engineering and Control*, 33 (11), 600–605, 1992.

- Rao, A. M., Madhu, E., Gupta, K., Impact of odd even scheme on transportation systems in Delhi. *Transportation in Developing Economies*, 3 (1), 2017.
- Reed, T., Kidd, J., INRIX Global Traffic Scorecard. *INRIX Research*, 2019.
- Reuschel, A., Fahrzeugbewegungen in der kolonne. *Osterreichisches Ingenieur Archiv*, 4, 193–215, 1950.
- Richards, P. I., Shock waves on the highway. *Operations Research*, 4 (1), 42–51, 1956.
- Riska, A., Smirni, E., Exact aggregate solutions for M/G/1-type Markov processes. *SIGMETRICS Performance Evaluation Review*, 30 (1), 86–96, 2002.
- Schrank, D., Eisele, B., Lomax, T., TTI's 2012 urban mobility report. *Texas A&M Transportation Institute. The Texas A&M University System*, 2012.
- Servi, L. D., D/G/1 queues with vacations. *Operations Research*, 34 (4), 619–629, 1986.
- Shinar, D., Compton, R., Aggressive driving: an observational study of driver, vehicle, and situational variables. *Accident Analysis & Prevention*, 36 (3), 429–437, 2004.
- Son, Y. T., Cassidy, M. J., Modanat, S. M., Evaluating steady-state assumption for highway queueing system. *Journal of Transportation Engineering*, 121 (2), 182–190, 1995.
- Stoffers, M., Cycling as heritage: representing the history of cycling in the Netherlands. *The Journal of Transport History*, 33 (1), 92–114, 2012.
- Storm, J., Bhulai, S., Kager, W., Mandjes, M., Analysis of roundabouts. Results presented at the conference Mathematics Applied in Transport and Traffic Systems, Delft, 2018. https://dirkab7tlqy5f1.cloudfront.net/CiTG/Over%20faculiteit/Afdelingen/Transport%20%26%20Planning/Conferences/Matts/JaapStorm_Pres_Rndbt_MATTS2018_Delft.pdf (accessed on 2 February 2019).
- Takagi, H., Analysis and application of polling models. In: *Performance Evaluation: Origins and Directions*. pp. 423–442, 2000.
- Tarko, A. P., Random queues in signalized road networks. *Transportation Science*, 34 (4), 415–425, 2000.
- Tarnoff, P. J., Parsonson, P. S., Selecting traffic signal control at individual intersections. *NCHRP Report* (233), 1981.

- Tian, N., Li, J., Zhang, Z.G., Matrix analytic method and working vacation queues—a survey. *International Journal of Information and Management Sciences*, 20 (4), 603–633, 2009.
- Toledo, T., Choudhury, C.F., Ben-Akiva, M.E., Lane-changing model with explicit target lane choice. *Transportation Research Record: Journal of the Transportation Research Board*, 1934 (1), 157–165, 2005.
- Tomescu, O., López, V.R.T., Nemțanu, F.C., Bățroș, I., Traffic signal green-wave control strategy based on drivers' behaviors. *Analele Universității Eftimie Murgu Reșița. Fascicula de Inginerie*, 18 (2), 146–156, 2011.
- Transportation Research Board, Highway capacity manual. Washington, DC, 2000.
- van den Broek, M.S., van Leeuwen, J.S.H., Adan, I.J.B.F., Boxma, O.J., Bounds and approximations for the fixed-cycle traffic-light queue. *Transportation Science*, 40 (4), 484–496, 2006.
- van Leeuwen, J.S.H., Delay analysis for the fixed-cycle traffic-light queue. *Transportation Science*, 40 (2), 189–199, 2006.
- van Lint, J.W.C., van Hinsbergen, C.P.I.J., Short-term traffic and travel time prediction models. *Artificial Intelligence Applications to Critical Transportation Issues*, 22 (1), 22–41, 2012.
- van Ommeren, J.C.W., The discrete-time single-server queue. *Queueing systems*, 8 (1), 279–294, 1991.
- van Wageningen-Kessels, F., van Lint, H., Vuik, K., Hoogendoorn, S., Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics*, 4 (4), 445–473, 2015.
- van Zuylen, H.J., Viti, F., Probabilistic model for queues, delays and waiting time at controlled intersections. In: *Proceedings of the 86th Annual Meeting of the Transportation Research Board*, 2007.
- Vinberg, E.B., A course in algebra. No. 56. American Mathematical Society, 2003.
- Viti, F., Li, M., van Zuylen, H.J., A probabilistic model for actuated traffic signals. In: *the 11th World Conference on Transport Research*, 2007.
- Viti, F., Tampere, C.M.J., Frederix, R., Traffic performance of short-distanced traffic lights with probabilistic spillback. In: *Proceedings of the 88th Annual Meeting of the Transportation Research Board*. pp. 11–15, 2009.
- Wagenbuur, M., Why are we waiting? Protest for longer green times, 2014. <https://bicycledutch.wordpress.com/2014/10/16/why-are-we-waiting-protest-for-longer-green-times/> (accessed on 27 February 2019).

- Wang, Y., van Schuppen, J. H., Vrancken, J., Prediction of traffic flow at the boundary of a motorway network. *IEEE Transactions on Intelligent Transportation Systems*, 15 (1), 214–227, 2014.
- Webster, F. V., Traffic signal settings. Tech. rep., 1958.
- Weiss, G. H., Maradudin, A. A., Some problems in traffic delay. *Operations Research*, 10 (1), 74–104, 1962.
- Wilkinson, J. H., The perfidious polynomial. *Studies in Numerical Analysis*, 24, 1–28, 1984.
- Wu, X., Liu, H. X., A shockwave profile model for traffic flow on congested urban arterials. *Transportation Research Part B: Methodological*, 45 (10), 1768–1786, 2011.
- Yannis, G., Golias, J., Antoniou, C., Effects of urban delivery restrictions on traffic movements. *Transportation Planning and Technology*, 29 (4), 295–311, 2006.
- Zhang, G., Wang, Y., Wei, H., Chen, Y., Examining headway distribution models with urban freeway loop event data. *Transportation Research Record: Journal of the Transportation Research Board*, 1999 (1), 141–149, 2007.
- Zheng, F., van Zuylen, H., The development and calibration of a model for urban travel time distributions. *Journal of Intelligent Transportation Systems*, 18 (1), 81–94, 2014.
- Zijm, W. H. M., Adan, I. J. B. F., Buitenhok, R., van Houtum, G. J., Capacity analysis of an automated kit transportation system. *Annals of Operations Research*, 93 (1-4), 423–446, 2000.

Summary

With a growing population, traffic congestion puts a strain on individual travellers, economy and the environment and forms a major problem for cities all over the world. To decrease congestion, it is crucial to utilise the existing infrastructure more efficiently, in particular, by choosing proper traffic-light control settings. This requires a thorough analysis of possible control policies and their influence on the system behaviour. In this thesis, we approach this problem using queueing theory. We present discrete-time models for urban traffic networks and theoretical results concerning the computation of the queue length. The thesis consists of three parts.

In **Part I**, we give the motivation and the background of our research. In **Chapter 1**, we review (queueing) traffic models and common techniques to obtain the queue-length distributions. We show that certain elements of traffic behaviour are often overlooked in analytic traffic models and that classical queueing techniques for traffic models require some implicitly-defined variables. **Chapter 1** concludes with a summary of the contributions of this thesis.

Part II contains our theoretical results on obtaining the steady-state distribution of the queue length for a certain type of queueing systems. In **Chapter 2**, we consider a classical **fixed-cycle traffic-light** model, which is a basic queueing model for an isolated **traffic-light intersection**. We derive the distribution of the queue length for this and other models in a new way, which does not include implicitly-defined variables. In **Chapter 3**, we generalise the results of **Chapter 2**. In particular, we consider a wider class of queueing systems, which includes also models described in **Part III**. For this class, we propose another way to find the queue-length distribution and give a necessary and sufficient condition for application of the method described in **Chapter 2**.

In **Part III**, we develop queueing models for urban traffic networks. We begin in **Chapter 4** by considering two traffic-light intersections in tandem. Taking into account acceleration of the vehicles and the correlation between arrivals, we develop a realistic model. Moreover, we analyse and compare two types of control: **fixed** and **semi-actuated** control. In **Chapter 5**, we generalise the model of **Chapter 4** for the case of any **tree-like network**, i.e., an acyclic network. Moreover, we propose a measure of green-wave efficiency. We observe that a high green-wave

efficiency does not necessarily mean short delays. The comparison with deterministic approaches of facilitating green waves shows the importance of considering the stochastic nature of the traffic. The model developed in [Chapters 4 and 5](#) is computationally demanding for large networks. For this reason, in [Chapter 6](#), we construct an approximation model for large traffic networks. In this chapter, we do not limit ourself to only traffic-light intersections but also analyse priority intersections. In this way, we are able to consider realistic networks. In our numerical results, we study the behaviour of a traffic network during a rush hour using as an example a part of the city of Enschede in the Netherlands.

In [Chapter 7](#), we conclude the thesis and suggest possible directions for further research.

Samenvatting

Een groeiende bevolking met een navenant groter wagenpark in combinatie met een grotere mobiliteit veroorzaken grote verkeersproblemen, zoals files in en rondom steden over de hele wereld, die verstrekkende gevolgen hebben voor zowel individuele reizigers, de economie en het milieu. Om minder last te hebben van verkeersopstoppen, is het belangrijk om de bestaande infrastructuur efficiënter te gebruiken. Een van de manieren is het goed regelen van verkeerslichten. Dit vereist een grondige analyse van mogelijke verkeerslichtregelingen en hun invloed op het systeem. In dit proefschrift benaderen we dit probleem met behulp van de wachtrijtheorie. We ontwikkelen discrete-tijdmodellen voor stedelijke verkeersnetwerken en vinden theoretische resultaten voor de berekening van de wachtrijlengte. Het proefschrift bestaat uit drie delen.

In **Deel I** bespreken we de motivatie en achtergrond van ons onderzoek. In **Hoofdstuk 1** geven we een overzicht van verkeersmodellen en veel gebruikte technieken in de wachtrijtheorie om de wachtrijlengte te bepalen. We laten zien dat bepaalde aspecten van het verkeersgedrag vaak over het hoofd worden gezien in analytische verkeersmodellen en dat klassieke wachtrijtechnieken vaak gebruik maken van een aantal impliciet gedefinieerde variabelen. Het hoofdstuk wordt afgesloten met een kort overzicht van de resultaten uit dit proefschrift.

Deel II bevat onze theoretische resultaten voor de stationaire verdeling van de wachtrijlengte voor een bepaald type wachtrijsystemen. In **Hoofdstuk 2** wordt het klassieke model voor een verkeerslicht met een vaste cyclustijd onderzocht. Dit klassieke verkeerslichtmodel is een basismodel voor een enkel kruispunt. We berekenen de verdeling van de wachtrijlengte zowel voor dit model als voor andere modellen op een nieuwe manier die geen impliciet gedefinieerde variabelen nodig heeft. In **Hoofdstuk 3** generaliseren we de resultaten van **Hoofdstuk 2** waarbij we een bredere klasse van wachtrijsystemen beschouwen, die ook de modellen uit **Deel III** omvat. Voor deze klasse vinden we een alternatieve manier om de verdeling van de wachtrijlengte te berekenen. Bovendien, geven we een noodzakelijke en voldoende voorwaarde voor het kunnen toepassen van de methode beschreven in **Hoofdstuk 2**.

In **Deel III** ontwikkelen we wachtrijmodellen voor stedelijke verkeersnetwer-

ken. In **Hoofdstuk 4** bestuderen we een netwerk van twee kruispunten. We houden rekening met zowel de correlatie tussen aankomsten als de versnelling van de voertuigen, wat ons model realistisch maakt. Bovendien analyseren en vergelijken we twee typen verkeerslichtregelingen: vaste en verkeersafhankelijke regelingen. In **Hoofdstuk 5** breiden we het model uit **Hoofdstuk 4** uit voor netwerken zonder cykels. Bovendien definiëren we een maat voor de groene-golfefficiëntie van dit netwerk. We laten zien dat een hoge groene golfefficiëntie niet noodzakelijk tot korte vertragingen leidt. Het vergelijken met deterministische methoden om een groene golf te krijgen, toont aan hoe belangrijk het is om het stochastische karakter van het verkeer te beschouwen. Het model uit **Hoofdstukken 4 en 5** is rekenkundig gezien veeleisend voor grote netwerken. Daarom ontwikkelen we in **Hoofdstuk 6** een benaderingsmodel voor grote verkeersnetwerken. In dit hoofdstuk, beschouwen we kruispunten met verkeerslichten en voorrangskruispunten. Op deze manier, kunnen we realistische netwerken analyseren. Voor onze numerieke resultaten nemen we als voorbeeld een deel van de stad Enschede in Nederland. Met behulp van ons model bestuderen we het gedrag van dit verkeersnetwerk tijdens het spitsuur.

Hoofdstuk 7 geeft een overzicht van het proefschrift en mogelijke richtingen voor verder onderzoek.

About the author

Anna Oblakova was born in Moscow, Russia, on the 21st of February 1992. In 2009, she obtained her high school diploma and began her studies at the Faculty of Mechanics and Mathematics at Moscow State University. Starting from 2011, she specialised in general topology and geometry and focused on isometric embeddings of metric spaces. She obtained her specialist degree (with honours) in 2014.

In 2015, Anna moved to the Netherlands and became a Ph.D. candidate within Dynafloat project at Stochastic Operations Research group of the University of Twente. During this project, she focused on applications of queueing theory to the field of urban traffic. Her Ph.D. research culminates with this dissertation.