TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Performance analysis at the crossroad of queueing theory and road traffic

Citation for published version (APA):
Timmerman, R. W. (2022). *Performance analysis at the crossroad of queueing theory and road traffic*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Eindhoven University of Technology.

Document status and date:
Published: 28/01/2022

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 17. Nov. 2023

Performance analysis at the crossroad of queueing theory and road traffic

Rik Timmerman

The work in this thesis has been sponsored by The Netherlands Organization for Scientific Research (NWO) under grant number 438-13-206, ARS T&TT, and De Verkeersonderneming.

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, Ghent University, Maastricht University, Tilburg University, University of Twente, VU Amsterdam, Wageningen University and Research, KU Leuven, Universiteit Hasselt.

© 2022 by Rik Timmerman

Performance analysis at the crossroad of queueing theory and road traffic by Rik Timmerman.

A catalogue record is available from the Eindhoven University of Technology Library

ISBN 978-90-386-5422-5

Cover design by Rik Timmerman

Printed by Gildeprint

Performance analysis at the crossroad of queueing theory and road traffic

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op vrijdag 28 januari 2022 om 16:00 uur

door

Rik Wesley Timmerman

geboren te Drunen

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof. dr. J.J. Lukkien
1e promotor:	prof. dr. J.S.H. van Leeuwaarden
2e promotor:	prof. dr. ir. I.J.B.F. Adan
co-promotor:	dr. ir. M.A.A. Boon
leden:	prof. dr. ir. O.J. Boxma
	dr. ir. N.P. Dellaert
	dr. V.L. Knoop (Technische Universiteit Delft)
	prof. dr. ir. J. Walraevens (Universiteit Gent)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Acknowledgments

A period of four years has come to an end. It is my great pleasure to thank all of you who have (in)directly contributed to the writing of this thesis in this period.

First and foremost I am indebted to the my supervisors. Marko, as a daily supervisor, we have had a lot of contact and shared a lot of experiences. You were always there for me when I needed you, no matter the topic or cause of the need. You are truly a good person and I have learned a lot from your way of looking at things. You are also a very nice person with a lot of humor (which is much appreciated). Johan, you have been a source of inspiration for me during the past years. Your enthusiasm and perfectionism in writing are greatly appreciated. Ivo, we may not have talked often, but if we did it was always a pleasure for me. You were always willing to free up some time for me in your agenda, you were always very cooperative, and you came up with nice views on all kinds of matters. Onno, although you are not an official supervisor looking at the promotional committee, I have very much appreciated all of your supervision. Your thoroughness, humor, and willingness to help others on all kinds of matters (being research, education, or other matters) must be a blessing for anyone that works with you. For me, it has been a blessing for sure and all your help will not be forgotten.

I would like to thank Joris Walraevens, Victor Knoop, and Nico Dellaert for being part of the promotion committee. Thank you for reading my thesis and for giving critical comments which have improved the thesis in various ways.

My PhD position has been part of the Dynafloat grant. I would like to thank Richard Boucherie, Jan-Kees van Ommeren, Sindo Núñez Queija, Rob van der Mei, Wim van Nifterick (from ARS T&TT), and Gerard Eijkelenboom (from De Verkeersonderneming) for being a critical audience at our joint meetings. Further, I would like to thank Anna and Sara for sharing many nice moments and for giving me a warm welcome into the Dynafloat group.

I would like to thank Guido Janssen for joint work on the complex contour integral for the FCTL queue and the heavy-traffic scaling of the FCTL queue

leading to Chapters 2 and 3 in this thesis. I would also like to thank Bo Klaasse for doing an excellent internship on the “goede dagen” assignment of De Verkeersonderneming and for co-writing a paper on it, which has lead to Chapter 8.

The stochastic operations research group is a very vibrant research group. Thank you all for useful discussions. In particular, I would like to thank Stella Kapodistria for several nice conversations. I am also grateful to Chantal, Ellen, and Patty. Without you, the group would really look very different. Chantal, thank you for our many nice conversations and your view on all types of matters. Further, I feel very lucky to have been part of a research group with many PhD-students and it was always nice to talk to all of you. A special thanks goes to Ellen: I have enjoyed our early conversations over a cup of tea very much. Hopefully we can extend this tradition in some way!

A special thanks goes to my officemates, Angelos, Mark, Mayank, and Viktoria. We have been together at the office for about 3 years and I have thoroughly enjoyed every minute of it. Angelos, I am glad to have been your coffee buddy (even though I do not drink coffee) during your morning breakfast. Mayank, you have always brought a very positive atmosphere to the office, thank you for that. Viktoria, you were always willing to help your officemates with all sorts of things and very often you brought some nice new perspectives. Mark, without elaborating further, thank you for being a good friend. I hope that we continue seeing each other! I would also like to thank Alberto, Chenyan, Martijn, Peter, and Wessel for several nice conversations after the “old gang” left.

Maybe I would not have started this PhD position without you Yvonne. Together we took the decision to do our bachelor final project with Marko and Johan. As you can see, our final project has in some way been the foundation for this thesis. Thank you for all the nice moments that we have shared during and after our studies. I am very happy that we are still in touch and play boardgames now and then with Eugène. Let’s try to keep this up!

Without the support of my family, I would probably not have been able to write this thesis. Opa De Wit, dankjewel voor al je interesse in mijn onderzoek. Hoewel Oma De Wit, en Opa en Oma Timmerman er niet bij zijn, weet ik zeker dat ze trots zouden zijn. Verder, Pap, Mam en Fleur, dankjulliewel voor het continu aanhoren van alle werkgerelateerde zaken die ik jullie heb verteld de afgelopen jaren. Zonder jullie steun had me dit allemaal veel zwaarder gevallen.

Rik Timmerman
Vlijmen, December 2021

Contents

Acknowledgments **v**

1 Introduction **1**
1.1 Motivation 1
1.2 Signalized intersection modeling 6
1.3 Queueing models for road traffic 17
1.4 Main contributions and outline of this thesis 29

2 Pollaczek contour integrals for the Fixed-Cycle Traffic-Light queue **33**
2.1 Introduction 33
2.2 Standard solution for the FCTL queue 34
2.3 Main results 35
2.4 Algorithmic methods 42
2.5 Proof of the Pollaczek contour-integral representation 46
2.6 Conclusion 55

Appendices **57**
2.A Root-finding algorithm 57
2.B Poisson case 59

3 Optimal capacity allocation for heavy-traffic Fixed-Cycle Traffic-Light queues and intersections **61**
3.1 Introduction 61
3.2 FCTL queue in heavy traffic 63

3.3 Capacity allocation problems 68
3.4 Numerical examples of capacity allocation 74
3.5 Proof of heavy-traffic theorem using the transform method 78
3.6 Conclusion 88

Appendices **89**
3.A Remaining proofs 89

4 Heavy-traffic scaling of vehicle-actuated traffic lights **97**
4.1 Introduction 97
4.2 Model description 99
4.3 Theoretical background 100
4.4 Simulation results 103
4.5 Conclusion 113

5 Fixed-Cycle Traffic-Light queue with multiple lanes and blockages **115**
5.1 Introduction 115
5.2 Detailed model description 119
5.3 PGFs and performance measures for the bFCTL queue 124
5.4 Examples 134
5.5 Conclusion 146

Appendices **149**
5.A Stability condition for the bFCTL queue 149

6 Approximation scheme for multidimensional queueing models **153**
6.1 Introduction 153
6.2 Approximation scheme 162
6.3 k-limited polling models 170
6.4 A two-class queue with alternating service discipline 182
6.5 Traffic lights with double-lane access control 187
6.6 Conclusion 192

Appendices **195**
6.A PGFs for k-limited polling models 195
6.B PGFs for traffic lights with double-lane access control 197

7 Platoon forming algorithms for intelligent street intersections 201
7.1 Introduction 201
7.2 Model formulation 204
7.3 Platoon forming algorithms 206
7.4 Speed profile algorithms 211
7.5 Performance analysis 226
7.6 Comparison traditional traffic light and PFAs 234
7.7 Conclusion 237

8 Detection of high traffic flow in uncongested traffic states 239
8.1 Introduction 239
8.2 Description of the location and the data 241
8.3 The main algorithm 244
8.4 Key insights and validation 254
8.5 Conclusion 263

9 Conclusions and future work 265
9.1 Summary of contributions in this thesis 265
9.2 Suggestions for extensions and future research 268

Bibliography 271

List of publications 293

Summary 295

About the author 299

Chapter 1

Introduction

1.1 Motivation

Everyone who uses means of transportation like cars and bikes experiences congestion. Congestion is a phenomenon that occurs when the interaction between vehicles on the road causes the vehicles to slow down, which typically is the case when there are (relatively) many vehicles on the road. Congestion has a negative impact on many levels: it increases travel time, costs money, increases pollution, etc. The KiM, the Dutch Institute for Transport Policy Analysis, has estimated the amount of direct costs associated to congestion on the Dutch highway system for 2018 to be a devastating 3.3 billion euros [196]. In addition, indirect costs add up to a number close to 1 billion euros [196]. As these numbers are only related to congestion at highways, we have a very conservative estimation of the total costs caused by congestion in the Netherlands because quite some secondary and urban roads are also congested. These numbers alone should be sufficient to convince anyone to take a look at measures to reduce congestion.

In fact, much research is aimed at reducing congestion. There are numerous countermeasures against congestion which range from obvious ones, like reducing the amount of traveling in a network, to very advanced ones, e.g. network-wide optimization of route choices. Despite ongoing research and taken countermeasures, congestion remains an enormous societal problem. One of the main reasons of the continuing problems is the discrepancy between capacity and demand on the road-traffic network. Congestion emerges (more easily)

when demand exceeds the capacity of a certain part of the network and two, a priori easy, solutions become clear: increasing the capacity or decreasing the demand. The former is often very costly and consumes much space as, generally, additional roads need to be built, which is sometimes met with (fierce) resistance. Therefore, this way of tackling congestion does not seem to be sustainable. Reducing demand is the other obvious option and this has a positive effect when achieved. An illustrative example is the absence of traffic jams in the lockdowns during the Covid-19 pandemic. However, a structural reduction of travel demand, e.g. during peak hours, seems difficult to achieve. Therefore, research on other countermeasures is conducted to mitigate congestion as much as possible. This mainly focuses on more efficient use of the existing roads.

A potential and notorious source of congestion is the traffic-light controlled intersection. Intersections are an inevitable part of road networks, although they decrease the capacity of the connecting roads: indeed, only vehicles from *some* of the connecting roads are able to cross the intersection simultaneously. Traffic lights are often installed to make sure that traffic from various directions can drive across the intersection safely and that traffic is organized smoothly. Besides this, traffic lights at intersections should be designed in such a way that the negative effects of intersections are mitigated as much as possible. Traffic-light settings play a key role in this respect as they govern when each vehicle is allowed to cross the intersection, which has a direct influence on the capacity of and delay experienced at the intersection. Good traffic-light settings are thus of substantial societal value and lead to less congestion by ensuring an efficient use of the available roads.

Nowadays, traffic lights are not always functioning optimally with respect to keeping congestion or, similarly, delay experienced by vehicles to a minimum. As an example: how often do we get frustrated by a traffic light that turns red just at the moment that we want to cross an intersection? Early research on traffic-light settings goes back to at least 1940, which shows that frustration caused by traffic lights, unfortunately, has a long tradition.

The current state-of-the-art in the Netherlands is the so-called intelligent traffic-light installation (in Dutch: intelligente verkeersregelinstallatie). This is usually a type of traffic light that adapts its green and red pattern to the presence of vehicles on each road by obtaining a current view of the state of the network. This allows for an efficient green and red time allocation [62]. On top of that, the intelligent traffic light might inform approaching vehicles regarding the green and red pattern for the coming period. This enables vehicles to anticipate on e.g. changes in the traffic light, which makes it possible for the vehicle to approach the intersection in a smart way. An example would be to

reduce speed early and accelerate in such a way that vehicles start crossing the intersection at the start of the green period *and* cross the intersection at a high/maximum speed. This decreases the time spent on the intersection by an individual vehicle, which leads to a more efficient use of the intersection. These intelligent traffic lights have the potential to reduce congestion significantly, as is shown by small-case real-life examples. An example can be found in Helmond where a 20% reduction in waiting time was obtained [66]. This reduction partly follows from a better coordination between the vehicles and the traffic light, but also from vehicles approaching the intersection in a different way leading to a better utilization of the intersection.

Despite many research attempts and field-case studies over the last decades, the search for *optimal* traffic-light settings remains problematic, even for intelligent traffic lights. Mathematically, traffic-light controlled systems are very hard to analyze rigorously. Therefore, finding optimal settings for a big intersection, and even more so for a network of intersections, is a hard and challenging task. Complicating factors include, but are not limited to, randomness and complicated traffic interactions; we next discuss those two factors.

Randomness in road traffic is omnipresent, e.g. at the level of an individual driver in the form of driver behavior; at the level of the number of vehicles driving towards an intersection during a given period of time; and at the level of congestion emergence. Other sources of randomness include the weather and the occurrence of traffic accidents. Stochasticity (severely) complicates the rigorous treatment of traffic models and limits our ability to give meaningful advice to traffic engineers. Despite complicating the models, randomness cannot be neglected and has to be taken into account as it significantly impacts the performance of any road-traffic system. A canonical way to address complex systems involving randomness and delays is the use of queueing theory.

Queueing theory is a research area that studies and develops mathematical models for general queueing phenomena with a broad range of applications. It makes sense that queueing-theoretic methods are a useful tool to analyze delays in road traffic, due to their ability to explain the *dynamics* leading to congestion. Therefore, the performance analysis of dynamic queueing models is an important step towards the optimization of traffic management policies.

The second complicating factor is the mere fact that several roads meet at an intersection as can be seen in Figure 1.1. Moreover, there might be a varying number of traffic streams from a single direction, potentially heading in different directions; there might be green lights for several (conflicting or not) groups of vehicle streams; there might be cyclists present (on separate lanes or not); pedestrians might cross the intersection (on a pedestrian crossing with a

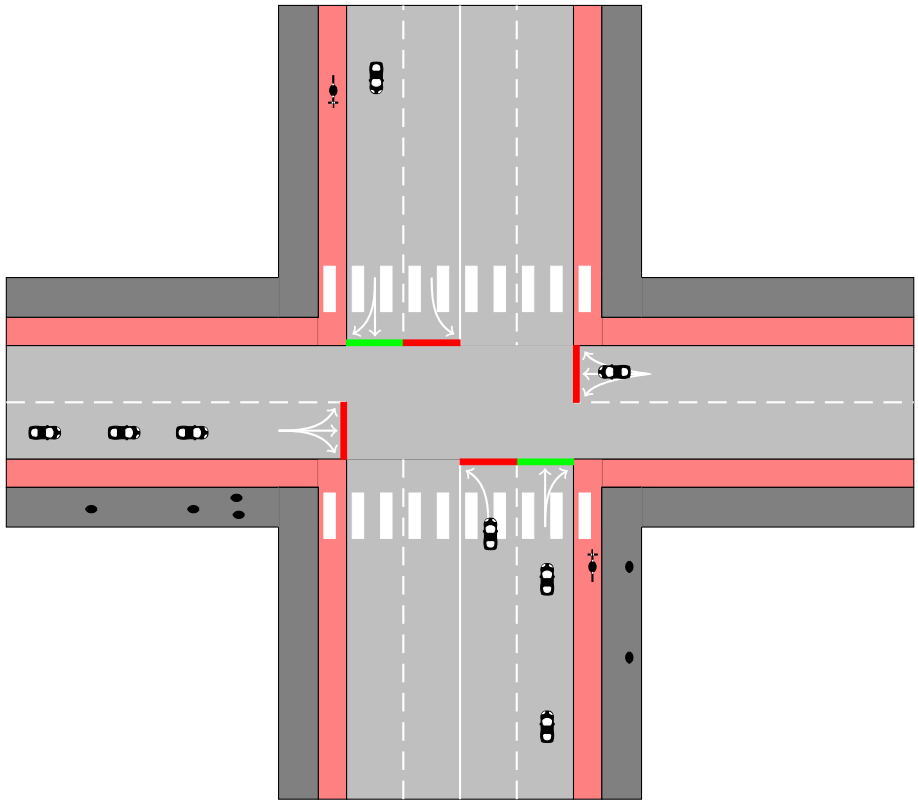


Figure 1.1: A graphical representation of a general intersection with vehicles, cyclists, and pedestrians. The intersection has six streams of cars, which are all governed by traffic lights. Moreover, there are some pedestrian crossings and also cyclists claim their share of the intersection's capacity.

separate traffic light or not); and many more complications might exist.

Typically, models involving randomness are well-understood as long as we study a single queue (e.g. corresponding to a single road or a single stream of vehicles). For intersections, however, we typically need to study higher-dimensional queueing models to capture queueing phenomena accurately as can be observed from Figure 1.1. The caveat is that such higher-dimensional stochastic models are much harder to analyze than single-dimensional models.

The study of higher-dimensional stochastic models has led to a very rich area of research, yet exact results are often hard to obtain and are typically, if existing at all, complicated in nature. However, it is sometimes possible to find good approximations or guidelines that lead to good, or even optimal, settings.

Another important factor to take into account when developing traffic-light control mechanisms is that infrastructural investments in road-traffic networks are long-term investments, typically ranging from 10 to 50 years. Therefore, special attention should be given towards the rise of (semi-)autonomous vehicles, as such vehicles are expected to occupy the road in the near future. In fact, at some places autonomous vehicles are already on the road, e.g. in Phoenix, Arizona [119]. Such vehicles are much easier to guide than vehicles driven by humans, be it only that humans might behave unexpectedly. Because of this, we have to develop new control algorithms for autonomous vehicles. Possibilities for new algorithms include, but are not limited to, different types of traffic lights, a better spread of vehicles over a network of roads, and speed-advisory or even speed-control algorithms for autonomous vehicles. We should thus think about such strategies and potentially needed road-side equipment *now* in order to prevent costly investments later on due to missing infrastructure.

Summarizing, there is an obvious need for smart(er) traffic-light control strategies. This might seem easy at first glance, but this is a very complex and challenging task. A reason why it is complicated to find good strategies, is that congestion modeling of traffic lights, which naturally leads to queueing models, is hard. This relates to the fact that we need to take random effects in road traffic into account and to the need to consider all roads that are connected to the intersection, which can be complicated as can be observed from Figure 1.1. Moreover, due to the fact that investments in road traffic are often long-term investments, we should also think about the long-term future when investing. Therefore, in this thesis, we focus on several queueing models for traffic lights and extend and deepen the knowledge on traffic-light modeling both in present-day and in futuristic settings. We do so in various ways such as (i) by developing new methodologies to address queueing models, (ii) by extending the general applicability of traffic-light models, and (iii) by obtaining (close-to) optimal traffic-light settings.

The remainder of this introductory chapter is organized as follows: we start with an overview of the literature on the modeling of signalized intersections in Section 1.2 and we continue in Section 1.3 with an exposition of some (mathematical) techniques and (queueing) models for road traffic that will be used frequently in this thesis. We close this chapter in Section 1.4 with a sketch of the main contributions and an outline of the remainder of this thesis.

1.2 Signalized intersection modeling

As demonstrated in the previous section, there is a need for good and new strategies to control intersections with traffic lights. The focus in this section is on the modeling of traffic-light controlled intersections and to this end, we give a partial overview of the literature on traffic-light models with a special emphasis on queueing models for traffic lights. We briefly discuss the most important directions of research below, before giving a more in-depth study of each research direction in Subsections 1.2.1 up to 1.2.4.

A common first step is to significantly simplify an intersection like the one in Figure 1.1. E.g., cyclists and pedestrians are often omitted in a traffic-light control study. It is often argued that they can be accounted for in the following way: either cyclists and/or pedestrians cross when a non-conflicting vehicle stream receives a green traffic light; or the cyclists and/or pedestrians impose an additional red time for all other vehicles if they receive an exclusive right of way. In this way, we are usually able to separate the analysis for vehicles and other traffic streams and significantly reduce the complexity of the model.

After such a simplification, we usually obtain a model that is amenable for some kind of queueing-theoretic performance analysis. An important and well-studied example of a tractable traffic-light model relates to a traffic light with fixed red and green times. Compared to other traffic-light strategies, traffic lights with such *fixed* settings allow for a relatively straightforward, exact analysis, as the analysis typically can be done separately for each incoming road or stream of vehicles arriving at the intersection. In Figure 1.1, this could e.g. correspond to the vehicle stream coming from the left. All that is needed for such a separation to work, is that a green traffic light for one lane, implies a red light for all conflicting flows. An example of what this means for an intersection with four streams of vehicles (and thus four traffic lights) is shown in Figure 1.2. This separation per lane leads to a separate analysis for each lane. Those models are one-dimensional queueing models which are much easier to grasp rigorously than higher-dimensional models. Another definite advantage of such a traffic-light model is that it can serve as a building block to study more complicated models because of its relative simplicity. At the same time, traffic lights with fixed settings have appealing properties in practice: in a network of intersections, the fixed red and green times can easily be used to coordinate the settings of all traffic lights in a network because the red and green times are fixed, see e.g. [40, 107, 155]. For the same reason, such traffic lights are predictable which might be beneficial if one wants to find an optimal route to go from A to B. One of the canonical traffic-light models with fixed settings is

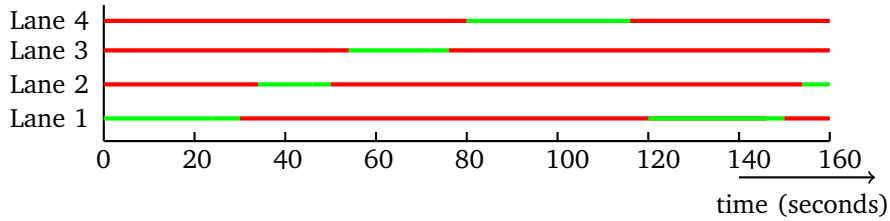


Figure 1.2: An example of a green- and red-time allocation for an intersection with four lanes and with fixed red and green times. We assume, for simplicity, that the clearance times correspond to red times for all lanes.

the Fixed-Cycle Traffic-Light (FCTL) queue. The literature on the FCTL queue and similar models is discussed in more detail in Subsection 1.2.1.

Another important set of examples of traffic-light controlled intersections are traffic lights with a vehicle-actuated strategy. In contrast with the FCTL queue, the green and red times are no longer fixed and might depend on the presence of vehicles, or the queue length, at each lane. An example of a vehicle-actuated strategy is as follows: if a queue dissolves during a green period, the remainder of the green period is skipped and the next lane receives a green light (after an appropriate clearance time). These strategies have obvious advantages above traffic lights with fixed settings. Vehicle-actuated strategies generally ensure that an empty queue does not receive a green light as long as there are vehicles waiting to cross the intersection at one of the other lanes. This could decrease the mean delay experienced by vehicles that cross the intersection. On the other hand, there are also disadvantages as it is e.g. difficult to coordinate the green and red times of traffic lights in a network since vehicle-actuated traffic lights are not as predictable as traffic lights with fixed settings. On top of that, the queueing models underneath such vehicle-actuated traffic lights are multidimensional as the start of the green period of each lane depends on the end of the green period of the previous lane. As such, vehicle-actuated traffic lights are far more difficult to grasp in a mathematically rigorous way than the queueing models underneath the FCTL queue or similar models with fixed settings. In practice, however, vehicle-actuated control of traffic lights is generally preferred above fixed control of traffic lights. We discuss part of the literature on vehicle-actuated traffic lights in Subsection 1.2.2.

The FCTL queue and many vehicle-actuated traffic-light strategies are traditionally studied for isolated intersections. However, only some traffic-light

controlled intersections can be studied in isolation in practice, because intersections are often part of an entire network. Although the focus of this thesis is not on such network models, we briefly discuss some research on networks of intersections. In the models discussed in the previous two paragraphs, arrivals of vehicles are commonly assumed to be independent. However, this is often an invalid assumption in a network setting, simply because a downstream intersection receives the output process of an upstream intersection. Such dependencies fundamentally complicate the analysis of network models. Next to this complication, the simultaneous study of intersections in a network leads to a model with a higher dimension than that of a model used to study an isolated intersection. This increases the complexity of the model even further. Models for isolated intersections can be used as building blocks for a network study and can (sometimes) be used to give a first-order approximation of the behavior of an intersection in a network setting, see e.g. [19,144]. Besides these complicating factors, the route choice of the drivers in a network becomes important and route choices might have a severe impact on the general traffic performance. An example is Braess's paradox [32], which states that adding roads to a network might lead, counterintuitively, to additional congestion. Studies on networks of intersections are obviously an important theme and some literature on traffic lights in a network of intersections is discussed in Subsection 1.2.3.

Research on signalized intersections is not limited to current-day traffic applications. There is a vast amount of research on intersection management anticipating the rise of intelligent vehicles, where the level of intelligence ranges from "able to communicate with road-side equipment" to fully autonomous vehicles. One of the bigger advantages is that intelligent vehicles are typically able to announce their arrival at the intersection before they physically arrive at the intersection. The traffic light (or more generally: an intersection controller) is then able to anticipate on the arriving vehicles and e.g. organize the vehicles into groups that cross the intersection together. This generally helps to decrease the delay of vehicles. There are several challenges however, such as how the groups of vehicles that cross the intersection together should be formed. The speed advice that is given to vehicles is also an interesting topic of study. Some literature devoted to traffic-light/intersection-access control for intelligent vehicles is discussed in Subsection 1.2.4.

1.2.1 Traffic lights with fixed settings

Models with fixed settings for traffic lights are typically relatively simple to study and understand. This traffic-light strategy has been well studied and one

of the first papers on this topic dates back to 1941 [53], which assumes constant arrival and departure times. Other notable, early papers are the papers by Wardrop [217], studying approximations for the mean delay and optimal green and red times; by Webster [218], studying empirical and simulation observations for approximations; and by Newell [141], studying approximations for a traffic light with fixed settings and binomially distributed arrivals.

A well-studied model in the realm of traffic lights with fixed signals is the FCTL queue. This is a queueing model which focuses on deriving the characteristics of the number of delayed cars in front of the traffic light. The FCTL queue will be formally introduced in Subsection 1.3.1 as it will play an important role in this thesis. Here, we give an overview of the literature on the FCTL queue.

The first complete steady-state analysis for the FCTL queue dates back to 1964 and was provided by Darroch [64]. Darroch focuses on the distribution of the so-called *overflow queue*, the queue length at the end of the green period, and computes its Probability Generating Function (PGF). From this PGF, any information about the queue-length distribution, e.g. the queue-length distribution at an arbitrary moment, can be derived. Darroch's solution requires the solution to a set of linear equations, which requires roots of a certain equation as input. This has led to the belief that the model is complicated to use. Perhaps this is the reason that many researchers have focused on obtaining approximations for various performance measures, see e.g. [148] for many such approximations and a comparison. See also [198] for bounds and approximations for e.g. the mean overflow queue.

Van Leeuwaarden [206] was the first to obtain an explicit expression for the PGF of the steady-state *delay* distribution by means of tagging vehicles and investigating the delay of such tagged vehicles.

A recent breakthrough was spurred by an observation made in a paper by Oblakova et al. [146]. The methodology developed in [146] avoids the use of roots altogether when computing the mean overflow queue. Oblakova et al. express this mean value in the form of a complex contour integral, which can be computed numerically. Another contribution in [146] is the extension of the FCTL queue to several other models, such as queueing models with slightly different queueing dynamics, random red and green times, and a model where some drivers might be distracted causing them to take a relatively long time to depart from the queue. Such generalizations make the FCTL queue more widely applicable, e.g. they allow one to model turning flows [146].

In [158], a traffic-light model with fixed settings is studied under slightly different assumptions than the ones that are made for the FCTL queue. Matrix-analytic techniques are employed to obtain several stationary performance mea-

sures such as the mean delay. The authors of [158] also perform a study in which they compare their theoretical results against a microscopic traffic simulator, which is supposed to model traffic behavior realistically, and a model from the Highway Capacity Manual. The authors in [158] argue that their model serves as a good approximation for the simulation model for small to moderate traffic loads and as a better approximation for reality for high loads than the model used in the Highway Capacity Manual.

Turning towards finding good or optimal settings, we note that an approach to obtain optimal green times for intersections with fixed settings can be found in e.g. [77]. This study extends approximation results from [198] to find integral green-time allocations which minimize the mean delay using techniques like integer programming optimization and graph theory. Other examples include the work of Van Zwieten [212], relying on linear and quadratic programming techniques; the work of Haijema [88], using theory from Markov decision processes; and many others. For a broader overview, see [76, 88, 212] and the references therein.

Besides the common steady-state type of analysis, there are only few studies on the transient behavior of traffic signals with fixed settings. This perhaps relates to the general belief that the transient analysis of queueing models is hard. However, numerical examples are relatively easy to study as long as general expressions or results are not pursued. E.g., in [216, Chapter 4], the time-dependent behavior of a traffic-light model with fixed red and green times and a possibly time-dependent arrival process is studied.

Although traffic lights with fixed settings are practically relevant in many settings, actuated traffic signals might be favorable depending on the practical situation at hand. If we e.g. allow the remainder of the green time to be skipped if a queue gets empty, we enter the realm of vehicle-actuated control strategies. Such strategies are the topic of the next subsection.

1.2.2 Vehicle-actuated traffic lights

There are various types of vehicle-actuated traffic lights. Looking at the case where one lane receives a green light, a common feature seems to be that the traffic light turns from green to red as soon as there are no longer vehicles waiting to cross the intersection on that lane. There exist generalizations of such a strategy to the case of multiple lanes receiving a green light simultaneously. Differentiation of various types of vehicle-actuated control might be done on the basis of when a traffic light switches from green to red if there are still vehicles queueing on the lane with a green light. In some cases, a change from

green to red only happens if there is *no* queue of vehicles anymore on the lane with a green light, whereas in other cases the traffic light might turn red even when the queue is non-empty. We refer to the former case as an exhaustive type of vehicle-actuated control and otherwise we refer to the strategy as a limited type of vehicle-actuated control. Another differentiation for vehicle-actuated control strategies is the following: sometimes, the actuated-control mechanism is present for all lanes and sometimes only for e.g. the main stream. We will mainly focus on the case where all lanes are governed by a vehicle-actuated mechanism. If we discuss a different type, we mention it explicitly.

One of the first studies involving vehicle-actuated traffic-light control dates back to 1940. In [81], a model is studied where there is a main road which always has a green light except if there is a vehicle on a conflicting road. In that case, a switch to the conflicting stream occurs after at most a predefined period [81], which is a simple example of a limited vehicle-actuated control strategy. Since then, a vast body of literature on vehicle-actuated traffic lights has developed and many techniques exist to find good or (in some way) optimal settings. We will not pursue a full overview and mainly focus on queueing models for vehicle-actuated traffic-light control.

In the literature on vehicle-actuated traffic-light control, it is, e.g., often assumed that there is only a *single* lane receiving a green light at each moment. Then, after a clearance time, the next lane receives a green light. In practice, this is often not the case as multiple lanes might receive a green light at the same time, which makes the studied models less relevant. Traffic-light strategies with fixed signaling are also often studied as if only a single lane receives a green light at each moment of time, but this does not lead to a discrepancy between theory and practice. This is because each lane can be studied separately under static signaling, enabling a single-dimensional study of the model as argued before, see e.g. Figure 1.2. This is not the case for vehicle-actuated traffic lights, which implies the need to study more complicated, higher-dimensional models. This might explain the difference in the amount of exact results for the underlying queueing models for vehicle-actuated and fixed settings.

A study with some exact computations that is close to a practically relevant version of vehicle-actuated traffic lights is discussed in [17]. The model at hand in [17] is a vehicle-actuated controlled traffic light where two non-conflicting roads receive a green light simultaneously. It is assumed that the traffic lights stay green until both queues are dissolved. An approximation for the mean queue length is provided which is based on exact light- and heavy-traffic limits, but even here, a fully exact analysis for the queue-length distribution is lacking.

On the other hand, there are many papers dealing with vehicle-actuated

traffic lights that do not focus on exact computations. A notable early pioneer is Newell [142, 143], who essentially uses a fluid model to study vehicle-actuated traffic lights. In [142], approximations for e.g. the waiting time for an intersection with two one-way streets and no turning flow under a vehicle-actuated control is given. It is claimed that such a strategy might reduce the waiting time with a factor 3 compared to fixed-time control. A qualitative property that is derived in [142], is that the traffic light should be switched from green to red as soon as a queue empties if one wants to minimize the mean delay. This relates to an exhaustive type of vehicle-actuated control. However, as soon as we turn to an intersection with two two-way streets, matters complicate and it is a priori not clear anymore what is optimal to do [143]. In fact, when two streams receive a green traffic light simultaneously, Newell and Osuna claim in [143] that a fixed-cycle strategy is better than a vehicle-actuated type of strategy.

Other early work at the interface of vehicle-actuated access control and queueing theory can be found in [65, 121]. The authors in [65] study an exhaustive type of vehicle-actuated control where one stream of vehicles receives a green light at each moment in time. Under several simplifying assumptions, they are able to come up with the mean waiting times using PGFs. Lehoczky [121] studies a similar model with two one-way streets, where vehicles arrive according to a general Markov chain as opposed to the Poisson arrival of vehicles considered in e.g. [142]. Again, only one lane at each moment in time is allowed to have a green light and there are switching or clearance times during which all traffic lights are red. In [121], the stability of the underlying queueing system is proven by studying the length of the green periods, which should remain finite to ensure stability. Subsequently, under appropriate assumptions, expressions for e.g. the mean delay for both lanes are derived.

In [63], simple properties of the arrival and service process are used to obtain the first and second moments of the steady-state queue lengths and waiting times by means of a recursion. The derived results can be applied to e.g. traffic systems, which include vehicle-actuated traffic lights. The corresponding type of vehicle actuation is as follows: once the traffic light turns green, each vehicle present *at that moment* is allowed to cross the intersection during the current green time. Any vehicles arriving in the queue after the traffic light turned green need to wait until the next start-of-green. The results in [63] are worth mentioning because they provide tractable and analytical results, yet the type of actuation mechanism seems less realistic for (current-day) traffic-light engineering.

A more recent study on vehicle-actuated traffic lights can be found in [137]. A steady-state analysis of an intersection with two one-way streets with an ex-

haustive type of control is presented using various methods from queueing theory. The model is solved by truncating the state space and the theoretical results are compared with simulations. In [230, Chapter 5], similar types of reasonings are used to study an intersection with four streets, each with multiple lanes and a limited actuated control.

There are many more studies on vehicle-actuated control, but there are not many that focus on an analysis of the underlying queueing model. Instead, they use e.g. reinforcement-learning techniques, model-predictive control strategies, or other optimization techniques to obtain good or optimal traffic-light settings. For some additional and recent references on vehicle-actuated control mechanisms we refer the interested reader to e.g. [76, Section 1.5.2].

To close this subsection, we refer to [145], which studies a semi-actuated controller in a network setting. The authors in [145] assume that only *some* of the lanes have a vehicle-actuated mechanism. The total cycle length is fixed and this enables the authors to provide an exact analysis for the model at hand based on techniques that are commonly used for the FCTL queue. The network setting studied in [145] is an important extension compared to the traffic-light models discussed in this and the previous subsection, as intersections tend to be part of an entire network of intersections that typically influence one another. Traffic-light control in a network setting is, therefore, the topic of the next subsection.

1.2.3 Networks of intersections

The complicated nature of the study of networks of intersections has not prevented researchers from looking into such models. We give some pointers to interesting overview papers and subsequently we focus on several works that incorporate queueing theory in the study of networks of signalized intersections.

In the literature on traditional traffic-light control for networks of intersections, a distinction is often made between fixed-time and actuated or adaptive systems. We refer the interested reader to e.g. [90, 95, 159] for historical accounts and reviews. Popular fixed-time control strategies include the MAXBAND algorithm and TRANSYT. For further information we refer to the references in [95, 159]. SCATS, SCOOT, RHODES, and UTOPIA are adaptive systems, see e.g. [90, 95, 159] and the references therein. Generally, those algorithms mainly focus on finding good settings. A disadvantage is that an analysis of e.g. the queue-length distribution is typically not feasible.

Nowadays, machine learning approaches for traffic-light control in a network have been developed as well. We refrain from giving an overview of this

rapidly evolving area of research and merely provide two references to review papers on machine learning approaches used for traffic-light control [41, 227].

Another option to find good settings for a network of traffic-light controlled intersections is to implement a self-learning policy. An example is the framework developed in [116], where an algorithm is sketched that allows for *real-time* optimization of the traffic-light policy in a network of intersections. It differs from the machine learning approaches in the previous paragraph in the sense that there is no offline learning: it uses heuristics to find close-to-optimal green times for the traffic lights in a network using only current information. By means of coordination between the various traffic lights in the network, green waves might be created automatically [116]. A case study using such a self-learning policy can be found in [23], where a comparison between a self-learning policy and a SCATS-like procedure is made for a small network of intersections in Brisbane. In [23] it is shown that the self-organizing policy has the potential to significantly reduce delays.

Only few papers try to explicitly capture the queueing dynamics at signalized intersections in a network setting. This is probably due to the complex task of taking all queues into account as well as the correlation structure for the arrivals inside the network. Moreover, spillback effects of an upstream intersection might complicate the analysis of the network, because they may result in (temporary) blockages of intersections which is difficult to capture in a queueing model. However, there are some studies that aim at capturing the queueing dynamics and we discuss them below.

Two examples can be found in [19] and in [145]. Both papers focus on the analysis of a small network of intersections, where the output process of an intersection is part of the input for the next intersection. Under additional assumptions, like independence of the arrivals between different cycles, the intersections can be studied in isolation which brings us back to studies discussed in Subsections 1.2.1 and 1.2.2. The work in [19] exclusively focuses on fixed red and green times, whereas [145] also allows for a so-called semi-actuated control. The expressions in both [19] and [145] are quite complicated due to the intrinsically complex nature of the model.

The study of a network of intersections also raises the question if one should coordinate the green times of several intersections, think e.g. of green waves. Oblakova et al. study a green wave for a series of intersections in [144], which builds on results obtained in [145]. The results in [144] indicate that a lower mean delay (compared to no green wave) might be obtained when optimal settings are found. To find the optimal settings, a genetic optimization algorithm is used. It turns out to be difficult to find such optimal settings as they for

example depend on the physical distances between intersections [144].

An interesting and recently developed line of research to study traffic lights in a network setting is the use of the so-called aggregation-disaggregation technique which has been developed for general large Markov chains, see e.g. the survey [175]. The authors in [155, 156] use this technique to formulate an approximating analysis for large-scale networks of traffic lights, respectively focusing on a stationary and a transient analysis. Both papers model each queue in front of a traffic light as a Markovian finite capacity queue (an $M/M/1/K$ queue to be precise) for which the states are divided in three aggregated sets. Subsequently, a network decomposition into subnetworks of three queues is performed, which allows the authors to circumvent the problems that come with network modeling (such as the need to study a very high-dimensional model) while still capturing most of the network structure such as spillback effects. Such decomposition methods are more generally applied, see e.g. [150, 154].

Another interesting line of research is constituted by stochastic optimization methods. The key idea is to replace a complicated objective function which is to be optimized with an approximating surrogate, a so-called metamodel, see e.g. [151]. This metamodel is in turn easier to analyze and, if sufficiently accurate as an approximation to the objective function, might serve as a way to approximate the optimal solution well. Several papers have been written on such an approach for traffic networks. In [50, 152, 153] such metamodels are created using traffic simulators and a useful approximation of the objective function seems to be obtained.

Lastly, we mention that the study in e.g. [152] indicates that the between-intersection dependence has a (potentially) significant influence on the performance measures. As such, an attempt to study a network of intersections ideally contains an accurate description of such dependencies.

Even though it is already difficult to obtain optimal traffic-light settings for current-day traffic, we also need to think about the future. What will happen when self-driving or autonomous vehicles occupy the road and will it be easier or more difficult to find good or optimal traffic-light settings? This is the topic of the next subsection.

1.2.4 The intersection of the future

It seems that self-driving vehicles have the potential to fundamentally change the way we look at transportation and that they allow for different control techniques than vehicles driven by humans. This has been recognized by many

researchers and a rich area of research has emerged. We partly discuss the literature here, mostly focusing on studies which use queueing theory.

To start with, we note that already existing models and control strategies, like the ones we discussed above, might be leveraged for the control of self-driving vehicles. However, those do not use the full potential of self-driving vehicles, because such vehicles can be controlled more easily than a human driver is able to control a regular vehicle. Exactly this ability to be controlled in a better way together with the possibility of self-driving vehicles connecting and communicating with other self-driving vehicles and road-side equipment, enables one to create different, more advanced, and more efficient control mechanisms.

A highly-cited work in this realm is a study by Tachet et al. [184]. Their aim is to organize efficient crossings for vehicles and they illustrate the benefits of their strategy in a queueing-theoretical framework. This framework is used to assess the quantitative properties of their so-called platoon-forming strategy, which relates to algorithms that form groups of vehicles, platoons, that cross the intersection together. The formation of platoons is based on arrival times of vehicles and a maximum platoon size. Tachet et al. claim that if their framework is employed, the capacity at intersections might be doubled compared to the capacity nowadays. This leads to significant reductions in the delay experienced by vehicles. This is mostly due to the platoon forming, which reduces the number of switches between different signal groups, and to shorter headways for vehicles on the same lane.

Organizing vehicles in a platoon seems a popular approach that is often used. Another example can be found in two papers by Miculescu and Karaman, see [133, 134]. They use ideas and results from a specific type of queueing models, polling models, to derive upper bounds for the mean delay performance of their algorithms. They use the so-called exhaustive service discipline meaning that cars keep joining a platoon in front of them as long as the gap between the platoon and the vehicle is not too big. Similar capacity gains can be obtained as in [184]. Besides platoon formation, they also consider the problem of how vehicles should approach the intersection. A linear programming approach is used, see [134, Section V.A.], where the vehicles are controlled in such a way that they are, under several constraints, as close to the intersection as possible. The algorithm leads to provably safe trajectories [134].

Another paper in which optimal trajectories for platoons of self-driving vehicles are studied, is [126]. Under the assumption of fixed-time signaling (during a certain prediction horizon), [126] assumes that platoons are formed and proposes a different optimal trajectory for vehicles driving towards the intersection than the one obtained in [134]. The authors in [126] take several quantities

into account in order to obtain such an optimal trajectory: the throughput of vehicles, driver comfort, the delay experienced by vehicles, and fuel consumption. They use a sequential quadratic programming technique to determine the optimal trajectory. They demonstrate that their approach can also be applied to a network of intersections. For different ways of finding good or optimal trajectories, see also the references in [126].

As can be seen, platoon forming algorithms and trajectory planning are important research directions for intersection control in futuristic settings with self-driving vehicles. We refrain from giving a full overview of this research field and instead give several pointers to papers in which relevant literature is discussed. We refer the interested reader to the survey papers [43, 108, 170] and references therein.

1.3 Queueing models for road traffic

Now that we have a high-level overview of relevant topics and models in traffic-light engineering, we turn our focus towards specific models and mathematical notions/techniques that are frequently used in the study of such models. Our aim is also to set the stage for the upcoming chapters, e.g. by focusing on a particular technique when introducing or discussing a model. We start with the FCTL queue in Subsection 1.3.1, followed by a discussion on polling models in Subsection 1.3.2. We wrap up with a look at traffic simulators, in particular SUMO, in Subsection 1.3.3.

1.3.1 The FCTL queue

We provide the classical analysis of the FCTL queue. The derivation below was first obtained by Darroch [64]. We start with a description of the FCTL queue.

The FCTL queue is a one-dimensional queueing model for a traffic light that is governed by a static signal. The FCTL queue is traditionally modeled in discrete time, which means that time is divided into slots of unit length. The green and red periods, of length g and r slots, respectively, are assumed to be fixed multiples of one slot. The total cycle thus has length $c = g + r$. Each slot corresponds to the time needed for a delayed vehicle to depart from the queue. Vehicles that arrive during a red period are delayed as are the vehicles that arrive during a green period and meet a non-empty queue. Such delayed vehicles leave the queue one-by-one during the green period. Vehicles that arrive during a green period and meet no other vehicles in the queue are treated differently

according to the following assumption, which is often referred to as the *FCTL assumption*.

Assumption 1.1 (FCTL assumption) *For those cycles in which the queue clears before the green period terminates, all vehicles that arrive during the residual green period pass through the system and experience no delay whatsoever.*

We further assume that the number of arrivals in each slot, $Y_{k,n}$, where $k = 1,\ldots,c$ denotes the slot and n the current cycle, are independent of one another.

This provides us with all details that we need to find the queue-length distribution at the end of the green period measured in number of vehicles. We also commonly refer to this queue-length distribution as the distribution of the *overflow queue*.

We follow the approach taken in [206]. As we are going to study the steady-state behavior of the FCTL queue, we need to require stability in order to ensure that the queue does not grow without bounds. When assuming that all $Y_{k,n}$ have the same distribution and if we define $Y \overset{d}{=} Y_{k,n}$, the stability condition turns out to be

$$c\mathbb{E}[Y] < g,$$

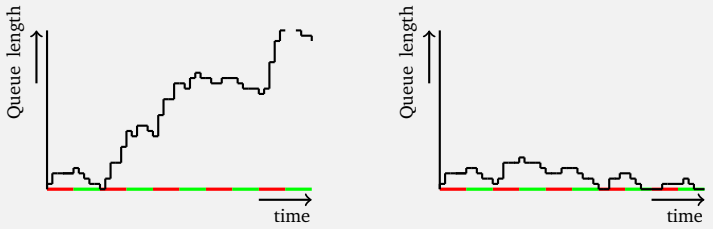
i.e. the green time g, or the maximum number of delayed vehicles that are allowed to depart during each cycle, should be bigger than the mean number of vehicles that arrive per cycle, $c\mathbb{E}[Y]$, see e.g. [206].

Stability and steady-state behavior

The concept of stability requires a further explanation. In laymen terms, it ensures that a queue does not grow beyond bounds, e.g. the number of cars in front of the traffic light remains finite. A term that is used often in the context of road-traffic research, is the vehicle-to-capacity ratio which should be less than 1. This also ensures that the number of vehicles stays finite and that the corresponding queue is stable.

We require stability, because we are interested in the steady-state behavior of the queue-length distribution. We do so to gain insight into the long-term behavior of the system. The steady-state analysis usually serves as a good approximation for reality when the circumstances do not change (too much) over a longer period of time. As we are considering the long-term development of the queue, it makes sense to impose

a condition on the system to ensure that there is, on average, enough capacity to meet the demand, which requires a stable queue or, alternatively phrased, a vehicle-to-capacity ratio which is less than 1. An illustration of a stable and unstable queue is presented in Figure 1.3.



(a) (b)

Figure 1.3: A graphical representation of the queue-length process for an unstable traffic light (a) and a stable traffic light (b). The colors on the horizontal axis indicate whether the traffic light is green or red. As can be seen, an unstable queue will, eventually, grow beyond bounds as there are on average more arrivals than departures, whereas the length of the queue remains bounded when the queue is stable.

The steady-state models that we consider are thus especially relevant when the traffic conditions remain similar during a certain period of time. For example, during a day there are several periods with similar traffic conditions, think e.g. of the morning and evening peak hours. One could thus identify periods with reasonably similar conditions throughout the day. As such, a steady-state analysis for each period sheds light onto the traffic behavior during each separate period and leads to meaningful insights.

With $X_{k,n}$ we denote the number of vehicles in the queue in slot $k = 1,\ldots,c$ during cycle n. We are especially interested in X_g, the steady-state overflow queue defined as $X_{g,n} \to X_g$ with $n \to \infty$ as essentially all performance measures can be derived from X_g (as is noted in [206]).

We obtain the following relations between the various $X_{k,n}$'s and $Y_{k,n}$'s. For

$k = g+1,\dots,c$, we have that

$$X_{k,n} = X_{k-1,n} + Y_{k,n},$$

as we only have arrivals during the red period. For $k = 1,\dots,g$, we have

$$X_{k,n} = \begin{cases} X_{k-1,n} + Y_{k,n} - 1 & \text{if } X_{k-1,n} > 0, \\ 0 & \text{if } X_{k-1,n} = 0, \end{cases}$$

where the latter case reflects the FCTL assumption and where $X_{0,n}$ is to be understood as $X_{c,n-1}$. This subsequently enables us to find the following relations. For $k = g+1,\dots,c$, we have, by conditioning on the number of vehicles in the previous slot, that

$$\mathbb{P}(X_{k,n} = i) = \sum_{l=0}^{i} \mathbb{P}(X_{k-1,n} = l)\mathbb{P}(Y_{k,n} = i - l), \tag{1.1}$$

and for $k = 1,\dots,g$, we have that

$$\mathbb{P}(X_{k,n} = i) = \begin{cases} \sum_{l=1}^{i+1} \mathbb{P}(X_{k-1,n} = l)\mathbb{P}(Y_{k,n} = i - l + 1) & \text{if } i > 0, \\ \mathbb{P}(X_{k-1,n} = 0) + \mathbb{P}(X_{k-1,n} = 1)\mathbb{P}(Y_{k,n} = 0) & \text{if } i = 0. \end{cases} \tag{1.2}$$

In order to derive results for X_g, we now need to turn to Probability Generating Functions (PGFs) to which we give a brief introduction before continuing the study of the FCTL queue.

Probability Generating Functions (PGFs)

PGFs are one of the classical and essential tools in the study of stochastic processes. They are discussed in several classical text books, such as [86, Chapter 5], [102, Chapter 1], and [189, Appendix C]. They are used in a wide variety of applications, such as in the study of random walks, see e.g. [86, Chapter 5.3]; in branching processes, see e.g. [86, Chapters 5.4 and 5.5] and [102, Chapter 8]; and in queueing theory, see e.g. [55]. As such, they play an important role in traffic-light models.

A PGF of a discrete, non-negative random variable A is defined as $A(z) = \mathbb{E}[z^A] = \sum_{k=0}^{\infty} P(A = k)z^k$. From PGFs one is able to obtain probabilities and moments from the associated random variable and as such

they prove useful in various situations as PGFs might be easier to obtain than the distribution of a random variable itself. If one is interested in obtaining probabilities, one can use a PGF to compute probabilities, e.g. we have that $\mathbb{P}(A = k) = A^{(k)}(0)/k!$. Similarly, we might obtain moments of a random variable from its PGF, e.g. $\mathbb{E}[A] = A'(1)$. If differentiation of $A(z)$ is difficult, one might resort to numerical schemes like the ones in [2, 51, 52].

One of the advantages of PGFs is that they enable us to easily manipulate sums of independent random variables: if A and B are independent and if $A(z)$ and $B(z)$ are their PGFs, then the PGF of $A+B$ is $A(z)B(z)$. As such, they are easy to work with, e.g. when one faces the recursions as in Equations (1.1) and (1.2).

Let us introduce several PGFs: let $Y_{k,n}(z)$ be the PGF of $Y_{k,n}$, $X_{k,n}(z)$ the PGF of $X_{k,n}$, and $X_g(z)$ the PGF of X_g.

We now have everything that we need in order to derive $X_{g,n}(z)$ in terms of $X_{g,n-1}(z)$. We have

$$\begin{aligned}
X_{g,n}(z) &= \sum_{i=0}^{\infty} \mathbb{P}(X_{g,n} = i) z^i \\
&= \sum_{i=1}^{\infty} \sum_{l=1}^{i+1} \mathbb{P}(X_{g-1,n} = l)\mathbb{P}(Y_{g,n} = i - l + 1) z^i + \mathbb{P}(X_{g-1,n} = 0) + \\
&\quad \mathbb{P}(X_{g-1,n} = 1)\mathbb{P}(Y_{g,n} = 0) \\
&= \sum_{l=1}^{\infty} \mathbb{P}(X_{g-1,n} = l) z^{l-1} \sum_{i=l-1}^{\infty} \mathbb{P}(Y_{g,n} = i - l + 1) z^{i-l+1} + \mathbb{P}(X_{g-1,n} = 0) \\
&= \sum_{l=1}^{\infty} \mathbb{P}(X_{g-1,n} = l) z^{l} \frac{Y_{g,n}(z)}{z} + \mathbb{P}(X_{g-1,n} = 0) \\
&= \big(X_{g-1,n}(z) - X_{g-1,n}(0)\big) \frac{Y_{g,n}(z)}{z} + X_{g-1,n}(0) \\
&= X_{g-1,n}(z) \frac{Y_{g,n}(z)}{z} + X_{g-1,n}(0) \left(1 - \frac{Y_{g,n}(z)}{z}\right),
\end{aligned}$$

which is in accordance with [206, Equation (7)].

Iterating further, we get that

$$X_{g,n}(z) = X_{c,n-1}(z) \prod_{i=1}^{g} \left(\frac{Y_{i,n}(z)}{z}\right) + \sum_{i=0}^{g-1} X_{i,n}(0) \left(1 - \frac{Y_{i+1,n}(z)}{z}\right) \prod_{k=i+2}^{g} \left(\frac{Y_{k,n}(z)}{z}\right), \quad (1.3)$$

where $X_{0,n}(0)$ is to be understood as $X_{c,n-1}(0)$. We also derive $X_{c,n}(z)$ in terms of $X_{g,n}(z)$:

$$\begin{aligned} X_{c,n}(z) &= \sum_{i=0}^{\infty} \sum_{l=0}^{i} \mathbb{P}(X_{c-1,n} = l)\mathbb{P}(Y_{c,n} = i-l) z^i \\ &= \sum_{l=0}^{\infty} z^l \mathbb{P}(X_{c-1,n} = l) \sum_{i=l}^{\infty} \mathbb{P}(Y_{c,n} = i-l) z^{i-l} \\ &= \sum_{l=0}^{\infty} z^l \mathbb{P}(X_{c-1,n} = l)\, Y_{c,n}(z) \\ &= X_{c-1,n}(z) Y_{c,n}(z) = \ldots = X_{g,n}(z) \prod_{i=g+1}^{c} Y_{i,n}(z). \end{aligned} \tag{1.4}$$

We note that, in steady state, $X_{g,n} \stackrel{d}{=} X_{g,n-1}$, i.e. the steady-state queue length at the end of the green period in cycle n should be the same in distribution as in cycle $n-1$. If moreover the $Y_{k,n}$ are independent and identically distributed, if we define $Y(z)$ to be the PGF of Y where $Y \stackrel{d}{=} Y_{k,n}$, and combine Equations (1.3) and (1.4), we obtain that

$$X_{g,n}(z) = X_{g,n-1}(z) \frac{Y(z)^c}{z^g} + \sum_{i=0}^{g-1} X_{i,n}(0) \left(1 - \frac{Y(z)}{z}\right) \left(\frac{Y(z)}{z}\right)^{g-i-1}.$$

This leads to the following steady state expression for $X_g(z)$:

$$X_g(z) = \frac{z^g \sum_{i=0}^{g-1} X_i(0) \left(1 - \frac{Y(z)}{z}\right) \left(\frac{Y(z)}{z}\right)^{g-i-1}}{z^g - Y(z)^c}, \tag{1.5}$$

where $X_0(0)$ is to be understood as $X_c(0)$ and with $X_i(0)$, $i = 0, \ldots, g-1$, the steady-state probability that the queue is empty at the end of slot i.

The only thing left to do is to find the unknowns $X_i(0)$ for $i = 0, \ldots, g-1$. We might employ the zeros within the unit circle of $z^g - Y(z)^c$, the denominator of $X_g(z)$, and the fact that we know that $X_g(z)$ is analytical within the unit circle. We thus need the numerator to be zero if $z^g - Y(z)^c = 0$. One can use Rouché's theorem, see e.g. [6], to show that there are g zeros of $z^g - Y(z)^c$ within the unit circle if the stability condition is satisfied. This leads to $g-1$ equations for the numerator of $X_g(z)$, as one zero ($z = 1$) leads to a trivial equation. As $X_g(z)$ is a PGF it should satisfy $X_g(1) = 1$, which gives one more equation. Using l'Hôpital's

rule, this equation reduces to

$$\sum_{i=0}^{g-1} X_i(0) = \frac{g - c\mathbb{E}[Y]}{1 - \mathbb{E}[Y]}.$$

Now, we have obtained a set of g linear equations in terms of g unknowns, which can be solved to obtain the $X_i(0)$. In [206], it is shown that the $X_i(0)$ are the solution of a Vandermonde system. Plugging the $X_i(0)$ into Equation (1.5), we have finished the characterization of $X_g(z)$. Instead of obtaining X_g, we thus have obtained $X_g(z)$. Nevertheless, this enables us to find the associated probabilities and moments of the overflow queue X_g by means of differentiation as indicated before or by means of numerical inversion schemes as can be found in e.g. [2, 52].

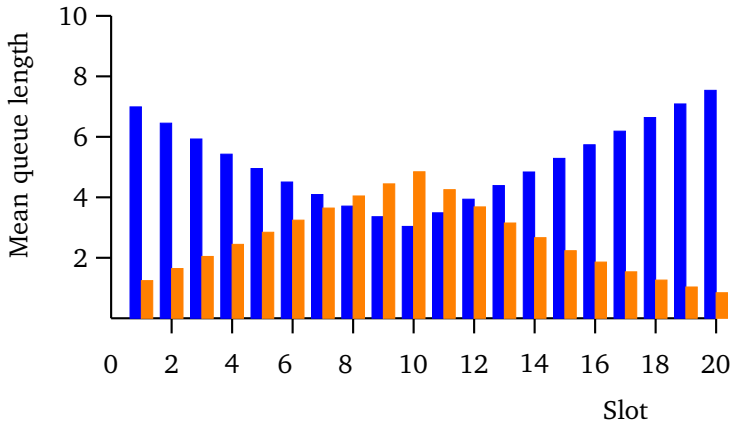


Figure 1.4: Mean queue length at the end of slots 1 to 20 for two FCTL queues with $2g = c = 20$. The blue bars correspond to a queue which starts with a green period in slot 1 and with Poisson arrivals in each slot with mean 0.45. The orange bars correspond to a queue which starts with a red period in slot 1 and with Poisson arrivals in each slot with mean 0.4.

We give a brief illustration of some results that can be obtained with the developed framework for the FCTL queue in Figure 1.4. We see the steady-state mean queue length at the end of slot i for $i = 1, \ldots, 20$ for two FCTL queues with $2g = c = 20$. The queue corresponding to the blue bars starts with a green period

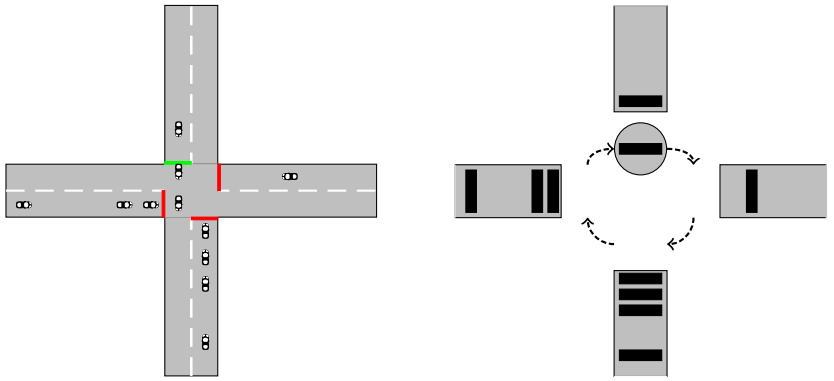
and for slots 1 to 10 the mean queue length decreases. After slot 10, the traffic light turns red and the mean queue length starts increasing again. The number of arrivals per slot is distributed according to a Poisson distribution with mean 0.45. The number of arrivals per slot in the other queue, corresponding to the orange bars, is distributed according to a Poisson distribution with mean 0.4. This queue starts with a red period in slot 1 and the mean queue length thus increases (as opposed to the other queue). After slot 10, the traffic light for this queue turns green and the mean queue length starts decreasing. The patterns in Figure 1.4 are general: increasing queues for red traffic lights and decreasing queues for green lights. The opposing effect of the decrease/increase in the mean queue length for different queues as in Figure 1.4 is also a quite universal phenomenon, as a green light for one queue usually implies a red traffic light for a queue with a conflicting stream of vehicles, see e.g. also Figure 1.2.

1.3.2 Polling models

As discussed before, the analysis for vehicle actuated systems is significantly more complex than for the FCTL queue due to the multidimensionality of the problem. From a queueing perspective, the most typical feature of traffic intersections with vehicle-actuated strategies is that they consist of multiple queues, but only one queue (or, more generally, a subset of all queues) can be served simultaneously. In queueing theory, there is a class of models that exhibits exactly this feature, and these are called polling models. The use of polling models in studies of road traffic and the fact that (ideas stemming from) polling models will often be used later on in this thesis, are the main reasons why we discuss polling models separately in this introduction.

In Figure 1.5, we illustrate both a simplified model for a traffic-light controlled intersection and a polling model. Looking at Figure 1.5, we immediately see the resemblance between the two models and as such, polling models might be leveraged to study traffic-light controlled intersections.

A polling model is a queueing model with one server which serves several queues according to some general rules. Most often, it is assumed that the server serves one queue at a time and after some time switches to the next queue, as can be seen in Figure 1.5(b). The moment that the server initiates a switch is determined by the *service discipline*. Such a switch might happen after a queue has become empty (exhaustive service); after either a queue has become empty or a fixed maximum number, k, of customers has been served (k-limited service); after all customers have been served that were waiting in the queue upon arrival of the server at the queue (gated service); and many more variants exist.



(a) (b)

Figure 1.5: Graphical representations of (a) a simplified model for an intersection with 4 lanes and a traffic light and (b) a polling model with 4 queues and a single server. Note that vehicles driving towards the intersection in (a) are displayed as arriving entities in the queue in (b). Some of those vehicles we count as in the queue, e.g. the top three in the bottom queue, whereas the fourth vehicle is not yet considered to be in the queue.

The right combination of arrival processes, service processes, switchover processes, and the service discipline enables one to realistically model all kinds of processes, among which are traffic lights. Indeed, the use of polling models in the study of traffic lights is evident in this respect as the incoming lanes represent the queues and the intersection itself represents the server. A recent review on further applications of polling models can be found in [18], an overview of commonly used techniques in [215], and a recent general overview in [24].

As we have already given an overview of traffic lights with a vehicle-actuated control, we refrain from giving more in-depth studies which use polling models to study traffic-lights and refer the reader to Subsection 1.2.2. We continue this subsection with a discussion on some commonly used concepts and techniques in the study of polling models, with a special emphasis on methods and techniques that will be of further use later on in this thesis.

Analysis techniques for polling models

Some polling models lend themselves for a relatively straightforward and exact analysis, whereas for other polling models there are no known methods to derive exact results. One of the crucial factors that differentiates these two sets of

models is the service discipline. In more detail, service disciplines that satisfy the so-called branching property often lend themselves for an exact analysis, whereas service disciplines that do not, are often intractable. The branching property was independently discovered by Fuhrmann [79] and Resing [168]. It states that service disciplines like the exhaustive and gated discipline satisfy the following principle: each time the server arrives at or departs from a queue (the so-called polling epochs), the joint number of customers at all queues can be represented by a multi-type branching process with immigration. This seems to be a crucial feature in being able to analyze a polling model in a relatively straightforward way. The exhaustive and gated service discipline are important service disciplines that satisfy the branching property, whereas the k-limited discipline is an important one (especially in traffic-light engineering) that does not satisfy the branching property.

The branching property enables one to relate the queue-length distributions at polling epochs to one another. Iterative use of these relations leads to an implicit functional equation for the PGF of the joint queue-length distribution at polling epochs. Although this functional equation usually does not allow for an explicit expression of the PGF, it allows for explicit computations of e.g. moments. Through numerical inversion of the PGF, the distribution can also be obtained numerically. This technique of relating the queue-length distributions at polling epochs to one another is sometimes referred to as the buffer occupancy method, see e.g. the paper by Levy and Sidi [123]. For an elaborate exposition of the buffer occupancy method, we refer to [16, Section 2.2].

If the branching property is not satisfied, there is no general framework available for the study of the queue-length distribution in polling models. Usually, intricate types of analysis are needed if an analysis is possible at all. An example of the latter is a method based on boundary value problems from mathematical physics, like the Riemann and Riemann-Hilbert boundary value problem. This method might be leveraged to obtain, e.g., the joint PGF of the queue-length distribution for a two-queue polling model with 1-limited service [29]. Unfortunately, boundary value problems can only be solved for a very limited set of polling models. For a general treatment of boundary value problems and their applications to queueing theory, we refer the interested reader to [57, 73].

As a last technique, we mention an approximation technique for polling models based on light- and heavy-traffic limits. This technique was first applied to polling models in [22], building on ideas developed in [166]. Basically, the light-traffic and heavy-traffic limit of the polling model at hand are established and those are used to approximate, e.g., the mean waiting time in the polling model by means of a simple function. A big advantage is that the approximation

results in a closed-form expression, which can subsequently be used for e.g. optimization purposes which is often not possible when using exact expressions. Typically, the light-traffic limit is relatively easy to derive. For service disciplines satisfying the branching property, heavy-traffic limits have been derived as well, see e.g. [54, 200]. When the service discipline violates the branching property, it is usually hard to obtain the heavy-traffic limit behavior. Some work in this respect has been done by Boon and Winands, who focus on k-limited polling models, see [20, 21]. They derive the heavy-traffic limit for k-limited polling models with asymmetric loads, leaving the symmetric case for future research. It seems difficult to come up with a unified approach for the heavy-traffic analysis of polling models that do not satisfy the branching property.

1.3.3 Road traffic simulators

Because of the complicated nature of road traffic and the limiting possibilities of e.g. queueing models to capture road traffic phenomena, many traffic simulators have been developed. Simulators might vary in the type of interactions they allow for, ranging from macroscopic to microscopic. The most interesting ones for our purposes are microscopic simulators. They explicitly model all kinds of vehicle-to-vehicle interactions, enabling a realistic and detailed study of the behavior of every single vehicle. These interactions are one of the core advantages of microscopic simulators, as they enable the simulator to obtain a good reflection of reality. A disadvantage of microscopic traffic simulators is that they are computationally expensive. Moreover, due to randomness in the interaction between vehicles, one often needs to execute several simulations to dampen the effect of randomness in order to get accurate results, which is a further complicating factor. We discuss one (microscopic) traffic simulator in more detail, SUMO, as we will use it in some of the subsequent chapters. Other well-known simulators are VISSIM and Aimsun. For a further overview of various traffic simulators we refer to [160].

SUMO (Simulation of Urban MObility) [129] is a popular, free, and open-source traffic simulator. SUMO is a microscopic traffic simulator with many options and possibilities that enables one to set up a realistic simulation for many case studies. It is e.g. possible to replicate simulations by fixing the seed for the random number generator and one might model many different arrival distributions of vehicles. Confidence intervals can be obtained by e.g. running the same simulation with different seeds and obtaining performance measures for each run, which can then be used to construct such confidence intervals.

Other possibilities include the introduction of several vehicle types (cars, buses, trucks, etc.), different behavior of drivers, and various traffic-light strategies.

We will use SUMO to study the steady-state behavior of various models that we consider. In this respect, we merely use SUMO as a realistic benchmark for current-day traffic, as it is difficult to test our models on-site in a real-life setting. SUMO is a cheap and easy-to-use alternative. Moreover, SUMO is known to produce an accurate representation of reality and it automatically outputs some of the performance characteristics that we are interested in, which are further reasons to choose SUMO as a simulation tool. Besides the aforementioned, SUMO also offers a nice visualization, as can be seen in Figure 1.6 and some output of SUMO is visualized in Figure 1.7. We see some very detailed information per vehicle in Figure 1.7 which shows the possibilities of SUMO in analyzing traffic performance. Some of the more interesting output variables for our purposes are the waiting time (time spent (almost) stationary) of vehicles and the time loss, which corresponds to the delay experienced by vehicles. Note that much more output might be generated, see also [82] for some examples.

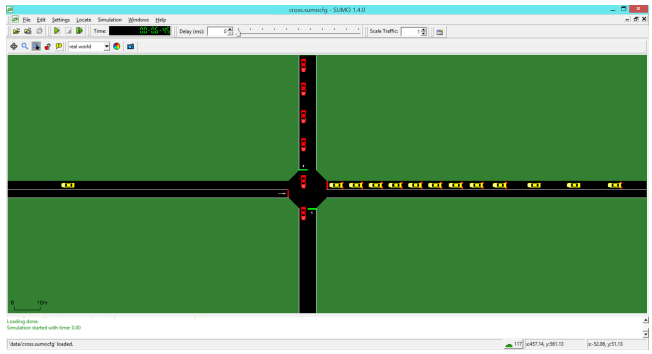


Figure 1.6: A snapshot of a SUMO simulation.

Having discussed some of the advantages of SUMO, we now turn to limitations of SUMO. As SUMO is a microscopic traffic simulator (which is one of its key strengths), it models every vehicle individually with varying driver characteristics. As such, there might be a considerable amount of randomness in the behavior of vehicles and thus in the outcome of a simulation, a common drawback of microscopic traffic simulators. This is something that needs to be accounted for. Moreover, the individual modeling of vehicles takes up a lot of computational power, which makes SUMO slow compared to other (non-

id	depart	departPos	departDelay	arrivalLane	arrivalSpeed	waitingTime	waitingCount	timeLoss
horizontal.0	100	5.1	0	2fi_0	10.78	0	0	15.16
horizontal.1	110	5.1	0	2fi_0	10.45	0	0	14.86
horizontal.2	120	5.1	0	2fi_0	10.54	0	0	14.13
vertical.0	103	5.1	3	3fi_0	11.04	31	1	50.99
vertical.1	113	5.1	3	3fi_0	10.2	22	1	42.96
vertical.2	123	5.1	3	3fi_0	10.76	13	1	34.79
vertical.3	133	5.1	3	3fi_0	9.83	2	1	26.8
vertical.4	143	5.1	3	3fi_0	10.94	0	0	18.84
vertical.5	153	5.1	3	3fi_0	10.07	0	0	14.77
vertical.6	163	5.1	3	3fi_0	10.98	0	0	14.55
horizontal.3	130	5.1	0	2fi_0	10.85	47	1	66.73
horizontal.4	140	5.1	0	2fi_0	10.27	40	1	58.77
horizontal.5	150	5.1	0	2fi_0	11.02	31	1	51.29
horizontal.6	160	5.1	0	2fi_0	11.09	22	1	43.67
horizontal.7	170	5.1	0	2fi_0	9.82	14	1	36.04
horizontal.8	180	5.1	0	2fi_0	9.92	4	1	28.06
horizontal.9	190	5.1	0	2fi_0	10.19	0	0	20.08
horizontal.10	200	5.1	0	2fi_0	9.97	0	0	15.92
horizontal.11	210	5.1	0	2fi_0	11.08	0	0	14.92
vertical.7	173	5.1	3	3fi_0	10.06	50	1	70.74

Figure 1.7: Some output of SUMO.

microscopic) simulation methods.

SUMO can be leveraged in various other ways as well. SUMO can for example be used for the verification of certain models, but is also used for real-time traffic predictions by the company Sweco in their smart traffic application. This shows that SUMO is indeed considered to be a good substitute for reality, both by academicians and practitioners.

1.4 Main contributions and outline of this thesis

To close the introduction, we give an overview of the remainder of this thesis and its main contributions to the existing literature. We give a chapter-by-chapter overview.

Chapter 2 provides an alternative analysis to the standard approach for obtaining the PGF of the distribution of the overflow queue of the FCTL queue. Instead of having to find roots and to solve a set of linear equations, as is the case in Subsection 1.3.1 and in [64,206], we provide a complex contour-integral expression for the PGF of the distribution of the overflow queue. This expression does not require any roots to be found and also avoids the need to solve a set of linear equations. Advantages and disadvantages of the new expression are investigated and discussed. This chapter is based on [231].

The contour-integral expression for the PGF of the overflow queue for the FCTL queue enables us to find a heavy-traffic scaling of the FCTL queue, which is one of the main contributions in Chapter 3. The scaling that we propose, relates the green period, g, to the cycle length, c, in the following way

$$g = c\mu + \beta\sigma\sqrt{c},$$

where μ is the mean arrival rate per slot, σ the standard deviation of the number of arrivals per slot, and $\beta > 0$ a parameter that can be chosen freely. This scaling gives rise to a Halfin-Whitt type of regime, see e.g. [89, 208]. We derive the convergence, after properly scaling, of the overflow queue to the so-called maximum of a Gaussian random walk with negative drift. The convergence result allows us to formulate approximations for e.g. the mean overflow queue, which allows us to construct green-time allocation problems which are solved to optimality. A benefit of our approach is that the optimal green times are easy to compute and explain. This chapter is based on [232].

In Chapter 4 we use the intuition that we developed in Chapter 3 and apply a similar scaling rule for the green periods to various traffic-light control strategies. We demonstrate by means of simulation that the queueing process for several vehicle-actuated strategies with limited control exhibits a similar type of Halfin-Whitt behavior as the FCTL queue in Chapter 3. We resort to simulation techniques, because the strategies we study are currently considered to be analytically intractable. This chapter is based on [234].

In Chapter 5 an extension of the FCTL queue is introduced. Motivated by simulation work in [96], we investigate a traffic light with both turning and straight-going traffic. The turning vehicles might be blocked by pedestrians who receive a green light at the same time as the turning vehicles. A turning vehicle is only blocked if there are pedestrians crossing, which in turn causes all vehicles behind the turning vehicle to be blocked as well. As this is a realistic scenario at various intersections in practice, it is important to gain insight into the influence of such pedestrian crossings and we formulate a tractable extension of the FCTL queue to close this gap in the literature. Moreover, we allow for a general number of delayed vehicles departing, say m, in each slot, whereas in the traditional FCTL queue at most one delayed vehicle departs in each slot. As we show in Chapter 5, this is another relevant extension of the FCTL queue. Chapter 5 is based on [237].

Motivated by k-limited polling models, their intractability, and their relevance in the study of vehicle-actuated traffic lights, we propose a novel approximation scheme for k-limited polling models in Chapter 6. It turns out

that many multidimensional queueing models can be approximated with our scheme, as long as a kernel-type functional equation can be derived for the PGF of the steady-state joint queue-length distribution. The scheme uses this functional equation along with several roots of the so-called kernel to approximate a finite number of probabilities. Our scheme leads to an approximation of the joint steady-state queue-length distribution. Next to k-limited polling models, we also study some specific traffic-light models for which no exact results have been derived as far as we are aware. Chapter 6 is based on [235].

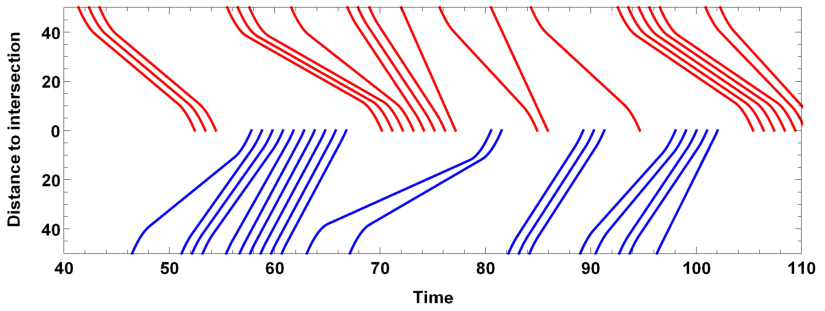


Figure 1.8: Two streams of vehicles (depicted in red and blue) approaching an intersection located at the middle of the figure. It can be seen that vehicles are grouped together in platoons and that vehicles start decelerating early (if they are delayed) and start accelerating in such a way that they cross the intersection at maximum speed. In this example, the vehicles minimize the amount of applied acceleration.

As argued before, it is not inconceivable that self-driving vehicles will occupy the roads in the near future. Anticipating the introduction of such vehicles, we turn to platoon forming algorithms in Chapter 7. We investigate and compare several traffic-light or intersection-access control strategies, where we focus on obtaining approximations for mean performance measures. The derived approximations are different in nature than the ones derived in Chapter 6 and are based on light- and heavy-traffic limits, similar to those in e.g. [22]. We also investigate how vehicles should approach the intersection and consider two strategies: one minimizes spillback to other queues, while the other minimizes the amount of acceleration. We obtain closed-form solutions for both strategies, adding to the practical value of our schemes. The key ideas used in Chapter 7 are visualized in Figure 1.8. Chapter 7 is based on [236].

Chapter 8 relates to joint work with De Verkeersonderneming, a Dutch com-

pany aimed at improving traffic flow and mobility. De Verkeersonderneming was looking for an algorithm to identify special days at highways. Those special days correspond to days with *both* a high traffic flow and the absence of congestion. Based on common notions in traffic engineering, like the fundamental diagram, and regression techniques, we are able to identify those special days. The algorithm has been tested at several sites and has proven to give reliable results. Chapter 8 is based on [233].

We close this thesis with a general discussion and conclusion in Chapter 9. We also briefly touch upon topics for future research.

Chapter 2

Pollaczek contour integrals for the Fixed-Cycle Traffic-Light queue

2.1 Introduction

In Subsection 1.3.1, we have provided a detailed model description and analysis of the Fixed-Cycle Traffic-Light (FCTL) queue. The final expression for the Probability Generating Function (PGF) of the distribution of the overflow queue or the queue length at the end of the green period, denoted with $X_g(z)$, involves several a priori unknown parameters. Those can be found using roots located within the unit circle from the denominator of $X_g(z)$ and subsequently solving a set of linear equations. As the authors in [146] show, the *mean* overflow queue can be obtained without any root finding and solving sets of linear equations. In this chapter we extend those results, showing that root-finding can be avoided when obtaining the PGF $X_g(z)$ by employing several basic notions from complex analysis, among which Cauchy's residue theorem, see e.g. [37, Chapter 2.7]. In the end, we obtain a Pollaczek-type contour-integral expression for $X_g(z)$, which can be evaluated numerically.

The type of contour integral reminds of early results on the classical single-server queue analyzed in ground-breaking work of Pollaczek, see [1,55,100] for historical accounts. A similar approach is taken in [101], where subsequently

heavy-traffic results for the Bulk-Service Queue (BSQ) are derived. Our results for the FCTL queue can easily be extended to a more general set of models, including the BSQ and the models considered in [146], see also Subsection 2.3.2.

Besides the absence of the need to find roots to obtain $X_g(z)$, another major advantage of the new representation is that the complex contour integral allows for different types of analysis than the representation based on the roots. E.g., we will employ the structure of the contour integral in Chapter 3 to obtain a heavy-traffic scaling based on the Halfin-Whitt regime, which would have been very difficult to obtain with the roots-based representation. A disadvantage of the complex contour integral is that one needs to evaluate the integral numerically, but also the roots have to be computed numerically in most cases.

Our main contributions can be summarized as follows:

(i) We provide a novel contour-integral expression for the PGF of the steady-state queue-length distribution at the end of the green period.

(ii) We show that this contour-integral expression can be used to obtain several interesting performance measures.

(iii) We show that a contour-integral type of expression can be derived for a much larger class of queueing models than just the FCTL queue.

Chapter outline

We continue this chapter with a brief recap of the analysis of the FCTL queue in Section 2.2, as this provides an excellent starting point for the derivation of the main result in this chapter. Subsequently, we present the new analysis which leads to a contour-integral expression in Section 2.3 after which we provide contour-integral expressions for a more general set of models. In Section 2.4 we study some numerical examples and reflect on the differences for the computations between the root-based and contour-integral formulas. The proof of the contour-integral representation is presented in Section 2.5 and we wrap up with some conclusions in Section 2.6.

2.2 Standard solution for the FCTL queue

We present a short exposition of the derivation of the steady-state overflow queue, $X_g(z)$. As in Subsection 1.3.1, we define Y to be the number of arrivals during one slot and define $Y(z) = \mathbb{E}[z^Y]$. Moreover, we denote with $A(z)$ the PGF

of the distribution of all arrivals during a cycle, i.e. $A(z) = Y(z)^c$. We assume $\mathbb{P}(Y = 0) > 0$, $Y'(1) < 1$, $A'(1) < g$, and $Y(z)$ to be analytic in a region $|z| < R$ with $R > 1$ and R maximal. Then, it can be shown, as detailed in Subsection 1.3.1 and as in [64, 206], that

$$X_g(z) = \frac{(z - Y(z)) \sum_{k=0}^{g-1} q_k z^k Y(z)^{g-1-k}}{z^g - A(z)}. \tag{2.1}$$

This expression still contains g unknowns $q_0, \ldots, q_{g-1}$, with q_k representing the probability that the queue empties before or during slot k, where slot 0 is to be understood as slot c. We thus have $X_k(0) = \mathbb{P}(X_k = 0) = q_k$, with X_k the distribution of the queue length at the end of slot k and with $X_k(z)$ its PGF. The q_k's can be found by exploiting the analytic properties of PGFs as explained in Subsection 1.3.1. With Rouché's theorem, it can be shown that the denominator of (2.1) has g zeros on or within the unit circle $|z| = 1$. Because a PGF is well-defined in $|z| \leq 1$, the numerator of $X_g(z)$ should vanish at each of the zeros. This gives g equations. One of the zeros equals 1, and leads to a trivial equation. However, the normalization condition $X_g(1) = 1$ provides an additional equation.

This summarizes the highest level of general development for the analysis of the FCTL queue: transform techniques yield an expression for $X_g(z)$ that in order to be evaluated demands finding $g-1$ roots in the complex plane of the function $z^g = A(z)$ and solving a set of g linear equations. In the next section, we show that the finding of the $g-1$ roots and solving the set of linear equations can be avoided.

2.3 Main results

In Subsection 2.3.1 we present the contour-integral representation for the PGF of the overflow queue for the FCTL queue. In Subsection 2.3.2 we derive similar contour-integral expressions for several generalizations of the FCTL queue, some of which are also considered in [146].

2.3.1 Standard FCTL queue

We now turn to the alternative expression for $X_g(z)$ which is based on a contour integral. Here is the main result in this chapter, continued with a sketch of the proof (the proof is deferred to Section 2.5):

Theorem 2.1 *There is an $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0)$ and for all $|w| < 1+\epsilon$,*

$$X_g(w) = \exp\left(\frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \frac{Y'(z)z - Y(z)}{z - Y(z)} \frac{w - Y(w)}{zY(w) - wY(z)} \ln\left(1 - \frac{A(z)}{z^g}\right) \mathrm{d}z\right), \quad (2.2)$$

with principal value of the logarithm.

Here, ϵ_0 satisfies the inequality $\epsilon_0 < \min\{t_0, R_0\}$, where $t_0 = \sup\{t \in \mathbb{R}_+ | Y'(t)t - Y(t) \leq 0\}$ and R_0 is the unique root of $Y(z) = z$ in $(1, \infty)$.

Remark 2.1 *Equation* (2.2) *is essentially equivalent with*

$$X_g(w) = \exp\left(\frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right) \frac{(z^g - A(z))'}{z^g - A(z)} \mathrm{d}z\right), \quad (2.3)$$

with principal value of the logarithm, except that the validity range is more delicate due to the more complicated argument of the ln *in Equation* (2.3)*. Equation* (2.2) *follows upon manipulating Equation* (2.3) *using partial integration (details can be found in Section 2.5).*

Sketch of the proof

The proof of Theorem 2.1 finds a way to go from representation (2.1) to contour integrals. A significant start in this direction was made by [146], who rewrote Equation (2.1) as

$$X_g(z) = \frac{(z - Y(z))z^{g-1} \sum_{k=0}^{g-1} q_k \left(\frac{Y(z)}{z}\right)^{g-1-k}}{z^g - A(z)}.$$

Then denote the g roots of $z^g = A(z)$ on and within the unit circle by $z_0 = 1, z_1, \ldots, z_{g-1}$. Now here is where the authors in [146] took an eye-opening step: instead of using the g roots in the traditional manner for finding the unknowns q_k and completing the transform Equation (2.1), use these roots for factorizing the numerator of Equation (2.1). Notice that this cannot be done immediately, because interpreted as a function of z, the numerator is by no means a polynomial of degree g or less. However, by treating the function $Y(z)/z$ as a variable itself, the summation in the numerator is a polynomial of degree $g-1$ and can be factorized as

$$\sum_{k=0}^{g-1} q_k \left(\frac{Y(z)}{z}\right)^{g-1-k} = q_0 \prod_{k=1}^{g-1} \left(\frac{Y(z)}{z} - \frac{Y(z_k)}{z_k}\right), \quad (2.4)$$

using that $X_g(z)$ is well-defined in the disk $|z| \leq 1$, that $z_1, \ldots, z_{g-1}$ are roots of the denominator and therefore also should be roots of the numerator, and that $Y(z)/z$ is injective (see Section 2.5). After normalization using $X_g(1) = 1$, the factorization in Equation (2.4) leads to the representation

$$X_g(z) = \frac{g - A'(1)}{z^g - A(z)} \cdot \frac{z - Y(z)}{1 - Y'(1)} \cdot z^{g-1} \prod_{k=1}^{g-1} \frac{Y(z)/z - Y(z_k)/z_k}{1 - Y(z_k)/z_k}. \tag{2.5}$$

Our proof proceeds by interpreting Equation (2.5) as the outcome of Cauchy's residue theorem, the classical tool from complex analysis to evaluate line integrals of analytic functions over closed curves. An important step is to write

$$\ln\left(z^{g-1} \prod_{k=1}^{g-1} \frac{Y(z)/z - Y(z_k)/z_k}{1 - Y(z_k)/z_k}\right) = \sum_{k=1}^{g-1} \ln\left(\frac{zY(z_k) - z_k Y(z)}{Y(z_k) - z_k}\right), \tag{2.6}$$

and to regard Equation (2.6) as the sum of residues at $z = z_k$, where we used that the z_k are either real or come in complex conjugates. To construct an analytic function that, in conjunction with Cauchy's theorem and the closed curve $|z| = 1 + \epsilon$, returns Equation (2.6) and has singularities at $z_1, \ldots, z_{g-1}$, leads us to consider the integrand in Equation (2.3). Here, the logarithmic function

$$\ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right) = \ln\left(\frac{Y(z)/z - Y(w)/w}{Y(z)/z - 1} w\right) \tag{2.7}$$

follows from Equation (2.6) and the singularities with appropriate residues are created through $(z^g - A(z))'/(z^g - A(z))$. First, we derive useful properties of the function $Y(z)/z$ present in Equation (2.7) such as injectivity in a sufficiently large region. Then, after careful consideration of the analytic properties of the integrand in Equation (2.3), we show that Cauchy's theorem gives Equation (2.5) from which Equation (2.3) follows. As mentioned before, Equation (2.2) is obtained by manipulating Equation (2.3) using partial integration. The formal proof of Theorem 2.1 is presented in Section 2.5.

Historical notes

Integrals of this sort go a long way back in the history of queueing theory and were first found in the ground-breaking work of Pollaczek on the classical single-server queue (see [1, 55, 100] for historical accounts). Let us point out the connection to the well known Pollaczek type integral for the BSQ [101]. The

analysis of the FCTL queue would be greatly simplified if all vehicles are delayed [198]. In that case we obtain a standard stochastic recursion driven by independent and identically distributed (i.i.d.) random variables and the FCTL queue reduces to the classical BSQ, a special case of the more general single-server queue investigated by Pollaczek. Let X_b denote the steady-state queue length of the BSQ, defined as the solution of the stochastic equation

$$X_b \stackrel{d}{=} \max\{X_b + A - g, 0\}.$$

Pollaczek's result then says that (see [101] for a direct derivation)

$$X_b(w) = \exp\left(\frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \ln\left(\frac{w-z}{1-z}\right)\frac{(z^g - A(z))'}{z^g - A(z)}\,\mathrm{d}z\right) \tag{2.8}$$

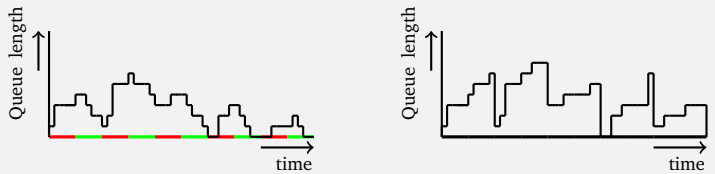
holds when $|w| < 1+\epsilon$ with ϵ positive and bounded by some constant. Observe the striking similarity with Equation (2.3). While the FCTL queue is harder to analyze than the BSQ, the two contour-integral representations Equation (2.3) and Equation (2.8) only differ in the logarithmic function. We find this quite surprising, particularly because there seems no way to interpret the FCTL queue as a reflected random walk (that is, a recursive structure with i.i.d. increments), while in the literature so far this seems to be a prerequisite for establishing Pollaczek-type contour integrals. Do observe that Equation (2.8) is valid in an area that includes the unit disk while Equation (2.3) is guaranteed only in an open set containing $[0,1]$, see Section 2.5. This objection does not hold against the representation in Equation (2.2) of $X_g(w)$.

The bulk-service queue and a comparison with the FCTL queue

The BSQ is a popular approximation of the FCTL queue [198]. The BSQ can be described as follows: there are arrivals according to a generally distributed random variable A and after a randomly distributed time B, a randomly distributed number, G, of customers are cleared from the queue after which the process repeats itself. When we choose $B = c$, $G = g$, and A to be the number of arrivals in a period of length c, it is evident that the BSQ might serve as an approximation to the FCTL queue, where the BSQ accumulates all departures at a single time whereas the FCTL queue has departures throughout the cycle.

To compare the FCTL queue and the BSQ, we assume for the moment that the arrivals at both queues are identical and that they start with the

same number of entities in the queue. If the queue is non-empty at the end of the green period for the FCTL queue or after a service completion in the BSQ, there is no difference between the number of departures in the FCTL queue and the BSQ and in such a case they thus behave identically. The only case when there might be a difference between the number of departures between the two queueing models is when the FCTL queue becomes empty during the green period: then there might be more than one departure per slot due to the FCTL assumption (and thus the total number of departures in a cycle might be larger than g). This is not possible in the BSQ. The only way in which the BSQ and the FCTL queue thus deviate when having the same arrivals and starting at the same level, is when the FCTL queue becomes empty during the green period. The FCTL queue, in some sense, thus provides a more detailed version of the BSQ with more complicated within-cycle-dynamics. To illustrate this, we provide sample paths for the BSQ and the FCTL queue in Figure 2.1.



(a) (b)

Figure 2.1: A sample path for the FCTL queue (a) and the BSQ (b). The colors on the horizontal axis indicate whether the traffic light is green or red. The arrivals in both figures are the same and we can clearly see the difference between the departure process in the FCTL queue, with within-cycle dynamics, and the BSQ without within-cycle dynamics: departures only occur once per cycle. Careful inspection of the sample paths in (a) and (b) tells us that the queue length at the departure moments in the BSQ is the same as in the FCTL queue for the considered sample paths, except for the one-but-last departure moment in the BSQ: there were some arrivals that could leave in the FCTL queue because of an empty queue and a green traffic light, which could not all depart in the BSQ at the next departure moment.

When the number of arrivals per slot in the FCTL queue is at most one, so for Bernoulli arrivals, there is *no* difference between the FCTL queue and the BSQ. This is because there are at most g departures in the FCTL queue during a green period (even when the queue is empty there is at most one departure per slot, because of the Bernoulli arrivals per slot), while in the corresponding BSQ there are also at most g departures. So, the BSQ and the FCTL queue are equivalent in this case. To see this in Equation (2.3), we substitute $Y(z) = 1 - p + pz$ into the logarithmic function in Equation (2.3), and observe that this gives the logarithmic function in Equation (2.8).

2.3.2 Generalizations of the FCTL queue

Oblakova et al. [146] have introduced several generalizations of the FCTL queue and established contour integrals for the first moment of the steady-state queue length for those models. We now show how contour-integral representations for these generalizations of the FCTL queue follow almost directly from the contour integral for the standard FCTL queue. We start from the definition of $X(z)$ in [146], a generalization of the function $X_g(z)$ that contains the FCTL queue and several extensions of the FCTL queue as special cases.

Definition 2.1 (Generalized FCTL queues, after [146]) *Consider the function $X(z)$ with $X(1) = 1$ and*

$$X(z) = \frac{\sum_{k=0}^{g-1} x_k z^k B(z)^{g-1-k}}{z^g - A(z)} \xi(z), \tag{2.9}$$

where $B(z)$ and $A(z)$ are PGFs and $\xi(z)$ is a function satisfying $\xi(1) = 0$, $\xi(z_l) \neq 0$ with $z_l \neq 1$ the roots of $z^g - A(z)$ inside the unit disk. Assume moreover that $B'(1) < 1$, $A'(1) < g$, that for some $\delta > 0$ the functions $A(z)$ and $B(z)$ are analytic within the disk $|z| < 1 + \delta$, and that $X(z)$ is analytic inside the unit disk and continuous up to the unit circle. Also assume that $t_0 > 1$, where $t_0 = \sup\{t \in \mathbb{R}_+ | B'(t)\, t - B(t) \leq 0\}$.

Here is the main result for the function $X(z)$:

Theorem 2.2 *Under Definition 2.1 there exists an $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0)$*

$$X(z) = \exp\left(\frac{1}{2\pi i}\oint_{|w|=1+\epsilon} \ln\left(\frac{zB(w) - wB(z)}{B(w) - w}\right) \frac{(w^g - A(w))'}{w^g - A(w)}\,\mathrm{d}w\right) \frac{1 - B'(1)}{z - B(z)} \frac{\xi(z)}{\xi'(1)}, \tag{2.10}$$

for all $|z| < 1 + \epsilon$, with principal value of the logarithm.

Proof. The proof will express Equation (2.9) as a product of the PGF of the standard FCTL queue and some analytic function. Denote the $g-1$ roots of $z^g - A(z)$ inside the unit circle by $z_1, \ldots, z_{g-1}$. We rewrite Equation (2.9), using $X(1) = 1$, as

$$X(z) = \frac{(g - A'(1))}{z^g - A(z)} \frac{\xi(z)}{\xi'(1)} \prod_{k=1}^{g-1} \frac{B(z)z_k - zB(z_k)}{B(z_k) - z_k}.$$

If we replace $Y(z)$ with $B(z)$ in $X_g(z)$, we see from Equation (2.5) that

$$X_g(z) = \frac{g - A'(1)}{z^g - A(z)} \frac{z - B(z)}{1 - B'(1)} \prod_{k=1}^{g-1} \frac{B(z)z_k - zB(z_k)}{B(z_k) - z_k}.$$

Using this, we see that we can express $X(z)$ in $X_g(z)$:

$$X(z) = X_g(z) \frac{1 - B'(1)}{z - B(z)} \frac{\xi(z)}{\xi'(1)}.$$

This gives the result. □

Let us now discuss some of the extensions contained in $X(z)$.

(i) The first extension concerns right-turning traffic. In this setting, the difference in discharge rate between delayed and non-delayed vehicles almost vanishes, i.e. the speed difference of delayed and non-delayed vehicles is almost negligible. This requires us to modify the FCTL assumption in order to put an upper bound on the number of vehicles that pass the traffic light without delay. This upper bound is set to one, whereas the FCTL assumption assumes this upper bound to be infinite. In this adjusted setting, we have that at most one vehicle can depart per green slot. Following [146], it can be shown that this model for right-turning traffic follows by setting $B(z) = Y(z)$, $A(z) = Y(z)^c$, and $\xi(z) = (z-1)Y(0)$, where $Y(z)$ is the PGF of the number of arrivals per slot. The contour-integral expression for the PGF thus follows from Theorem 2.2.

(ii) Another extension of the classical FCTL queue is one that accounts for disruptions of the traffic flow by e.g. pedestrians. To account for such disruptions, one could extend the red period for the traffic light of vehicles or shorten

the green period [146] according to some probability distribution. This extension thus requires an FCTL queue with random (but finite) green and red times, for which we choose $g = G$ with G denoting the maximum green time. Setting $B(z) = Y(z)$, $A(z) = \sum_{r,g} p_{r,g} Y(z)^{r+g} z^{G-g}$ with $p_{r,g}$ the probability that a cycle consists of g green and r red slots, and $\xi(z) = z - Y(z)$, then shows that this extension of interrupted flows is contained in Theorem 2.2.

(iii) The third extension we mention, relates to uncertainty in departure times of vehicles. Usually, we assume that in each slot corresponding to a green traffic light, a delayed vehicle might depart. In the case of distracted drivers we assume that a driver does not depart in such a slot with some probability p [146]. In the next slot, again, the driver does not depart with probability p. This results in drivers requiring a geometrically distributed number of slots before leaving the queue. We thus get $B(z) = Y(z)(1-p+pz)$, $A(z) = Y(z)^c(1-p+pz)^g$, and $\xi(z) = z - Y(z)(1-p+pz)$. Theorem 2.2 thus can be used to obtain a performance analysis for this model.

(iv) A fourth extension deals with relaxing the independence assumption of the arrival process during the red slots [146]. In this extension, the arrivals during a red time within a cycle may be dependent as opposed to the i.i.d. assumption on the Y (however, the arrivals during green slots still need to be i.i.d.). For this FCTL queue we should choose $B(z) = Y(z)$, $A(z) = A_r(z)Y(z)^g$, where $A_r(z)$ denotes the PGF of the arrival process during the whole red period, and $\xi(z) = z - Y(z)$.

In comparison with [146], we give an expression for the PGFs in terms of contour integrals. The results in [146] can be recovered by evaluating the derivative at $z = 1$ of our expression for the appropriate PGF. For insights into the differences between the various FCTL queue extensions we refer to the elaborate numerical study in [146].

2.4 Algorithmic methods

We now discuss the computational challenges that come with calculating the steady-state queue-length distribution, using either the contour integrals in Theorem 2.1 or the standard expression in terms of roots. The algorithms using contour integrals in this section are based on the representation in Equation (2.3) (but one could also take Equation (2.2)). Notice that we only need to expand $X_g(w)$ at $w = 0$ and $w = 1$, so inside the validity range of Equation (2.3).

2.4.1 From PGF to performance measures

The mean stationary overflow queue $\mathbb{E}[X_g]$ is given by $X_g'(1)$ and takes the form

$$\mathbb{E}[X_g] = \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} \frac{Y(z) - zY'(1)}{Y(z) - z} \frac{(z^g - A(z))'}{z^g - A(z)} \, \mathrm{d}z. \tag{2.11}$$

This result was recently obtained in [146] using a direct proof that converted the classical expression for $\mathbb{E}[X_g]$ in terms of complex-valued roots into the integral expression as in Equation (2.11).

From the PGF $X_g(z)$ we can in principle determine all stationary moments. Define

$$\begin{aligned} f(w) &:= \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} g(w,z) \frac{(z^g - A(z))'}{z^g - A(z)} \, \mathrm{d}z, \\ g(w,z) &:= \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z} \right), \\ h_k(w) &:= \begin{cases} 1, & k = 0, \\ h_{k-1}(w) f'(w) + h_{k-1}'(w), & k = 1, 2, \ldots. \end{cases} \end{aligned}$$

The moments $\mathbb{E}[X_g^k]$ then follow from symbolically differentiating the PGF as in Equation (2.3), and these derivatives can be expressed as

$$X_g^{(k)}(w) := \frac{\mathrm{d}^k}{\mathrm{d}w^k} X_g(w) = \frac{\mathrm{d}^k}{\mathrm{d}w^k} \exp\big(f(w)\big) = h_k(w) \exp\big(f(w)\big),$$

for $k = 0, 1, 2, \ldots$. Using this recursive expression, $X_g^{(k)}(w)$ can be expressed in terms of $f(w)$ and the first k derivatives of $f(w)$, denoted by $f^{(1)}(w), \ldots, f^{(k)}(w)$ with

$$\begin{aligned} f^{(j)}(w) &:= \frac{\partial^j}{\partial w^j} \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} g(w,z) \frac{(z^g - A(z))'}{z^g - A(z)} \, \mathrm{d}z \\ &= \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} g^{(j)}(w,z) \frac{(z^g - A(z))'}{z^g - A(z)} \, \mathrm{d}z \end{aligned}$$

and $g^{(j)}(w,z) := \frac{\partial^j}{\partial w^j} g(w,z)$, for $j = 1, 2, \ldots, k$. After substituting $w = 1$, we can express the first k moments of X_g in terms of k contour integrals that only involve the model primitives and the first k moments of Y. Using $f(1) = 0$, the variance of X_g given by $\mathrm{Var}(X_g) = h_2(1) + h_1(1) - (h_1(1))^2$ takes the form

$$\mathrm{Var}(X_g) = \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} \frac{z^2 \mathrm{Var}(Y) - zY(z)(1 + \mathbb{E}[Y^2] - 2\mathbb{E}[Y])}{(z - Y(z))^2} \frac{(z^g - A(z))'}{z^g - A(z)} \, \mathrm{d}z.$$

To determine the stationary distribution of the overflow queue we use that

$$\mathbb{P}(X_g = k) = \frac{1}{k!}\frac{\mathrm{d}^k}{\mathrm{d}w^k} X_g(w)\Big|_{w=0} = \frac{1}{k!} h_k(0) \exp\big(f(0)\big).$$

We observe that

$$\mathbb{P}(X_g = 0) = \exp\big(f(0)\big) = \exp\left(\frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \ln\left(\frac{z\mathbb{P}(Y=0)}{z - Y(z)}\right)\frac{(z^g - A(z))'}{z^g - A(z)}\,\mathrm{d}z\right).$$

Expressions for the other probabilities $\mathbb{P}(X_g = k)$ follow in a similar way as for $\mathbb{E}[X_g^k]$, but require evaluating the resulting function at $w = 0$ instead of $w = 1$ and dividing by $k!$. $\mathbb{P}(X_g = k)$ can thus be expressed in terms of $f(0), f^{(1)}(0), \ldots, f^{(k)}(0)$, again an expression that involves explicit contour integrals only.

2.4.2 Roots or integrals?

Compared with root finding, contour integrals have advantages and disadvantages. On the one hand, avoiding the implicitly defined roots is nice, because the integrals are explicit expressions in terms of the model primitives g, r, and $Y(z)$. On the other hand, the number of terms required to evaluate $f^{(j)}(w)$ grows exponentially in j. For, e.g., tail probabilities this symbolic differentiation becomes computationally cumbersome.

While in the early queueing literature root finding was considered to be prohibitively difficult, with the computational methods available nowadays it is possible to find the complex-valued roots of $z^g - A(z)$ with great accuracy. In Appendix 2.A we present the root-finding algorithm that we use in this chapter, which after extensive testing was found to be accurate and reliable for all considered choices of $A(z)$. The idea behind the algorithm is to approximate $A(z)$ with its Taylor series of order n, $A_n(z)$, reducing the problem to finding roots of polynomial equations. We also present some results that show that the roots of the n-th system converge to the roots of $z^g - A(z)$ as $n \to \infty$, and provide an explicit characterization of the roots for the case when $A(z)$ is the PGF of a Poisson random variable. In that case, the roots can be written in terms of the Lambert W-function, see Appendix 2.B.

Extensive tests with both algorithms did not result in any numerical issues, except for two obvious limitations: for tail probabilities the symbolic differentiation within the integrand becomes a bottleneck and for root finding loss of accuracy is expected when the number of roots g becomes excessively large (although a thousand roots present no difficulties). It seems that for most practical purposes both methods lead to reliable and accurate algorithms.

In terms of computation time, contour integration generally seems to be slower than root finding. For moments there is little difference, because both methods lead to explicit expressions. Our experiments have indicated, however, that the root-based PGF seems to be more suitable for determining the queue-length probabilities, because the roots have to be determined only once, whereas the contour integral approach requires the evaluation of another integral for each probability.

To illustrate the algorithms we now show some results for the FCTL queue with $g = 20$ and $c = 50$ in Figure 2.2. We consider Poisson arrivals with on average μ vehicles arriving per slot and four scenarios: $\mu = 0.2$ (light traffic), $\mu = 0.3$ (moderate traffic), $\mu = 0.36$ (heavy traffic), and $\mu = 0.38$ (extreme traffic). These arrival rates correspond to a vehicle-to-capacity ratio $\rho = \mu c/g$ ranging from 0.5 to 0.95. The results are calculated with both the roots-based and contour-integral expression, and are indistinguishable on the scale of the displayed figures.

Figure 2.2(a) shows the mean queue lengths $\mathbb{E}[X_1],\ldots,\mathbb{E}[X_c]$ through one cycle. Observe the strong cyclic behavior and the high sensitivity for ρ. Figure 2.2(b) shows the queue-length distribution at the start of the cycle, the moment that the traffic signal turns green and queue lengths are expected to peak. Observe the difference between operating at 75% or 95% of maximal capacity: the probability that more than 20 vehicles are waiting is only 0.002 for $\mu = 0.3$ and 0.32 for $\mu = 0.38$. Figure 2.2(c) depicts the distribution of the effective green time G, defined in [198] as the number of slots used for departure of delayed vehicles that arrive throughout the whole cycle. We have

$$\mathbb{P}(G = k) = \begin{cases} q_0 & \text{for } k = 0, \\ q_k - q_{k-1} & \text{for } k = 1,\ldots,g-1, \\ 1 - q_{g-1} & \text{for } k = g. \end{cases}$$

Since only one delayed vehicle departs per slot, this can also be considered to be the distribution of the platoon length consisting of delayed vehicles departing during one cycle. Observe that $\mathbb{P}(G = g)$ is practically zero when $\rho = 0.5$, but as high as 0.71 when $\mu = 0.38$, which means that only in 29% of the cycles the green time is long enough to let the queue vanish before the end of the green period.

Finally, we consider the delay distribution of an arbitrary vehicle arriving in the 10-th slot, which is during the green period. The stationary delay of a vehicle arriving in slot k, denoted by $D_{[k]}$, is defined as the number of slots between arrival and departure, not including the slot of arrival. Figure 2.2(d) shows $D_{[10]}$, which can be computed directly from X_9, i.e. the number of vehicles

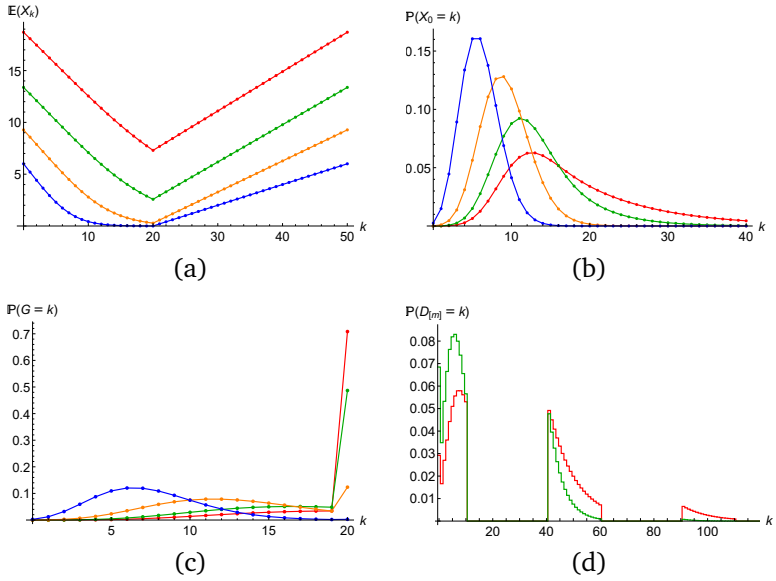


Figure 2.2: Several performance measures for the FCTL queue in Subsection 2.4.2 with $g = 20$, $c = 50$, and Poisson arrivals. The colors blue, orange, green, and red correspond to vehicle-to-capacity ratios of, respectively, 0.5, 0.75, 0.9, and 0.95. The subfigures show (a) the mean queue lengths during a cycle, (b) the queue-length distribution at the start of green periods, (c) the distribution of the effective green periods, and (d) the delay distribution of vehicles arriving in slot 10 (for $\rho = 0.9, 0.95$ only).

waiting at the start of the 10-th slot. If $X_9 = 0$ we have that $D_{[10]} = 0$; otherwise the delay can be expressed as a function of the number of vehicles present at the arrival of the tagged vehicle. This function (studied in detail in [206]) should take into account interruptions due to red periods, which explains the fragmented histograms in Figure 2.2(d).

2.5 Proof of the Pollaczek contour-integral representation

The proof of Theorem 2.1 contains several challenging steps and requires among others a proof that the function $Y(z)/z$ is injective in a region that contains the

unit disk, and a way to account for the branch cut caused by the logarithm in Equation (2.7) being taken over negative values. As explained briefly in Section 2.3, the proof of Theorem 2.1 exploits the factorized form as in Equation (2.5) and investigates in detail the logarithmic function in Equation (2.7). We present some useful properties of the function $Y(z)/z$, visible in both Equations (2.5) and (2.7). We then proceed to use Cauchy's theorem to obtain the contour-integral representation in Equation (2.3) for the case that $1 < w < 1+\epsilon$, and finally manipulate Equation (2.3) to obtain Equation (2.2) on the full range $|w| < 1+\epsilon$.

2.5.1 Auxiliary results

Before we prove Theorem 2.1, we present some auxiliary results for the function $Y(z)/z$. In [146, Theorem 1] it was shown that the function $Y(z)/z$ is injective on the disk $|z| \le 1$, so that all $Y(z_k)/z_k \neq Y(z_l)/z_l$ when $z_k \neq z_l$. For our proof we also need injectivity, but then for the larger disk with radius $t_0 > 1$. More specifically, let

$$t_0 := \sup\{t \in (0,R) \mid Y'(t)\,t - Y(t) \le 0\},$$

where R is the maximum value such that $Y(z)$ is analytic in the region $|z| < R$.

Lemma 2.3 *The function $Y(t)/t$ is strictly decreasing in $t \in (0, t_0]$.*

Proof. We have that

$$\frac{Y(t)}{t} = \frac{y_0}{t} + y_1 + y_2 t + \ldots, \qquad 0 < t < R,$$

is strictly convex since $y_0 > 0$, with derivative

$$\left(\frac{Y(t)}{t}\right)' = \frac{Y'(t)\,t - Y(t)}{t^2}, \qquad 0 < t < R. \tag{2.12}$$

Since $Y'(1) < Y(1) = 1$, we have that $t_0 > 1$. Now consider the following cases: (i) $y_k = 0$ for $k = 2,3,\ldots$ and (ii) there is a $k = 2,3,\ldots$ such that $y_k \neq 0$. For case (i), $Y(t)/t = y_0 t^{-1} + y_1$ is strictly decreasing in $t > 0$ since $y_0 > 0$. For case (ii), $y_k > 0$ for some $k \ge 2$, and so

$$Y'(t)\,t - Y(t) = -y_0 + \sum_{k=2}^{\infty} (k-1)\,y_k\,t^k \tag{2.13}$$

is strictly increasing in $t \in (0, R)$. From the definition of t_0, we then get that

$$Y'(t)\,t - Y(t) < 0, \quad t \in (0, t_0), \tag{2.14}$$

and so $Y(t)/t$ is strictly decreasing in $t \in (0, t_0)$ by Equation (2.12). □

Lemma 2.4 *The function $Y(z)/z$ is injective on the open disk $|z| < t_0$, so that for $|z| < t_0$, $|w| < t_0$:*

$$\frac{Y(z)}{z} = \frac{Y(w)}{w} \Rightarrow z = w. \tag{2.15}$$

Proof. In case (i), $y_k = 0$ for $k = 2, 3, \ldots$, we have $Y(z)/z = y_0 z^{-1} + y_1$ and the result is trivial since $y_0 > 0$. For case (ii), there is a $k = 2, 3, \ldots$ such that $y_k \neq 0$, we let $|z| < t_0$, $|w| < t_0$. Then

$$\begin{aligned}\left|\frac{Y(z)}{z} - \frac{Y(w)}{w}\right| &= \left|y_0 \frac{w - z}{zw} + \sum_{k=2}^{\infty} y_k (z^{k-1} - w^{k-1})\right| \\ &= |z - w| \left| - \frac{y_0}{zw} + \sum_{k=2}^{\infty} y_k \frac{z^{k-1} - w^{k-1}}{z - w}\right|.\end{aligned}$$

Let $t := \max\{|z|, |w|\} < t_0$. Then $|y_0/(zw)| \geq y_0/t^2$ while

$$\left|\frac{z^{k-1} - w^{k-1}}{z - w}\right| = \left|z^{k-2} + wz^{k-3} + \cdots + zw^{k-3} + w^{k-2}\right| \leq (k-1)\,t^{k-2}.$$

Therefore, when $z \neq w$,

$$\left|\frac{Y(z)}{z} - \frac{Y(w)}{w}\right| \geq |z - w| \Big(\frac{y_0}{t^2} - \sum_{k=2}^{\infty} (k-1) y_k t^{k-2} \Big) > 0$$

by Equation (2.13) and Equation (2.14). This proves Equation (2.15). □

Lemma 2.5 *Let $\epsilon > 0$ be such that $1 + \epsilon < t_0$, and take $w \in (1, 1 + \epsilon)$. For $|z| < t_0$,*

$$\frac{wY(z) - zY(w)}{Y(z) - z} \in (-\infty, 0] \Leftrightarrow 1 \leq z \leq w. \tag{2.16}$$

Furthermore

$$-1 < z < 1 \Rightarrow \frac{wY(z) - zY(w)}{Y(z) - z} > 0. \tag{2.17}$$

Proof. For $a \le 0$,

$$\frac{wY(z) - zY(w)}{Y(z) - z} = a \Leftrightarrow \frac{Y(z)}{z} = \frac{Y(w) - a}{w - a}.$$

Since $1 < Y(w) < w$, the function $(Y(w) - a)/(w - a)$ increases from $Y(w)/w$ at $a = 0$ to 1 at $a = -\infty$ when a decreases from 0 to $-\infty$. Since $Y(v)/v$ decreases strictly in $v \in [1, w]$, there is for any $a \le 0$ a unique $v = v(a) \in [1, w]$ such that

$$\frac{Y(v)}{v} = \frac{Y(w) - a}{w - a}.$$

Since by Lemma 2.4 $Y(z)/z$ is injective in $|z| < t_0$, we get Equation (2.16).

We next show Equation (2.17). Obviously, Equation (2.17) holds for $z = 0$. For $z \neq 0$, we have

$$\frac{wY(z) - zY(w)}{Y(z) - z} = w \frac{Y(z)/z - Y(w)/w}{Y(z)/z - 1}.$$

By Lemma 2.3, we have

$$\frac{Y(z)}{z} - \frac{Y(w)}{w} > \frac{Y(z)}{z} - 1 > 0,$$

if $0 < z < 1 < w < t_0$, and so Equation (2.17) holds for $z \in (0, 1)$. Next, by Lemma 2.4, we have $Y(z)/z \neq Y(w)/w$ when $z \in (-1, 0)$ and $1 < w < t_0$. Also, $Y(z)/z \to -\infty$ when $z \uparrow 0$. Because of continuity of $Y(z)/z$ in $z \in (-1, 0)$ and because $Y(z)/z$ is real, we thus have that

$$\frac{Y(z)}{z} < \frac{Y(w)}{w} < 1, \qquad z \in (-1, 0),$$

and so Equation (2.17) also holds for $z \in (-1, 0)$. □

As a consequence of Lemma 2.5, taking the principal value logarithm in Equation (2.7) when $1 < w < 1 + \epsilon < t_0$, we obtain a function of z that is analytic in the open disk $|z| < t_0$, with a branch cut along $[1, w]$. Indeed, Lemma 2.5 tells us that, with $w > 1$, the only negative values for $(wY(z) - zY(w)) / (Y(z) - z)$ in the entire complex circle with radius at most t_0 are attained for $1 \le z \le w$. We thus might take the principal value logarithm of $(wY(z) - zY(w)) / (Y(z) - z)$ and this implies that we need to take care of the branch cut along $1 \le z \le w$. By means of this logarithm, we create the appropriate residues at the roots of the function $z^g - A(z)$, which is the function that we consider in the next part of the proof of Theorem 2.1.

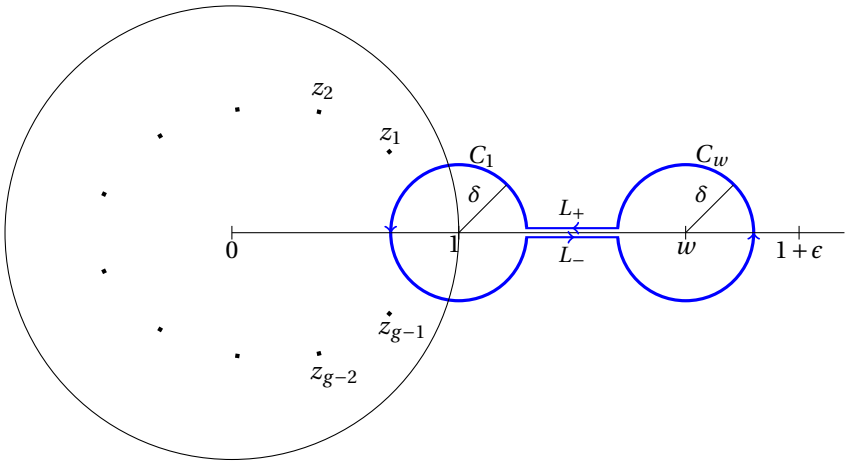


Figure 2.3: The four components, C_1, C_w, L_+, and L_-, of contour C.

2.5.2 Contour integral for (2.3)

We next consider the function $z^g - A(z)$ that has its zeros in $|z| \le 1$ at $z = z_0 = 1, z_1, \ldots, z_{g-1}$, while its other zeros have modulus greater than one. Let R_0 be the zero outside $|z| \le 1$ of smallest modulus; we have that R_0 is real and larger than one. Take $\epsilon > 0$ such that $1+\epsilon < \min\{t_0, R_0\}$ and consider the integral

$$I(w) = \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right) \frac{(z^g - A(z))'}{z^g - A(z)} \, \mathrm{d}z. \tag{2.18}$$

Choose $\delta > 0$ such that $\delta < \frac{1}{2}(w-1)$ and $\delta < 1 + \epsilon - w$ while $|z_k - 1| > \delta, k = 1, \ldots, g-1$. Now let C be the positively oriented contour consisting of the circles $C_1(\delta)$ and $C_w(\delta)$ of radii δ around 1 and w, respectively, together with the line segments $L_\pm(\delta) = \{z = t \pm i0 \mid 1+\delta \le t \le w-\delta\}$, where $\pm i0 := \lim_{c \downarrow 0} \pm ci$ and $i^2 = -1$. See Figure 2.3 for the positioning of the contour C with its four components in the disk $|z| < 1 + \epsilon$ and relative to the zeros of $z^g - A(z)$. Then, by Cauchy's theorem,

$$I(w) = \sum_{k=1}^{g-1} \ln\left(\frac{wY(z_k) - z_k Y(w)}{Y(z_k) - z_k}\right) + \frac{1}{2\pi i} \oint_C \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right) \frac{(z^g - A(z))'}{z^g - A(z)} \mathrm{d}z. \tag{2.19}$$

On the line segments $z = t \pm i0$, $1+\delta \le t \le w-\delta$, we use that

$$wY(t) - tY(w) > 0 > Y(t) - t.$$

With the principal value choice for ln, we then get for $1+\delta \le t \le w-\delta$:

$$\ln\left(\frac{wY(t\pm i0) - (t\pm i0)Y(w)}{Y(t\pm i0) - (t\pm i0)}\right) = \ln\left(\frac{wY(t) - tY(w)}{t - Y(t)}\right) \pm \pi i. \tag{2.20}$$

Therefore, also using that $t^g - A(t) > 0$ and $1 < t < 1+\epsilon$, we have

$$\begin{aligned}
&\frac{1}{2\pi i}\oint_C \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\frac{(z^g - A(z))'}{z^g - A(z)}\,\mathrm{d}z \\
&= \frac{1}{2\pi i}\int_{1+\delta}^{w-\delta}\left\{\left[-\left(\ln\left(\frac{wY(t) - tY(w)}{t - Y(t)}\right) + \pi i\right) + \left(\ln\left(\frac{wY(t) - tY(w)}{t - Y(t)}\right) - \pi i\right)\right]\right. \\
&\quad \left.\frac{(t^g - A(t))'}{t^g - A(t)}\,\mathrm{d}t\right\} + \frac{1}{2\pi i}\oint_{C_1(\delta)} + \frac{1}{2\pi i}\oint_{C_w(\delta)} \\
&= -\int_{1+\delta}^{w-\delta}\frac{(t^g - A(t))'}{t^g - A(t)}\,\mathrm{d}t + \frac{1}{2\pi i}\oint_{C_1(\delta)} + \frac{1}{2\pi i}\oint_{C_w(\delta)},
\end{aligned} \tag{2.21}$$

where

$$\frac{1}{2\pi i}\oint_{C_x(\delta)} = \frac{1}{2\pi i}\oint_{C_x(\delta)} \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\frac{(z^g - A(z))'}{z^g - A(z)}\,\mathrm{d}z,$$

with $x = 1, w$. Now, since $g - A'(1) > 0$ (due to stability), we have that

$$\begin{aligned}
&\int_{1+\delta}^{w-\delta}\frac{(t^g - A(t))'}{t^g - A(t)}\,\mathrm{d}t = \ln\left(t^g - A(t)\right)\Big|_{1+\delta}^{w-\delta} \\
&= \ln\left(w^g - A(w)\right) + O(\delta) - \ln\left[(g - A'(1))\delta + O(\delta^2)\right] \\
&= \ln\left(\frac{w^g - A(w)}{g - A'(1)}\right) - \ln\delta + O(\delta),
\end{aligned} \tag{2.22}$$

where we have used that

$$t^g - A(t) = 0 + (t^g - A(t))'_{t=1}(t-1) + O\big((t-1)^2\big), \quad t \to 1.$$

As to the last integral on the last line of Equation (2.21), we use that

$$wY(z) - zY(w) = (wY'(w) - Y(w))(z-w) + O(|z-w|^2),$$

$$Y(z) - z = Y(w) - w + O(|z-w|),$$
$$z^g - A(z) = w^g - A(w) + O(|z-w|),$$

with non-vanishing numbers $wY'(w) - Y(w)$, $Y(w) - w$, and $w^g - A(w)$. Therefore, we have

$$\frac{1}{2\pi i}\oint_{C_w(\delta)} = O(\delta \ln \delta), \quad \delta \downarrow 0. \tag{2.23}$$

The middle integral on the last line of Equation (2.21) is more delicate since both $Y(z) - z$ and $z^g - A(z)$ vanish at $z = 1$. For $z = 1 + \delta e^{i\phi}$ with $0 < \phi < 2\pi$ and $\delta \downarrow 0$, we get

$$\begin{aligned}\frac{wY(z) - zY(w)}{Y(z) - z} &= \frac{w - Y(w) + O(|z-1|)}{1 + Y'(1)(z-1) - z + O(|z-1|^2)} \\ &= -\frac{w - Y(w) + O(|z-1|)}{(1 - Y'(1))(z-1) + O(|z-1|^2)} \\ &= -\frac{w - Y(w)}{1 - Y'(1)}\,\frac{1}{\delta}e^{-i\phi}\big(1 + O(\delta)\big).\end{aligned}$$

Hence, since $w - Y(w) > 0$, $1 - Y'(1) > 0$, we obtain

$$\begin{aligned}\ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right) &= \ln\left|\frac{wY(z) - zY(w)}{Y(z) - z}\right| + i\arg\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right) \\ &= \ln\left(\frac{w - Y(w)}{1 - Y'(1)}\right) - \ln\delta + i(\pi - \phi) + O(\delta).\end{aligned} \tag{2.24}$$

Next, as $z \to 1$, we have

$$\frac{(z^g - A(z))'}{z^g - A(z)} = \frac{g - A'(1) + O(|z-1|)}{(g - A'(1))(z-1) + O(|z-1|^2)} = \frac{1}{z-1} + O(1), \tag{2.25}$$

since $g - A'(1) > 0$. Hence, from Equations (2.24) and (2.25), with $z = 1 + \delta e^{i\phi}$ and $\mathrm{d}z = i\delta e^{i\phi}\mathrm{d}\phi$ in the integral over C_1, we get

$$\begin{aligned}\frac{1}{2\pi i}\oint_{C_1(\delta)} &= \frac{1}{2\pi i}\int_0^{2\pi} \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\frac{(z^g - A(z))'}{z^g - A(z)}\, i\delta e^{i\phi}\mathrm{d}\phi \\ &= \frac{1}{2\pi i}\int_0^{2\pi} \left[\ln\left(\frac{w - Y(w)}{1 - Y'(1)}\right) - \ln\delta + i(\pi - \phi) + O(\delta)\right]\cdot \\ &\quad \left[\frac{1}{\delta}e^{-i\phi} + O(1)\right] i\delta e^{i\phi}\mathrm{d}\phi\end{aligned}$$

$$= \ln\left(\frac{w - Y(w)}{1 - Y'(1)}\right) - \ln\delta + O(\delta), \tag{2.26}$$

where we have also used that $\int_0^{2\pi} (\pi - \phi)\, \mathrm{d}\phi = 0$.

Using Equations (2.22), (2.23), and (2.26) in Equation (2.21) yields

$$\begin{aligned}
&\frac{1}{2\pi i} \oint_C \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right) \frac{(z^g - A(z))'}{z^g - A(z)}\, \mathrm{d}z \\
&= \ln\left(\frac{g - A'(1)}{w^g - A(w)}\right) + \ln\delta + O(\delta) + O(\delta \ln\delta) + \ln\left(\frac{w - Y(w)}{1 - Y'(1)}\right) - \ln\delta + O(\delta) \\
&= \ln\left(\frac{g - A'(1)}{w^g - A(w)} \cdot \frac{w - Y(w)}{1 - Y'(1)}\right) + O(\delta).
\end{aligned}$$

Returning then to Equations (2.18)-(2.19), letting $\delta \downarrow 0$, we see that

$$I(w) = \ln\left[\frac{g - A'(1)}{w^g - A(w)} \cdot \frac{w - Y(w)}{1 - Y'(1)} \prod_{k=1}^{g-1} \frac{wY(z_k) - z_k Y(w)}{Y(z_k) - z_k}\right] = \ln\left[X_g(w)\right] \tag{2.27}$$

by Equation (2.5). Here we have also used that the zeros z_k are real or come in conjugate pairs (as is noted in Subsection 2.3.1) so that for $w \in (1, 1+\epsilon)$ both $X_g(w)$ and the product $\prod_{k=1}^{g-1}$ in Equation (2.27) are real and positive, with

$$\ln\left(\prod_{k=1}^{g-1} \frac{wY(z_k) - z_k Y(w)}{Y(z_k) - z_k}\right) = \sum_{k=1}^{g-1} \ln\left(\frac{wY(z_k) - z_k Y(w)}{Y(z_k) - z_k}\right).$$

This proves Equation (2.3) for $w \in (1, 1+\epsilon)$.

2.5.3 Completion of the proof

The extension of the validity range of Equation (2.3) beyond the set $1 < w < 1+\epsilon$ is compromised by the appearance of the factor $\ln[(wY(z) - zY(w))/(Y(z) - z)]$ in the integrand. The validity range can be extended to an open set containing the interval $[0, 1]$, allowing computation of moments and derivatives. To see this, let

$$Q(z, w) = \frac{wY(z) - zY(w)}{Y(z) - z} = Y(w) \frac{1 - \frac{Y(z)/z}{Y(w)/w}}{1 - Y(z)/z}, \qquad |z|, |w| \le 1 + \epsilon.$$

For $0 \le w \le 1$ and $|z| = 1 + \epsilon$, we have

$$0 < Y(0) \le Y(w) \le 1,$$

$$\left|\frac{Y(z)/z}{Y(w)/w}\right| \le \left|\frac{Y(z)}{z}\right| \le \frac{Y(1+\epsilon)}{1+\epsilon} < 1,$$

and so $Q(z,w)$ is bounded away from $(-\infty,0]$ when $0 \le w \le 1$ and $|z| = 1+\epsilon$. By continuity of Q as a function of w, this continues to hold for w in an open set Ω containing $[0,1]$ and $|z| = 1+\epsilon$. This implies that $\ln Q(z,w)$ is analytic in $w \in \Omega$, with the principal value ln, extending the validity of Equation (2.3) to $w \in \Omega$ by analyticity. We have extensive numerical evidence that the set of w for which $Q(z,w) \notin (-\infty,0]$, all z with $|z| = 1+\epsilon$, contains a disk around 0 with radius not significantly smaller than $1+\epsilon$. This would extend the validity of Equation (2.3) beyond the unit disk $|w| \le 1$.

We now re-express the integral form in Equation (2.3) to a form that is valid for all w, $|w| < 1+\epsilon$. We choose ϵ here such that $1+\epsilon < \min\{t_0, R_0\} =: 1+\epsilon_0$ as in Subsection 2.5.2. Let w be fixed with $1 < w < 1+\epsilon$. We compute, for $|z| = 1+\epsilon$,

$$\frac{(z^g - A(z))'}{z^g - A(z)} = \frac{g}{z} + \frac{\left(1 - \frac{A(z)}{z^g}\right)'}{1 - \frac{A(z)}{z^g}} = \frac{g}{z} + \frac{\mathrm{d}}{\mathrm{d}z}\left[\ln\left(1 - \frac{A(z)}{z^g}\right)\right],$$

where we can choose the principal value of ln since

$$\left|\frac{A(z)}{z^g}\right| \le \frac{A(1+\epsilon)}{(1+\epsilon)^g} < 1, \quad |z| = 1+\epsilon.$$

As in Equations (2.20)-(2.21), we have

$$\begin{aligned}
&\frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\frac{g}{z}\mathrm{d}z \\
&\qquad = g\ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\Big|_{z=0} + \frac{g}{2\pi i}\oint_C \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\frac{1}{z}\mathrm{d}z \\
&\qquad = g\ln w - g\int_{1+\delta}^{w-\delta}\frac{1}{z}\mathrm{d}z + O(\delta\ln\delta) \\
&\qquad = g\ln w - g\ln\left(\frac{w-\delta}{1+\delta}\right) + O(\delta\ln\delta),
\end{aligned}$$

and this vanishes as $\delta \downarrow 0$. Therefore, see Equation (2.18), we have

$$\begin{aligned}
I(w) &= \frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\frac{\mathrm{d}}{\mathrm{d}z}\left[\ln\left(1 - \frac{A(z)}{z^g}\right)\right]\mathrm{d}z \\
&= \frac{-1}{2\pi i}\oint_{|z|=1+\epsilon} \frac{\mathrm{d}}{\mathrm{d}z}\left[\ln\left(\frac{wY(z) - zY(w)}{Y(z) - z}\right)\right]\ln\left(1 - \frac{A(z)}{z^g}\right)\mathrm{d}z,
\end{aligned}$$

where we have used partial integration with the continuous differentiable functions $\ln\left(1-A(z)/z^g\right)$ and $\ln\left[(wY(z)-zY(w))/(Y(z)-z)\right]$ on the closed contour $|z|=1+\epsilon$. We compute

$$\frac{\mathrm{d}}{\mathrm{d}z}\left[\ln\left(\frac{wY(z)-zY(w)}{Y(z)-z}\right)\right] = \frac{Y'(z)z-Y(z)}{Y(z)-z}\,\frac{Y(w)-w}{wY(z)-zY(w)},$$

and obtain

$$I(w) = \frac{-1}{2\pi i}\oint_{|z|=1+\epsilon} \frac{Y'(z)z-Y(z)}{Y(z)-z}\,\frac{Y(w)-w}{wY(z)-zY(w)}\ln\left(1-\frac{A(z)}{z^g}\right)\mathrm{d}z, \qquad (2.28)$$

which is valid for any $w \in (1, 1+\epsilon)$.

We now extend Equation (2.28) to all w with $|w| < 1+\epsilon$ using Lemma 2.5. Let $0 < \epsilon_1 < \epsilon$. We have $|Y(z)-z| > 0$ when $|z| = 1+\epsilon$ and

$$\left|wY(z)-zY(w)\right| > 0,$$

when $|z| = 1+\epsilon$ and $|w| \le 1+\epsilon_1$ by Lemma 2.5 and $Y(0) \neq 0$. Therefore, by continuity and compactness, $(wY(z)-zY(w))(Y(z)-z)$ is bounded away from 0 when $|z| = 1+\epsilon$ and $|w| \le 1+\epsilon_1$. This implies that the right-hand side of Equation (2.28) is analytic in w, $|w| < 1+\epsilon_1$, by analyticity of Y. Since $X_g(w) = \exp(I(w))$ for $1 < w < 1+\epsilon$, we then get by analyticity of X_g that

$$X_g(w) = \exp\left(\frac{-1}{2\pi i}\oint_{|z|=1+\epsilon} \frac{Y'(z)z-Y(z)}{Y(z)-z}\,\frac{Y(w)-w}{wY(z)-zY(w)}\ln\left(1-\frac{A(z)}{z^g}\right)\mathrm{d}z\right) \qquad (2.29)$$

holds for all w, $|w| \le 1+\epsilon_1$ and any $\epsilon_1 \in (0,\epsilon)$. Then a simple rearrangement of the integrand in Equation (2.29) yields Theorem 2.1.

2.6 Conclusion

We have presented novel formal solutions for the FCTL queue in the form of contour integrals. Theorem 2.1 presents the contour-integral representation for the PGF of the overflow queue $X_g(z)$. From this PGF, essentially all relevant information about the stationary behavior of the FCTL can be obtained, by taking derivatives at $z = 1$ for the moments, derivatives at $z = 0$ for the distribution, and by using simple recursions to obtain the queue lengths at all time epochs within the cycle and the stationary delay distribution. A contour-integral expression for the first moment was recently obtained by Oblakova et al. [146] and the present

chapter can be seen as an extension of that work. Together, those results present an alternative approach for the FCTL queue and its generalizations, using contour integrals instead of factorizations in terms of complex roots that need to be determined numerically. In [146] generalizations of the FCTL assumption were considered, for which we have obtained similar Pollaczek-type contour integrals for the PGF.

In classical queueing theory, a prominent line of research is related to heavy traffic, an asymptotic regime in which the traffic intensity approaches 100%. Next to more probabilistic methods such as weak convergence techniques and coupling, another way to obtain heavy-traffic results is through the asymptotic evaluation of Pollaczek-type integrals, see e.g. [55,110] for single-server queues and [101] for classical bulk-service queues. Now that Pollaczek-type integrals for the FCTL queue are available, it is worthwhile to explore the possibilities for a heavy-traffic analysis, which is one of the topics studied in Chapter 3.

It is of interest to provide a proper comparison between the various methods that are available for studying the FCTL queue. We think e.g. of obtaining performance measures from a PGF based on the root-based expression and the contour-integral expression. It would be interesting to know in which cases one could better use the expression for the PGF which is based on roots and when the one based on the contour integral is more suitable.

Appendix

2.A Root-finding algorithm

We present a root-finding algorithm and some supporting results. A similar algorithm was used in [19]. The idea behind the algorithm is that roots of polynomial equations are generally easy to find, at least numerically. Therefore, we approximate $A(z)$ (which typically is a non-polynomial function) with its Taylor series $A_n(z)$ of order n. Solving this truncated equation boils down to root-finding of a polynomial. If the roots of the truncated equation are sufficiently close to the roots of $z^g - A(z)$, we can find the latter roots easily from the roots of $z^g - A_n(z)$ by using a Newton-Raphson type method.

Algorithm 2.1 Root-finding based on truncated Taylor series of $z^g - A(z)$.

1: Input: $A(z), g$ (and often $A(z) = Y(z)^c$).
2: Define: $D(z) = z^g - A(z)$.
3: Compute n: $\max\{100, 50 + \max\{c, g\}\}$.
4: Compute Taylor expansion $D_n(z)$ of $D(z)$ of order n.
5: Numerically solve $D_n(z) = 0$ for $|z| \leq 1$, obtaining roots $\hat{z}_1, \ldots, \hat{z}_g$.
6: Use $\hat{z}_1, \ldots, \hat{z}_g$ as input for a method to find the roots of $D(z)$ for $|z| \leq 1$, obtaining roots $z_1, \ldots, z_g$.
7: Return $z_1, \ldots, z_g$.

We now present two propositions in support of Algorithm 2.1. The first proposition states that under very mild conditions the number of roots of $z^g -$

$A(z)$ on or within the unit circle is equal to the number of roots of the truncated equation $z^g - A_n(z)$ on or within the unit circle. The second proposition shows that the roots of the truncated equation converge to the roots of $z^g - A(z)$ (when n tends to infinity).

Proposition 2.6 *Let $D(z) = z^g - A(z)$ and let $D_n(z) := z^g - A_n(z)$, where $A_n(z)$ denotes the n-th order Taylor approximation of $A(z)$. Upon assuming that $A(z)$ is a PGF; that $A(z)$ is analytic in the disk $|z| < 1+\delta$ for some $\delta > 0$; and that $g < A'(1)$, $D_n(z) = 0$ has as many roots on or within the unit circle as $D(z)$ (i.e. g).*

Proof. Rouché's theorem says that if f and g are analytic inside some region K with closed contour ∂K and if $|g(z)| < |f(z)|$ on ∂K, then f and $f+g$ have the same number of zeros inside K.

The conditions that $A(z)$ has to be analytic in $|z| < 1+\delta$ and $g < A'(1)$ together imply

$$(1+\gamma)^g > A(1+\gamma), \tag{2.30}$$

for some $\gamma \in (0, \delta)$, see e.g. [64]. Assume $|z| = 1+\gamma$. Then:

$$|z|^g = (1+\gamma)^g > A(1+\gamma) \geq A_n(1+\gamma) = A_n(|z|) \geq |A_n(z)|,$$

where the strict inequality follows from Equation (2.30) and the remaining inequalities from the fact that $A(z)$ is a PGF. So we may apply Rouché's theorem on $f(z) = z^g$ and $g(z) = -A_n(z)$. Since for any $\gamma > 0$, z^g has g roots within the circle $|z| = 1+\gamma$, we conclude that $D_n(z)$ has g roots as well, just as $D(z)$. □

Proposition 2.7 *Let $D(z)$ and $D_n(z)$ be as defined in Proposition 2.6. Let z_j, $j = 1, \ldots, g$, be the roots of $D(z)$ on or within the unit circle. Then*

$$|D_n(z_j)| \leq \sum_{j=n+1}^{\infty} a_k,$$

for $j = 1, \ldots, g$, where $a_k = \mathbb{P}(A = k)$.

Proof. We directly obtain from the definition of z_j that

$$\begin{aligned} |D_n(z_j)| &= |D_n(z_j) - D(z_j)| \\ &= \left| z_j^g - \sum_{i=0}^{n} a_i z_j^i - z_j^g + \sum_{i=0}^{\infty} a_i z_j^i \right| = \left| \sum_{i=n+1}^{\infty} a_i z_j^i \right| \leq \sum_{i=n+1}^{\infty} a_i, \end{aligned}$$

because $a_i \geq 0$ and $|z_j| \leq 1$. □

From Proposition 2.7 we see that if we let n tend to infinity, then $D_n(z_j)$ tends to 0. This implies that the roots obtained by using $D_n(z)$ will be close to the actual roots of $D(z)$ when n is sufficiently high.

2.B Poisson case

To close this chapter, we provide an explicit expression for the roots of $z^g - A(z)$ in case of Poisson arrivals, i.e. we take $A(z) = \mathrm{e}^{c\mu(z-1)}$. Let $W(\cdot)$ denote the principal value of the Lambert W-function, see e.g. [60]. Then

$$z_k = -\frac{g}{c\mu} W\left(-\frac{c\mu}{g}\mathrm{e}^{2\pi ik/g}\mathrm{e}^{-c\mu/g}\right), \quad k = 1,\ldots,g-1,$$

with i the imaginary unit satisfying $i^2 = -1$. It is then straightforward to show that the z_k are the solutions of $z^g - A(z) = 0$ within the unit circle as we have that

$$\begin{aligned}
A(z_k) &= \exp(c\mu(z_k - 1)) \\
&= \exp\left[c\mu \cdot \left\{-\frac{g}{c\mu} W\left(-\frac{c\mu}{g}\mathrm{e}^{2\pi ik/g}\mathrm{e}^{-c\mu/g}\right)\right\}\right] \mathrm{e}^{-c\mu} \\
&= \frac{W\left(-\frac{c\mu}{g}\mathrm{e}^{2\pi ik/g}\mathrm{e}^{-c\mu/g}\right)^g}{\left(-\frac{c\mu}{g}\mathrm{e}^{2\pi ik/g}\mathrm{e}^{-c\mu/g}\right)^g} \mathrm{e}^{-c\mu} \\
&= \left(-\frac{g}{c\mu}\right)^g W\left(-\frac{c\mu}{g}\mathrm{e}^{2\pi ik/g}\mathrm{e}^{-c\mu/g}\right)^g \mathrm{e}^{-2\pi ik}\mathrm{e}^{c\mu}\mathrm{e}^{-c\mu} \\
&= \left(-\frac{g}{c\mu}\right)^g W\left(-\frac{c\mu}{g}\mathrm{e}^{2\pi ik/g}\mathrm{e}^{-c\mu/g}\right)^g = z_k^g,
\end{aligned}$$

where we used some properties of the Lambert W-function and that $\mathrm{e}^{2\pi ik} = \mathrm{e}^{-2\pi ik} = 1$ for $k = 1,\ldots,g-1$.

Chapter 3

Optimal capacity allocation for heavy-traffic Fixed-Cycle Traffic-Light queues and intersections

3.1 Introduction

Optimizing traffic-light settings is particularly relevant when the vehicle-to-capacity ratio approaches the maximal sustainable level. To deal with such scenarios, we establish heavy-traffic limit theorems for the FCTL queue that provide accurate performance approximations for one queue in heavy traffic. We use these heavy-traffic approximations to approximatively solve optimization problems that aim for an optimal division of green times among multiple conflicting traffic streams. It turns out that the reduced complexity of the heavy-traffic approximations leads to tractable optimization problems and close-to-optimal signal prescriptions. Our optimization problems are reminiscent of the so-called capacity allocation problem, originally formulated by Kleinrock [111] for dividing capacity among multiple independent $M/M/1$ queues, with the aim of minimizing the average waiting time in all queues. This optimization problem has an elegant explicit solution and was later generalized by Wein [219] for Jackson networks with product-form solutions. Wein [219] solved the opti-

mization problem by relaxing the original problem through insertion of classical heavy-traffic approximations. We adopt a similar approach, but need to deal with the specific challenges that come with considering FCTL queues rather than standard queues.

The heavy-traffic scenario that we consider lets the cycle length grow large while at the same time the load or vehicle-to-capacity ratio approaches 100%. As far as we are aware, this is the first study that applies this scenario for the FCTL queue. Related scalings in continuous-time single-server queues are referred to as "nearly-deterministic regime" [177, 178] and in multi-server settings as the Halfin-Whitt regime or Quality-and-Efficiency-Driven (QED) regime [89, 208]. The term QED regime was coined because queueing systems in this regime can deal with high vehicle-to-capacity ratios while the probability of no delay stays strictly between 0 and 1. We show that similar favorable properties exist for the heavy-traffic FCTL queue.

To establish the FCTL heavy-traffic results, we use the transform expressions obtained in Chapter 2. In particular, we use the contour-integral representation for the PGF of the overflow queue, Equation (2.2) in Theorem 2.1. Establishing scaling limits requires showing convergence of transforms such as the PGF which proves to be quite challenging. The main idea of our proof is to expand the integrand of the contour integral and to show that in the heavy-traffic regime only the first few terms of the expansion (up to leading order) dominate the numerical value of the integral. Making such observations rigorous, however, requires careful analysis. While this analysis is new, in classical queueing theory, establishing heavy-traffic results through the asymptotic evaluation of contour integrals was done in e.g. [55, 110] for single-server queues and in [101] for classical bulk-service queues.

In the heavy-traffic regime that we consider, the scaled queue length turns out to converge to a reflected Gaussian random walk, a stochastic process that occurs in a range of other applications and that has been studied in great detail [15, 39, 99]. We exploit this connection to convert known results for the reflected Gaussian random walk into heavy-traffic approximations for the FCTL queue. These heavy-traffic approximations are considerably easier than the exact (contour-integral) expressions, which presents analytic advantages when considering the optimization problem of finding the optimal traffic-light settings for intersections with cyclic arrangements of multiple conflicting traffic flows. The heavy-traffic approximations let us obtain closed-form expressions for such optimal settings. A similar strategy to obtain optimal green times for vehicle-actuated traffic lights is formulated in [197, Chapter 6], where an approximation is found for the mean delay per lane, which is then used to approximate an

objective function. This optimization problem is then solved to optimality using the approximated objective function and Lagrange multiplier techniques, similar in spirit to what we do in Subsection 3.3. The optimal green-time allocation for vehicle-actuated traffic lights obtained in [197] also has a similar structure as the optimal allocations that we obtain in Section 3.3. Traffic lights with fixed settings are also studied in [197] where the problem is formulated and solved as a Mixed Integer Program. Instead of the need for such optimization schemes, we present one-line calculations for close-to-optimal green-time allocations.

Our main contributions can be summarized as follows:

(i) For the FCTL queue we obtain novel heavy-traffic limit theorems by asymptotic evaluation of contour integrals, showing that the scaled queue length converges to a reflected Gaussian random walk.

(ii) We leverage the limit theorems to obtain sharp performance approximations for one queue in heavy traffic, utilizing existing results for the reflected Gaussian random walk.

(iii) We consider optimization problems that find the optimal division of green times among multiple conflicting traffic streams and show that inserting heavy-traffic approximations leads to tractable optimization problems and close-to-optimal signal prescriptions.

Chapter outline

This chapter is organized as follows. In Section 3.2 we present the heavy-traffic analysis of the FCTL queue. Using the resulting heavy-traffic approximations, we present in Section 3.3 the close-to-optimal traffic-light settings for the situation of multiple conflicting traffic streams. Numerical examples are presented in Section 3.4. We present the main heavy-traffic proof in Section 3.5 and provide a conclusion in Section 3.6.

3.2 FCTL queue in heavy traffic

First, we briefly review the traditional analysis of the FCTL queue. Subsequently, we give the heavy-traffic scaling that we use in the remainder of the chapter and provide several results for the queue-length process under the introduced heavy-traffic scaling.

The queue-length process at the end of the green period for the FCTL queue gives rise to a Lindley-type recursion. We have that

$$X_{g,n+1} = \max\{0, X_{g,n} + A_n - g\}, \tag{3.1}$$

with A_n the number of *delayed* vehicles arriving in cycle n. Observe that (3.1) is not a standard Lindley recursion, due to the FCTL assumption and hence the intricate dependency between the delayed arrivals A_n and $X_{g,n}$.

We shall focus on the limiting queue length $X_g := \lim_{n\to\infty} X_{g,n}$, which is well defined if the system is stable. I.e. we require that $c\mathbb{E}[Y] = c\mu < g$, with Y the number of arrivals per slot and where we define μ to be the mean number of arrivals per slot. As before, we refer to X_g as the overflow queue. The PGF of X_g was first obtained in [64] and in Theorem 2.1 an alternative expression is given. The latter allows us to establish a heavy-traffic limit theorem for the overflow queue. We consider a heavy-traffic regime that connects the cycle length and the green period according to

$$g = c\mu + \beta\sigma\sqrt{c}. \tag{3.2}$$

Here σ denotes the standard deviation of Y and $\beta > 0$ is a parameter that can be chosen freely, and optimal choices for β will be obtained in Section 3.3.

The main intuition for considering the regime in Equation (3.2) is as follows. In heavy traffic, there will be many delayed cars, and during each cycle g delayed cars can depart while on average $c\mu$ new delayed cars will arrive. We therefore choose the green period as roughly $c\mu$, but add $\beta\sigma\sqrt{c}$ to account for variability of the number of newly arriving cars. Observe that for large cycles, $\beta\sigma\sqrt{c}$ will be considerably smaller than $c\mu$. In the heavy-traffic regime that we consider where c will be large, $c\mu$ is the dominant term and is needed to ensure stability, while $\beta\sigma\sqrt{c}$ is a hedge against uncertainty. To understand the effect of this hedge, substitute (3.2) into (3.1) to obtain $X_{g,n+1} = \max\{0, X_{g,n} + A_n - c\mu - \beta\sigma\sqrt{c}\}$. After dividing the term $A_n - c\mu - \beta\sigma\sqrt{c}$ by the standard deviation of the number of arrivals per cycle, $\sigma\sqrt{c}$, we expect it to be approximately normally distributed (with mean $-\beta$ and standard deviation 1) when c grows large because of the Central Limit Theorem (CLT), see e.g. [86, Section 5.10]. This is not entirely straightforward, because A_n cannot be interpreted as the sum of c independent random variables, and hence the CLT cannot be applied directly. We therefore resort to the transform method. We take the expression for the PGF of X_g established in Theorem 2.1, and show that this transform converges in the heavy-traffic regime (3.2) with $c \to \infty$ to the transform of a non-degenerate random variable M_β. The convergence of transforms then implies the convergence of the underlying random variables. Here,

M_β is a special random variable equal to the all-time maximum of the so-called Gaussian random walk with drift $-\beta$ and standard deviation 1, see e.g. [99] for a detailed study of various characteristics of M_β, including expressions and approximations for all moments. We will give more details on M_β later, and first present our main heavy-traffic limit theorem. Let $\xrightarrow{d}$ denote convergence in distribution.

Theorem 3.1 *Assume that $\mathbb{E}[z^Y]$ is analytic in a disk of radius R with $R > 1$, $\mu < 1$, and $\sigma^2 > 0$. Under scaling* (3.2)*, as $c \to \infty$,*

$$\frac{1}{\sigma\sqrt{c}} X_g \xrightarrow{d} M_\beta, \tag{3.3}$$

$$\mathbb{P}\left(\frac{X_g}{\sigma\sqrt{c}} = 0\right) = \mathbb{P}(M_\beta = 0)\left(1 + O\left(\frac{1}{\sqrt{c}}\right)\right), \tag{3.4}$$

and for $k \geq 1$,

$$\mathbb{E}[X_g^k] = \left(\sigma\sqrt{c}\right)^k \mathbb{E}[M_\beta^k]\left(1 + O\left(\frac{1}{\sqrt{c}}\right)\right). \tag{3.5}$$

The proof is deferred to Section 3.5. Theorem 3.1 has two practical implications. First, since the scaled overflow queue X_g converges to a non-degenerate limiting variable, the scaling rule (3.2) can serve as a guiding principle for choosing the cycle length as a function of the traffic pressure. That is, since there exists a non-degenerate limit, scaling rules that let g scale faster (e.g. $g = c\mu + \beta\sigma c^{2/3}$ or $g = (\mu + \beta)c$) or slower (e.g. $g = c\mu + \beta\sigma c^{1/3}$ or $g = c\mu + \beta$) likely lead to degenerate behavior in the large cycle limit $c \to \infty$, that is X_g converges with high probability to either 0 or ∞. The second practical implication is that known results for the limit M_β can be converted into approximations for X_g. As Theorem 3.1 suggests, for large enough c, we have

$$\mathbb{E}[X_g] \approx \sigma\sqrt{c}\,\mathbb{E}[M_\beta],$$
$$\mathbb{P}(X_g = 0) \approx \mathbb{P}(M_\beta = 0).$$

Let $\zeta(.)$ denote the Riemann zeta function. For $0 < \beta < 2\sqrt{\pi}$ it was shown in [99] that

$$\mathbb{E}[M_\beta] = \frac{1}{2\beta} + \frac{\zeta(1/2)}{\sqrt{2\pi}} + \frac{\beta}{4} + \frac{\beta^2}{\sqrt{2\pi}} \sum_{r=0}^{\infty} \frac{\zeta(-1/2 - r)}{r!(2r+1)(2r+2)} \left(\frac{-\beta^2}{2}\right)^r, \tag{3.6}$$

$$\mathbb{P}(M_\beta = 0) = \sqrt{2}\beta \exp\left\{\frac{\beta}{\sqrt{2\pi}} \sum_{r=0}^{\infty} \frac{\zeta(1/2 - r)}{r!(2r+1)} \left(\frac{-\beta^2}{2}\right)^r\right\}. \tag{3.7}$$

These expressions give heavy-traffic approximations for the overflow queue that are accurate when β is small and c is sufficiently large. Expression (3.6) also reveals that for small β, $\mathbb{E}[M_\beta] \approx 1/(2\beta)$, a particularly easy approximation that will be helpful when we optimize signal settings later in the chapter.

The random variable M_β

As mentioned before, the random variable M_β is equal to the all-time maximum of a Gaussian random walk with drift $-\beta$ and standard deviation 1. Perhaps the most intuitive way to understand *why* this process pops up, is to consider the following. In a continuous queueing model, a limiting process that is often encountered is the all-time maximum of the Brownian motion with negative drift (under a similar scaling as we use), see e.g. [83]. The normal distribution popping up, which has a strong connection with the Brownian motion, is no surprise either as the scaled version of the A_n's are approximately normally distributed with mean $-\beta$ and standard deviation 1. However, as we are not dealing with a continuous queueing model but with a queueing model in discrete time, we need to adjust the Brownian motion encountered in the scaling of continuous queueing models to a Gaussian random walk. As [208] states: "The only difference, one could say, is that Brownian motion is a continuous-time process, whereas the Gaussian random walk only changes at discrete points in time.", so in this sense, the fact that the Gaussian random walk turns up is not a surprise.

The all-time maximum of a Gaussian random walk with negative drift is a quite challenging process to understand. In contrast with the all-time maximum of Brownian motion with negative drift (which has an exponential distribution), the all-time maximum Gaussian random walk does not allow for easy and exact computations. Indeed, as detailed in e.g. Equation (3.6), we have an expression for $\mathbb{E}[M_\beta]$ in terms of infinite series or in terms of integrals as detailed in Proposition 3.2 below. For further properties of the process M_β we refer the interested reader to e.g. [99].

We also derive other approximations for $\mathbb{E}[X_g]$ and $\mathbb{P}(X_g = 0)$ that are more accurate, in particular for smaller c and larger β. Let us introduce the integrals

$$G_0(b) = \int_0^\infty \frac{t^2}{b^2+t^2} \frac{\mathrm{e}^{-b^2-t^2}}{1-\mathrm{e}^{-b^2-t^2}} \mathrm{d}t,$$

$$G_1(b) = \int_0^\infty \frac{\mathrm{e}^{-b^2-t^2}}{1-\mathrm{e}^{-b^2-t^2}}\mathrm{d}t,$$

that can be computed numerically by standard software packages. In addition, [101] provides ζ-series such as in Equations (3.6) and (3.7), for $G_0(b)$ and $G_1(b)$ as well as rapidly convergent series involving the standard Gaussian and the complementary error function, see Equations (4.27), (4.29), and (4.31) in [101]. One consequence of the results of the latter type is the series representation

$$G_0'(b) = -\sqrt{\pi}\sum_{k=0}^{\infty}\int_{b\sqrt{k+1}}^{\infty}\mathrm{e}^{-t^2}\mathrm{d}t, \tag{3.8}$$

that shows that $G_0'(b)$ is negative and strictly increasing in $b>0$, which will be used later on. We prove the following result in Appendix 3.A.

Proposition 3.2 *The mean overflow queue satisfies, as $c\to\infty$,*

$$\mathbb{E}[X_g] = \frac{\sqrt{2}}{\pi}\left(\sigma\sqrt{c}+\frac{\beta\sigma^2}{2\mu}\right)G_0(b(\beta)) + \frac{\theta\beta}{\pi}G_1\left(\frac{\beta}{\sqrt{2}}\right) + O\left(\frac{1}{\sqrt{c}}\right), \tag{3.9}$$

where

$$b(\beta) = \frac{\beta}{\sqrt{2}}\left(1+\frac{\beta\sigma}{\mu\sqrt{c}}\right)^{-1/2},$$

$$a = \frac{\mu_3-\mu^3-3(1+\mu)\sigma^2}{\mu}, \tag{3.10}$$

$$\theta = \frac{\sigma^2}{\mu\sqrt{2}}\left(\frac{\mu}{\sigma^2}+\frac{1}{3}\left(\frac{\mu}{\sigma^2}\right)^2 a - 1\right), \tag{3.11}$$

with μ_3 the third moment of Y.

A direct consequence of Proposition 3.2 is the slightly easier approximation

$$\mathbb{E}[X_g] = \frac{\sqrt{2}}{\pi}\sigma\sqrt{c}\,G_0\left(\frac{\beta}{\sqrt{2}}\right) + O(1). \tag{3.12}$$

Tables 3.1 and 3.2 show the asymptotic approximations we have just derived. We assume here that c is allowed to be any real positive number. As we couple g and c as in Equation (3.2), often either g or c is non-integer. It is easy to deal with non-integer c, as we can simply adjust the $Y(z)^c$ term to account for

Table 3.1: Various results for $\mathbb{P}(X_g = 0)$ and $\mathbb{E}[X_g]$ for several values of g and c with Poisson arrivals with mean $\mu = 0.3$ in each slot and with $\beta = 0.1$.

		$\mathbb{P}(X_g = 0)$	
g	c	true value	Approx. (3.7)
10	32.3	0.1649	0.1334
20	65.2	0.1551	0.1334
30	98.2	0.1509	0.1334
50	164.3	0.1468	0.1334
100	330.0	0.1427	0.1334
200	662.0	0.1399	0.1334
500	1659.2	0.1375	0.1334
		$\mathbb{E}[X_g]$	
g	c	true value	Approx. (3.9)	Approx. (3.12)
10	32.3	13.935	13.985	13.826
20	65.2	19.767	19.803	19.644
30	98.2	24.238	24.267	24.109
50	164.3	31.324	31.346	31.188
100	330.0	44.340	44.356	44.198
200	662.0	62.744	62.754	62.597
500	1659.2	99.254	99.261	99.104

non-integer c, see also extension (iv) in Subsection 2.3.2. As expected, the approximations become more accurate for larger c. However, the approximations also serve as useful, somewhat looser approximations for small and moderate values of c. In conclusion, we have derived two asymptotic approximations for $\mathbb{E}[X_g]$, the first-order approximation (3.12) with error $O(1)$, and the refined approximation (3.9) with error $O(1/\sqrt{c})$. Although both approximations perform well already for small values of c, (3.9) is more accurate than (3.12) for values of β as large as 2. Both approximations will be employed in the next section for the purpose of solving optimization problems.

3.3 Capacity allocation problems

We now turn to optimal green-time allocations for an intersection with n lanes, where each lane is modeled separately as an FCTL queue. Let μ_i denote the

Table 3.2: Various exact results for $\mathbb{P}(X_g = 0)$ and $\mathbb{E}[X_g]$ for several values of g and c with Poisson arrivals with mean $\mu = 0.3$ in each slot and with $\beta = 1$.

		$\mathbb{P}(X_g = 0)$	
g	c	true value	Approx. (3.7)
10	24.3	0.8450	0.8005
20	53.3	0.8312	0.8005
30	83.3	0.8253	0.8005
50	144.7	0.8200	0.8005
100	301.6	0.8138	0.8005
200	621.2	0.8098	0.8005
500	1593.8	0.8063	0.8005

		$\mathbb{E}[X_g]$		
g	c	true value	Approx. (3.9)	Approx. (3.12)
10	24.3	0.3944	0.4437	0.3414
20	53.3	0.5664	0.5996	0.5055
30	83.3	0.6960	0.7225	0.6319
50	144.7	0.8998	0.9199	0.8326
100	301.6	1.2722	1.2860	1.2021
200	621.2	1.7971	1.8001	1.7251
500	1593.8	2.8369	2.8387	2.7633

mean number of arrivals per slot at lane i, σ_i the standard deviation of the number of arrivals per slot at lane i, g_i the green time allocated to lane i within one cycle of length c, and $\mathbb{E}[X_{g_i}^{(i)}]$ the mean overflow queue at lane i. While the lanes operate independently once the green times are fixed, they do depend on each other through the cycle time c and, obviously, the green time of one lane corresponds to a red period for the other lanes. We leverage this independence across lanes and the asymptotic approximations developed in Section 3.2 to formulate several optimization problems that search for the vector of green times that minimizes the total expected overflow queue.

3.3.1 Minimizing the sum of overflows

Consider the problem of finding the green times that minimize the sum of the mean queue lengths at the end of the green periods $\sum_{i=1}^{n} \mathbb{E}[X_{g_i}^{(i)}]$. Assume that c is fixed, and let $r_T < c$ represent the time that cannot be used as green time.

This r_T could e.g. model clearing times between lanes. Hence, we have that $c = r_T + \sum_{i=1}^{n} g_i$. Again applying the substitution as in (3.2), $g_i = \mu_i c + \beta_i \sigma_i \sqrt{c}$, for $i = 1, \ldots, n$, this gives the following optimization problem:

$$\begin{aligned} \underset{\beta_1,\ldots,\beta_n}{\text{minimize}} \quad & \sum_{i=1}^{n} \mathbb{E}[X_{g_i}^{(i)}] \\ \text{subject to} \quad & \sum_{i=1}^{n} \beta_i \sigma_i \sqrt{c} = c(1-\mu_T) - r_T; \\ & \beta_i > 0,\ i = 1, \ldots, n, \end{aligned} \tag{3.13}$$

with $\mu_T = \sum_{i=1}^{n} \mu_i$. The first constraint in (3.13) relates to the requirement $c = r_T + \sum_{i=1}^{n} g_i$ and together with the constraints $\beta_i > 0$ for all i, it is ensured that each g_i might be chosen so as to ensure a vehicle-to-capacity ratio less than 1 for each lane as $c(1-\mu_T) - r_T > 0$.

Optimization problem (3.13) seems mathematically intractable due to the lack of an explicit expression for the objective function $\sum_{i=1}^{n} \mathbb{E}[X_{g_i}^{(i)}]$. We shall therefore use approximations based on Equations (3.12) and (3.9) to replace the objective function with a heavy-traffic approximation, which then leads to a tractable, more structured optimization problem.

Using (3.12) gives the following optimization problem:

$$\begin{aligned} \underset{\beta_1,\ldots,\beta_n}{\text{minimize}} \quad & \sum_{i=1}^{n} \frac{\sigma_i}{\pi} \sqrt{2c}\, G_0(\beta_i/\sqrt{2}) \\ \text{subject to} \quad & \sum_{i=1}^{n} \beta_i \sigma_i \sqrt{c} = c(1-\mu_T) - r_T; \\ & \beta_i > 0,\ i = 1, \ldots, n. \end{aligned} \tag{3.14}$$

Theorem 3.3 *Optimization problem* (3.14) *is solved by setting*

$$\beta_i = \frac{c(1-\mu_T) - r_T}{\sqrt{c} \sum_{j=1}^{n} \sigma_j} =: \beta_*. \tag{3.15}$$

Proof. Introduce the Lagrange multiplier $\lambda_0 \in \mathbb{R}$, so that

$$\frac{\partial}{\partial \beta_i} \left(\sum_{j=1}^{n} \frac{\sigma_j \sqrt{2c}}{\pi} G_0(\beta_j/\sqrt{2}) \right) = \lambda_0 \frac{\partial}{\partial \beta_i} \left(\sum_{j=1}^{n} \beta_j \sigma_j \sqrt{c} - c(1-\mu_T) + r_T \right),$$

for $i = 1, \ldots, n$. This gives

$$G_0'(\beta_i/\sqrt{2}) = \pi\lambda_0. \tag{3.16}$$

The function $G_0'(b)$ is negative and strictly increasing in $b > 0$, see Equation (3.8). Combining this with the fact that λ_0 is independent of the index i, we conclude that the β_i are the same for $i = 1, \ldots, n$ and should satisfy

$$\beta_i\sqrt{c}\sum_{j=1}^{n}\sigma_j = c(1-\mu_T) - r_T,$$

for $i = 1, \ldots, n$, which completes the proof. □

Theorem 3.3 shows that the optimal parameters β_i should be equal for all lanes. We now turn to the second approximation for the problem formulated in Equation (3.13), based on the refined heavy-traffic approximation in (3.9):

$$\begin{aligned} \underset{\beta_1,\ldots,\beta_n}{\text{minimize}} \quad & \sum_{i=1}^{n}\frac{\sqrt{2}}{\pi}\left(\sigma_i\sqrt{c} + \frac{\beta_i\sigma_i^2}{2\mu_i}\right)G_0\left(b_i(\beta_i)\right) + \frac{\theta_i\beta_i}{\pi}G_1\left(\frac{\beta_i}{\sqrt{2}}\right) \\ \text{subject to} \quad & \sum_{i=1}^{n}\beta_i\sigma_i\sqrt{c} = c(1-\mu_T) - r_T; \\ & \beta_i > 0,\ i = 1, \ldots n. \end{aligned} \tag{3.17}$$

Theorem 3.4 *Optimization problem* (3.17) *is solved by*

$$\beta_i = \beta_* + \Omega_i(\beta_*), \quad i = 1, \ldots, n, \tag{3.18}$$

with β_* *as in* (3.15)*,*

$$\Omega_i(\beta_*) = \sqrt{\frac{2}{c}}\frac{1}{G_0''\left(\beta_*/\sqrt{2}\right)}\left(\frac{\sum_{j=1}^{n}K_j}{\sum_{j=1}^{n}\sigma_j} - \frac{K_i}{\sigma_i}\right),$$

and

$$\begin{aligned} K_i = & \frac{\sigma_i^2}{\sqrt{2}\mu_i}G_0\left(\frac{\beta_*}{\sqrt{2}}\right) - \frac{\beta_*\sigma_i^2}{2\mu_i}G_0'\left(\frac{\beta_*}{\sqrt{2}}\right) - \frac{\beta_*^2\sigma_i^2}{2\sqrt{2}\mu_i}G_0''\left(\frac{\beta_*}{\sqrt{2}}\right) + \\ & \theta_i G_1\left(\frac{\beta_*}{\sqrt{2}}\right) + \frac{\theta_i\beta_*}{\sqrt{2}}G_1'\left(\frac{\beta_*}{\sqrt{2}}\right). \end{aligned} \tag{3.19}$$

The proof of Theorem 3.4 is presented in Appendix 3.A. The result may seem complicated at first glance, but in fact reveals a remarkably elegant structure. The term $\Omega_i(\beta_*)$ can be thought of as a refinement of β_*, due to using the refined approximation (3.9) instead of (3.12). An intriguing finding is that $\Omega_i(\beta_*)$ can be written explicitly in terms of β_*. From that perspective, the rule in (3.18) can be interpreted as a two-step procedure. First divide the green time into parts of length $g_i = \mu_i c + \beta_* \sigma_i \sqrt{c}$, and then correct or refine this division using $\beta_* + \Omega_i(\beta_*)$ instead of β_*. Note that in this second step, lane i gets a larger or smaller share depending on the sign of

$$\frac{\sum_{j=1}^n K_j}{\sum_{j=1}^n \sigma_j} - \frac{K_i}{\sigma_i}.$$

Generally, the solution to optimization problem (3.17) will lead to more accurate results for the minimization of $\sum_{i=1}^n \mathbb{E}[X_{g_i}^{(i)}]$ than the solution to the optimization problem (3.14), as the approximation of the individual $\mathbb{E}[X_{g_i}^{(i)}]$ terms is more accurate. We return to this observation in Section 3.4, Example 1.

Remark 3.1 *The solutions of both optimization problems formulated above generally result in non-integer values for g_i. Depending on the exact setting, we might opt, e.g., for rounding the values to the nearest integer or rounding the value of g_i down (along with checking for stability). An alternative procedure is to allow for a random green time. If G_i denotes such a random green time, we can choose it in the following way: G_i is equal to $\lfloor g_i \rfloor$ with probability p and equal to $\lceil g_i \rceil$ with probability $1-p$ such that $g_i = p\lfloor g_i \rfloor + (1-p)\lceil g_i \rceil$. We show how this can be accounted for in Remark 3.2.*

3.3.2 Minimizing the weighted sum of overflows

In practice, it might be preferable to give more priority to certain lanes, which can be modeled by introducing weights associated with each lane. Assume that lane i gets weight $d_i > 0$ and formulate the optimization problem

$$\begin{aligned} \underset{\beta_1,\ldots,\beta_n}{\text{minimize}} \quad & \sum_{i=1}^n d_i \mathbb{E}[X_{g_i}^{(i)}] \\ \text{subject to} \quad & \sum_{i=1}^n \beta_i \sigma_i \sqrt{c} = c(1-\mu_T) - r_T; \\ & \beta_i > 0,\ i = 1,\ldots,n. \end{aligned} \tag{3.20}$$

Due to the weights $d_1, \ldots, d_n$ we cannot (approximately) solve the problem (3.20) explicitly with the same heavy-traffic approximations that are used in (3.14) and (3.17). We therefore resort to the approximation $\mathbb{E}[X_{g_i}^{(i)}] \approx \sigma_i \sqrt{c}/(2\beta_i)$ as derived from Equation (3.6), and solve the problem

$$\begin{aligned} \underset{\beta_1,\ldots,\beta_n}{\text{minimize}} \quad & \sum_{i=1}^{n} d_i \frac{\sqrt{c}\sigma_i}{2\beta_i} \\ \text{subject to} \quad & \sum_{i=1}^{n} \beta_i \sigma_i \sqrt{c} = c(1-\mu_T) - r_T; \\ & \beta_i > 0,\ i = 1, \ldots, n. \end{aligned} \tag{3.21}$$

Proposition 3.5 *Optimization problem* (3.21) *is solved by*

$$\beta_i = \frac{\sqrt{d_i}(c(1-\mu_T) - r_T)}{\sqrt{c}\sum_{j=1}^{n} \sqrt{d_j}\sigma_j}. \tag{3.22}$$

Proof. Follows from the same Lagrange multiplier technique as in the proof of Theorem 3.3. □

Equation (3.22) reduces to Equation (3.15) for $d_i = 1$.

We next use a more accurate approximation for the $\mathbb{E}[X_{g_i}^{(i)}]$ and define the following minimization problem.

$$\begin{aligned} \underset{\beta_1,\ldots,\beta_n}{\text{minimize}} \quad & \sum_{i=1}^{n} d_i \frac{\sigma_i}{\pi} \sqrt{2c}\, G_0(\beta_i/\sqrt{2}) \\ \text{subject to} \quad & \sum_{i=1}^{n} \beta_i \sigma_i \sqrt{c} = c(1-\mu_T) - r_T; \\ & \beta_i > 0,\ i = 1, \ldots, n. \end{aligned} \tag{3.23}$$

Corollary 3.6 *There exists a unique solution to optimization problem* (3.23)*, which can be obtained numerically.*

Proof. Along the same lines as in the proof of Theorem 3.3, we get that there exists a Lagrange multiplier $\lambda_0 \in \mathbb{R}$ satisfying

$$G_0'(\beta_j/\sqrt{2}) = \frac{\pi\lambda_0}{d_j\sqrt{c}}. \tag{3.24}$$

As G_0' is a strictly increasing function, it is invertible and thus Equation (3.24) can be solved for β_j. This implies that a Lagrange multiplier λ_0 exists, that the

problem formulated in Equation (3.23) is solvable, that the solution is unique, and that the optimal values can be obtained numerically. □

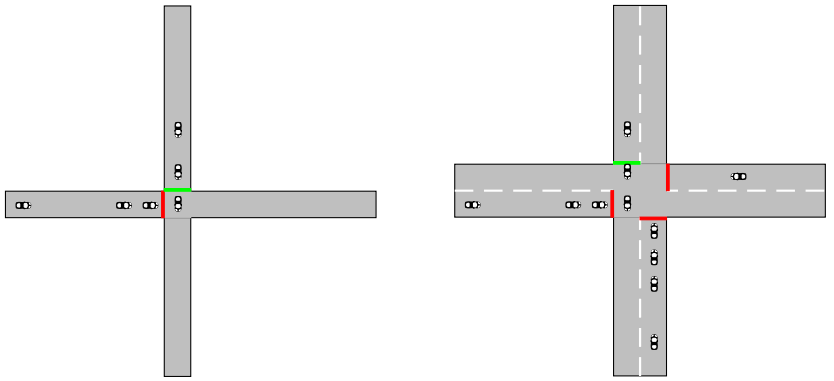
While the minimization problem in (3.23) cannot be solved analytically, Corollary 3.6 implies that a numerical solution can be found.

3.4 Numerical examples of capacity allocation

We now numerically investigate the capacity allocation procedures developed in Section 3.3, that in turn use the asymptotic approximations for the mean overflow established in Section 3.2. In particular, the first-order approximation (3.12) and the refined approximation (3.9) were both used to solve capacity allocation problems in the asymptotic regime where cycle times become large. This led to asymptotic dimensioning rules that prescribe how to divide the cycle time over the various lanes, and in particular how to choose the green time in an (asymptotically) optimal manner. Because the capacity allocation problems in Section 3.3 were solved analytically, we have conducted many numerical experiments for assessing the effectiveness of the asymptotic results for various cycle lengths and distributional assumptions on the arrival processes. From these many experiments, we concluded that the asymptotic dimensioning rules perform well, also for settings with a small or moderate cycle length and/or relatively small vehicle-to-capacity ratios. We shall now substantiate these findings by discussing two examples in more detail.

3.4.1 Two-lane example

First consider an example with two lanes as depicted in Figure 3.1(a). Due to the fixed cycle, both lanes operate as independent FCTL queues. The challenge, however, is to determine the optimal capacity allocation that dictates how the cycle time should be divided. In this example, we set the cycle length according to the sum of the green times and we choose an all-red or clearance time of r_T slots. We consider Poisson arrivals at lane 1 and geometric arrivals at lane 2, both with a mean arrival rate of 0.4 vehicles per slot. We further choose $r_T = 5$ and study various values of c. We determine the optimal β_i according to the first-order dimensioning rule in (3.15) and the refined rule in (3.18). In Table 3.3 we display the optimal β_i according to the two dimensioning rules together with the resulting green times. We see in Table 3.3 that the green times only weakly depend on the distribution (Poisson or geometric). This, at least partly, relates to the scaling rule (3.2) that we propose: if the mean arrival



(a) (b)

Figure 3.1: Graphical representations of (a) the two-lane example considered in Subsection 3.4.1 and (b) the four-lane example in Subsection 3.4.2.

rate of two vehicle streams is the same (as is the case in this example), the only difference in the green time is caused by differences in the standard deviation of the arrival processes and by the parameters β_i. The latter are the same for all flows under the dimensioning rule as in (3.15) and only differ slightly under the dimensioning rule as in (3.18).

As such, the difference between the green times based on the first-order dimensioning rule and the refined dimensioning rule in Table 3.3 is generally small. These small differences in the green-time allocations can be explained

Table 3.3: Optimal green times and β_i's according to Theorem 3.3 (rule (3.15)) and Theorem 3.4 (rule (3.18)). For rule (3.18) we use the notation β_i^c and g_i^c. We consider a Poisson arrival stream with mean 0.4 at lane 1, a geometric arrival stream at lane 2 with mean 0.4, and $r_T = 5$. We study various values of c.

	Dimensioning rule (3.15)		Dimensioning rule (3.18)	
c	g_1 (β_1)	g_2 (β_2)	g_1^c (β_1^c)	g_2^c (β_2^c)
30	12.46 (0.132)	12.54 (0.132)	12.46 (0.132)	12.54 (0.132)
50	22.29 (0.512)	22.71 (0.512)	22.29 (0.511)	22.71 (0.513)
100	46.87 (1.086)	48.13 (1.086)	46.84 (1.082)	48.16 (1.090)
200	96.03 (1.792)	98.97 (1.792)	95.92 (1.780)	99.08 (1.803)
500	243.51 (3.077)	251.49 (3.077)	243.11 (3.049)	251.89 (3.101)

Table 3.4: Exact values of the mean overflow queue with the green time based on Theorem 3.3, $\mathbb{E}[X_{g_i}^{(i)}]$, and on Theorem 3.4, $\mathbb{E}[X_{g_i^c}^{(i)}]$, respectively. The green times are randomized as in Remark 3.1. The table also displays an approximation of the mean overflow queue based on Equation (3.12) for the g_i and an approximation based on Equation (3.9) for the g_i^c. The results are for Poisson arrivals with mean 0.4 at lane 1, geometric arrivals with mean 0.4 at lane 2, and $r_T = 5$. We study various values of c.

c	$\mathbb{E}[X_{g_1}^{(1)}]$	Eq. (3.12)	$\mathbb{E}[X_{g_1^c}^{(1)}]$	Eq. (3.9)
30	11.53	11.19	11.53	11.35
50	2.396	2.285	2.402	2.417
100	0.6978	0.6383	0.7066	0.7286
200	0.1686	0.1431	0.1742	0.1887
500	0.00609	0.00412	0.00666	0.00960
c	$\mathbb{E}[X_{g_2}^{(2)}]$	Eq. (3.12)	$\mathbb{E}[X_{g_2^c}^{(2)}]$	Eq. (3.9)
30	13.60	13.24	13.60	13.52
50	2.870	2.704	2.863	2.923
100	0.8577	0.7553	0.8500	0.8930
200	0.2156	0.1693	0.2104	0.2343
500	0.00865	0.00488	0.00801	0.0124

by the fact that the first-order approximation for the mean overflow queue is already sharp, see Table 3.4, where we take the various green-time allocations as in Table 3.3 while randomizing the green times as in Remark 3.1, and compute the exact value and approximations for the mean overflow queue. The minor differences in the green-time allocations in Table 3.3 also lead to relatively small differences in the mean overflow queue as can be observed in Table 3.4. The larger green times are allocated to the flow with the larger standard deviation of the number of arrivals per slot (and thus also to the flow with the larger mean overflow queue), which makes sense intuitively: if there is any excess green time, it should be allocated to the longest queue (within certain boundaries). We also found the optimal integer green-time allocation for the cases studied in Table 3.3 and the optimal green times generally agree with the rounded values of the non-integer green times presented in Table 3.3, certainly when using (3.18). Summarizing, both dimensioning rules yield results that are close to the optimum. As a last remark, we note that the first-order rule (3.15) is already a good way of dimensioning this two-lane intersection.

3.4.2 Four-lane example with weights

We next consider the influence of weights for an intersection with four lanes, see also Figure 3.1(b), again assuming an all-red time r_T of 5 slots. We apply the dimensioning rule in (3.23) and obtain the optimal β_i numerically (see Corollary 3.6).

We show results for equal weights $d_i = 1$ in Table 3.5 and unequal weights $d_i = i$ in Table 3.6. We assume geometrically distributed arrivals at lane 1 with mean 0.3 and Poisson arrivals at lanes 2, 3, and 4 with means 0.3, 0.1, and 0.1 respectively. We display the green time and the optimal β_i for each lane in both tables. With equal weights, the β_i are the same and the difference in green times is solely due to differences in the mean and the standard deviation of the arrival process because the β_i are all the same, see Table 3.5. With unequal weights, the β_i increase with the weight d_i, as expected, although the influence of the weights on the green times remains limited as can be observed in Table 3.6. This makes sense, since the amount of green time that one can freely allocate is rather limited as well, especially for small c. E.g., if $c = 30$, we only have one green slot to allocate freely (since we need $\mu_T c = 24$ for stabilizing all flows and r_T being equal to 5). This is clearly visible in Tables 3.5 and 3.6. If c increases, the number of green slots that we can allocate freely increases, e.g. if $c = 500$ we can distribute 95 slots to minimize the weighted sum of the mean overflow queues. In this case, we thus see a bigger, although still rather small, difference between the case where $d_i = 1$ in Table 3.5 and the case with $d_i = i$ in Table 3.6. We also computed the optimal integer green-time allocation for the cases studied in Tables 3.5 and 3.6 and they mostly coincide with the rounded green times that we obtain when we round the obtained green times

Table 3.5: Dimensioning rule (3.23). Optimal green times and the β_i's for a four-lane example with geometric arrivals with mean 0.3 in lane 1, Poisson arrivals with mean 0.3 in lane 2, Poisson arrivals with mean 0.1 in lane 3, Poisson arrivals with mean 0.1 in lane 4, with $r_T = 5$, and $d_i = 1$ for various values of c.

c	g_1 (β_1)	g_2 (β_2)	g_3 (β_3)	g_4 (β_4)
30	9.346 (0.101)	9.304 (0.101)	3.175 (0.101)	3.175 (0.101)
50	16.73 (0.392)	16.52 (0.392)	5.876 (0.392)	5.876 (0.392)
100	35.19 (0.831)	34.55 (0.831)	12.63 (0.831)	12.63 (0.831)
200	72.11 (1.371)	70.62 (1.371)	26.13 (1.371)	26.13 (1.371)
500	182.9 (2.354)	178.8 (2.354)	66.65 (2.354)	66.65 (2.354)

Table 3.6: Dimensioning rule (3.23). Optimal green times and the β_i's for a four-lane example with geometric arrivals with mean 0.3 in lane 1, Poisson arrivals with mean 0.3 in lane 2, Poisson arrivals with mean 0.1 in lane 3, Poisson arrivals with mean 0.1 in lane 4, with $r_T = 5$, and $d_i = i$ for various values of c.

c	g_1 (β_1)	g_2 (β_2)	g_3 (β_3)	g_4 (β_4)
30	9.243 (0.071)	9.300 (0.100)	3.212 (0.123)	3.245 (0.141)
50	16.24 (0.280)	16.51 (0.390)	6.053 (0.471)	6.199 (0.536)
100	33.93 (0.629)	34.58 (0.836)	13.08 (0.975)	13.41 (1.079)
200	69.88 (1.119)	70.75 (1.388)	26.93 (1.549)	27.44 (1.664)
500	179.6 (2.122)	179.1 (2.375)	67.79 (2.516)	68.48 (2.614)

in Tables 3.5 and 3.6. The main source for differences seems to be the rounding effect, often causing the rounded green times, $\tilde{g}_i$, to add up to $c - \sum_i \tilde{g}_i = r_T - 1$ rather than r_T. Modulo this effect, the optimal green times and the obtained green times in Tables 3.5 and 3.6 coincide up to one slot. This indicates that our dimensioning rules are, again, yielding close-to-optimal results while being easy to compute and interpret in terms of the input parameters.

3.5 Proof of heavy-traffic theorem using the transform method

In this section, we present the proof of Theorem 3.1, which we regard as the main mathematical novelty in this chapter. The theorem shows weak convergence of the scaled overflow queue to a non-degenerate limit. The general proof structure is explained in Subsection 3.5.1 and executed in Subsection 3.5.2.

3.5.1 Sketch of the proof of Theorem 3.1

As before, let $X_g(w)$ denote the PGF of the stationary overflow queue. In Theorem 2.1, we derived that there is an $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0)$

$$X_g(w) = \exp\left(\frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \frac{(Y'(z)z - Y(z))(w - Y(w))}{(z - Y(z))(zY(w) - wY(z))} \ln\left(1 - \frac{Y^c(z)}{z^g}\right) \mathrm{d}z\right), \tag{3.25}$$

for any $|w| < 1+\epsilon$ with principal value of the logarithm and where $Y(z)$ is the PGF of the number of arrivals in a single slot. We switch to the moment generating

function (MGF) by a change of variables, replacing w by $\exp(t/(\sigma\sqrt{c}))$.

We will prove that the MGF of the FCTL overflow queue converges to the MGF of the M_β given by, see [1],

$$\mathbb{E}[\mathrm{e}^{tM_\beta}] = \exp\left(\frac{1}{2\pi i}\int_{\mathscr{C}} \frac{t}{u(t-u)} \ln\left(1-\mathrm{e}^{-\beta u+\frac{1}{2}u^2}\right)\mathrm{d}u\right),$$

where $t \in \mathbb{C}$ and $\mathscr{C}$ is a curve going from $-i\cdot\infty$ to $+i\cdot\infty$, passing t to the right. We choose $\mathscr{C}: u = \beta + iv$, $-\infty < v < \infty$, and then we get for $\mathrm{Re}(t) < \beta$ that

$$\mathbb{E}[\mathrm{e}^{tM_\beta}] = \exp\left(\frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{t}{(\beta+iv)(t-\beta-iv)} \ln\left(1-\mathrm{e}^{-\frac{1}{2}\beta^2-\frac{1}{2}v^2}\right)\mathrm{d}v\right). \tag{3.26}$$

Then, we will prove that

$$X_g(w) = \mathbb{E}\left[\mathrm{e}^{tM_\beta}\right]\left(1+O\left(\frac{1}{\sqrt{c}}\right)\right), \tag{3.27}$$

with $w = \exp(t/(\sigma\sqrt{c}))$, as $c\to\infty$, uniformly in t in any bounded set contained in $\mathrm{Re}(t) \le \frac{1}{2}\beta$, proving Equation (3.3) in Theorem 3.1. We work from the integral

$$I_c(w) := \frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \frac{Y'(z)z - Y(z)}{z-Y(z)} \frac{w-Y(w)}{zY(w)-wY(z)} \ln\left(1-\frac{Y^c(z)}{z^g}\right)\mathrm{d}z, \tag{3.28}$$

with $w = \exp(t/(\sigma\sqrt{c}))$, see Equation (3.25), towards the integral

$$J(t) := \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{t}{(\beta+iv)(t-\beta-iv)} \ln\left(1-\mathrm{e}^{-\frac{1}{2}\beta^2-\frac{1}{2}v^2}\right)\mathrm{d}v,$$

see Equation (3.26). We do this by using the dedicated saddle point method presented in [101] for the bulk-service queue in heavy traffic. To avoid certain technical complications, we assume, as in [101], that the maximum of $|Y(z)|$ over z, $|z| = r$, is uniquely achieved at $z = r$ for any $r \in (0, R)$. Under this assumption, see [101], the function

$$h(z) := -\ln z + \frac{c}{g}\ln Y(z) \tag{3.29}$$

has a unique saddle point z_{sp} in $(1, R)$ with

$$h(z_{sp}) < 0 = h'(z_{sp}),$$
$$h''(z_{sp}) \to \frac{\sigma^2}{\mu},$$

when $c \to \infty$ and such that $\mathrm{Re}[h(z)]$, $|z| = z_{sp}$, is strictly maximal at $|z| = z_{sp}$. This saddle point converges to 1 as $c \to \infty$, and $z_{sp} < z_0$, where z_0 is the zero of $z^g - Y^c(z)$ outside the unit disk of smallest modulus. We shall take $1 + \epsilon = z_{sp}$ in Equation (3.28). As $c \to \infty$, we have, due to rapid decay of $|Y^c(z)/z^g|$ along $|z| = z_{sp}$ from $z = z_{sp}$ onwards, that we may restrict the integration over z in Equation (3.28) to only a small portion of $|z| = z_{sp}$ near $z = z_{sp} \to 1$. Also, we have $w = \exp(t/(\sigma\sqrt{c})) \to 1$, $c \to \infty$, since t is in a bounded set.

Our proof has the following main steps.

I. Approximating the integrand in Equation (3.28)

$$\frac{Y'(z)z - Y(z)}{z - Y(z)} \frac{w - Y(w)}{zY(w) - wY(z)} \text{ by } \frac{w-1}{(z-1)(w-z)} \tag{3.30}$$

for z and w near 1.

II. Substituting $z = z(x)$, $-\delta \le x \le \delta$ with $z(0) = z_{sp}$ to achieve that

$$\frac{Y^c(z(x))}{(z(x))^g} = \exp\left(gh(z_{sp}) - \frac{1}{2}gh''(z_{sp})x^2\right) \tag{3.31}$$

assumes the form of a Gaussian (steepest descent curve).

III. Showing that

$$\begin{aligned} gh(z_{sp}) &\to -\frac{1}{2}\beta^2, \\ h''(z_{sp}) &\to \frac{\sigma^2}{\mu}, \end{aligned} \tag{3.32}$$

as $c \to \infty$. Substituting $v = x\sqrt{gh''(z_{sp})}$, $-\delta \le x \le \delta$, we see from Equations (3.31) and (3.32), that we approximate

$$\ln\left(1 - \frac{Y^c(z(x))}{(z(x))^g}\right) \text{ by } \ln\left(1 - e^{-\frac{1}{2}\beta^2 - \frac{1}{2}v^2}\right) \tag{3.33}$$

as $c \to \infty$.

IV. Showing that the total effect on $(w-1)/((z-1)(w-z))$ in Equation (3.30) of the substitutions $z = z(x)$, $v = x\sqrt{gh''(z_{sp})}$ amounts to approximating

$$\frac{(w-1)\mathrm{d}x}{(z-1)(w-z)} \text{ by } \frac{t\mathrm{d}v}{(\beta + iv)(t - \beta - iv)},$$

where $w = \exp(t/(\sigma\sqrt{c}))$ and $c \to \infty$.

V. Completing the proof of Equation (3.27).

3.5.2 Full proof of Theorem 3.1

We shall next present the details for the five main steps.
Step I. We have in $|z-1| \le \frac{1}{2}(R-1) =: \delta$

$$\begin{aligned} Y(z) &= 1 + \mu(z-1) + O(|z-1|^2), \\ Y'(z) &= \mu + O(|z-1|), \end{aligned} \tag{3.34}$$

so that

$$\begin{aligned} z - Y(z) &= (1-\mu)(z-1)(1+O(|z-1|), \\ zY'(z) - Y(z) &= -(1-\mu)(1+O(|z-1|)). \end{aligned}$$

Therefore, in a set of z's, $|z-1| \le \delta_1$ with $\delta_1 > 0$,

$$\frac{zY'(z) - Y(z)}{z - Y(z)} = \frac{-1}{z-1}\big(1 + O(|z-1|)\big). \tag{3.35}$$

We shall show below that for $|z-1|$ and $|w-1| \le \frac{1}{2}(R-1) = \delta$, we have that:

$$zY(w) - wY(z) = (1-\mu)(z-w)\big(1 + O(|z-1| + |w-1|)\big). \tag{3.36}$$

Therefore, also using Equation (3.34) with w instead of z,

$$\frac{w - Y(w)}{zY(w) - wY(z)} = \frac{(1-\mu)(w-1)\big(1+O(|w-1|)\big)}{(1-\mu)(z-w)\big(1+O(|z-1|+|w-1|)\big)} \tag{3.37}$$

holds in a set of z's and w's, $|z-1| \le \delta_2$ and $|w-1| \le \delta_2$ with $\delta_2 > 0$. Combining Equations (3.35) and (3.37), we get

$$\frac{Y'(z)z - Y(z)}{z - Y(z)} \frac{w - Y(w)}{zY(w) - wY(z)} = \frac{w-1}{(z-1)(w-z)}\big(1 + O(|z-1| + |w-1|)\big), \tag{3.38}$$

holding in a set of z's and w's, $|z-1| \le \delta_3$ and $|w-1| \le \delta_3$ with $\delta_3 > 0$.

We finally show that Equation (3.36) holds when $|z-1|$ and $|w-1| \le \delta$. We have

$$Y(v) = 1 + \mu(v-1) + \sum_{k=2}^{\infty} c_k (v-1)^k,$$

for $|v-1|<\delta$ and where $\sum_{k=2}^{\infty}|c_k(v-1)^k| \le \sum_{k=2}^{\infty} k|c_k|\delta^k < \infty$, and so

$$zY(w) - wY(z) = (1-\mu)(z-w) + \sum_{k=2}^{\infty} c_k \Big(z(w-1)^k - w(z-1)^k \Big). \tag{3.39}$$

For $k = 2,3,\ldots$, we have

$$z(w-1)^k - w(z-1)^k =$$
$$(w-1)^k - (z-1)^k + (z-1)(w-1)\Big((w-1)^{k-1} - (z-1)^{k-1}\Big).$$

Using $a^n - b^n = (a-b)\sum_{i=0}^{n-1} a^i b^{n-1-i}$ with $a = w-1$, $b = z-1$, and $n = k, k-1$, we get

$$z(w-1)^k - w(z-1)^k = \tag{3.40}$$
$$(w-z)\left[\sum_{j=0}^{k-1}(w-1)^j(z-1)^{k-1-j} + \sum_{j=0}^{k-2}(w-1)^{j+1}z^{k-1-j}\right].$$

Let $m = \max\{|z-1|, |w-1|\}$. The modulus of the quantity within the [] of the right-hand side of Equation (3.40) is bounded by

$$km^{k-1} + (k-1)m^k \le (|z-1| + |w-1|)\,(k\delta^{k-2} + (k-1)\delta^{k-1})$$

since $m \le |z-1| + |w-1|$ and $|z-1|, |w-1| \le \delta$. Therefore

$$\left|\sum_{k=2}^{\infty} c_k \Big(z(w-1)^k - w(z-1)^k \Big)\right| \tag{3.41}$$
$$\le |z-w|\,(|z-1| + |w-1|) \sum_{k=2}^{\infty} |c_k| \Big(k\delta^{k-2} + (k-1)\delta^{k-1} \Big).$$

The infinite series at the right-hand side of Equation (3.41) has a finite value and does not depend on z, w when $|z-1|, |w-1| \le \delta$. From this and Equation (3.39) we get Equation (3.36) for such z, w.

Step II. We have

$$\frac{Y^c(z)}{z^g} = \exp\big(g\,h(z)\big),$$

with $h(z)$ given by Equation (3.29). We define $z = z(x)$ for real x of small modulus by setting

$$h(z(x)) = h(z_{sp}) - \frac{1}{2}x^2 h''(z_{sp}).$$

In Section 3 of [101], it is shown that there is a $\delta > 0$, independent of $c \geq 1$, such that $z(x)$ is given by a power series

$$z(x) = z_{sp} + ix + \sum_{k=2}^{\infty} c_k (ix)^k, \qquad |x| \leq \delta,$$

with real c_k and $i^2 = -1$. We thus have $z'(x) = i + O(|x|)$, which shows that the curve $(x, z(x))$ is tangent to the circle $|z| = z_{sp}$ at $z = z_{sp}$.

Substituting $z = z(x)$, $-\delta \leq x \leq \delta$, in Equation (3.28) produces an approximation of $I_c(w)$ with exponentially small error. Note that $\mathrm{d}z = z'(x)\mathrm{d}x = \big(i + O(|x|)\big)\mathrm{d}x$. When we use, furthermore, Equation (3.38), we get

$$I_c(w) = \frac{1}{2\pi} \int_{-\delta}^{\delta} \frac{w-1}{(z(x)-1)(w-z(x))} \ln\left(1 - \frac{Y^c(z(x))}{(z(x))^g}\right)(1+O)\mathrm{d}x, \tag{3.42}$$

where O abbreviates $O\big(|x| + |z(x)-1| + |w-1|\big)$. Note that $\frac{Y^c(z(x))}{(z(x))^g}$ is given by Equation (3.31) in Gaussian form.

Step III. We have that z_{sp} is the solution of $h'(z) = 0$ with z larger than, but close to, 1. From

$$0 = h'(z_{sp}) = a_1 + a_2(z_{sp}-1) + \frac{1}{2}a_3(z_{sp}-1)^2 + \ldots,$$

where $a_i = h^{(i)}(1)$, we get

$$\begin{aligned} z_{sp} - 1 &= \frac{-a_1/a_2}{1 + a_3(z_{sp}-1)/2a_2 + \ldots} - \frac{a_1}{a_2} + \frac{a_1 a_3}{2a_2^2}(z_{sp}-1) + \ldots \\ &= -\frac{a_1}{a_2} - \frac{a_3}{2a_2}\left(\frac{a_1}{a_2}\right)^2 + \ldots. \end{aligned} \tag{3.43}$$

Next, from Equation (3.43), using $h(1) = 0$, we get

$$\begin{aligned} h(z_{sp}) &= a_1(z_{sp}-1) + \frac{1}{2}a_2(z_{sp}-1)^2 + \frac{1}{6}a_3(z_{sp}-1)^3 + \ldots \\ &= -\frac{a_1^2}{2a_2} - \frac{a_3 a_1^3}{6a_2^3} - \ldots. \end{aligned}$$

We express $a_i = h^{(i)}(1)$, $i = 1, 2, 3$, in terms of μ, σ, β, and c. We have

$$h'(z) = -\frac{1}{z} + \frac{c}{g}\frac{Y'(z)}{Y(z)},$$

and so, from $g = c\mu + \beta\sigma\sqrt{c}$, $Y(1) = 1$, and $Y'(1) = \mu$ we have

$$a_1 = h'(1) = \frac{c\mu}{g} - 1 = -\beta\sigma\frac{\sqrt{c}}{g} = \frac{-\beta\sigma}{\mu\sqrt{c}}\left(1 + O\Big(\frac{1}{\sqrt{c}}\Big)\right).$$

Next, we have that

$$h''(z) = \frac{1}{z^2} + \frac{c}{g}\frac{Y''(z)Y(z) - (Y'(z))^2}{(Y(z))^2},$$

and so

$$\begin{aligned} a_2 = h''(1) &= 1 + \frac{1}{\mu}\left(1 + O\Big(\frac{1}{\sqrt{c}}\Big)\right)\Big(Y''(1) - \big(Y'(1)\big)^2\Big) \\ &= \frac{1}{\mu}\big(Y''(1) + \mu - \mu^2\big) + O\Big(\frac{1}{\sqrt{c}}\Big) = \frac{\sigma^2}{\mu} + O\Big(\frac{1}{\sqrt{c}}\Big). \end{aligned}$$

In a similar fashion, $a_3 = h'''(1)$ can be computed as a quantity that remains bounded as $c \to \infty$.

We then find, subsequently,

$$z_{sp} - 1 = \frac{\beta}{\sigma\sqrt{c}}\left(1 + O\Big(\frac{1}{\sqrt{c}}\Big)\right), \tag{3.44}$$

$$h(z_{sp}) = \frac{-\beta^2}{2c\mu}\left(1 + O\Big(\frac{1}{\sqrt{c}}\Big)\right),$$

$$h''(z_{sp}) = h''(1) + O(z_{sp} - 1) = \frac{\sigma^2}{\mu}\left(1 + O\Big(\frac{1}{\sqrt{c}}\Big)\right).$$

It then follows that

$$gh(z_{sp}) = -\frac{1}{2}\beta^2 + O\Big(\frac{1}{\sqrt{c}}\Big), \tag{3.45}$$

$$h''(z_{sp}) = \frac{\sigma^2}{\mu}\left(1 + O\Big(\frac{1}{\sqrt{c}}\Big)\right).$$

For later use in Step IV, we also mention that

$$\frac{\sqrt{gh''(z_{sp})}}{\sigma\sqrt{c}} = 1 + O\Big(\frac{1}{\sqrt{c}}\Big), \tag{3.46}$$

$$(z_{sp} - 1)\sqrt{gh''(z_{sp})} = \beta\left(1 + O\Big(\frac{1}{\sqrt{c}}\Big)\right). \tag{3.47}$$

Note that for $-\delta \le x \le \delta$ we have from Equation (3.45):

$$\begin{aligned}\ln\left(1-\frac{Y^c(z(x))}{(z(x))^g}\right) &= \ln\left(1-\exp\left(gh(z_{sp})-\frac{1}{2}gh''(z_{sp})x^2\right)\right) \\ &= \ln\left(1-\mathrm{e}^{-\frac{1}{2}\beta^2-\frac{1}{2}v^2}\right)\left(1+O\Big(\frac{1}{\sqrt{c}}\Big)\right),\end{aligned} \tag{3.48}$$

where we have set $v = x\sqrt{gh''(z_{sp})}$.

Step IV. Let t be in a bounded set with $\mathrm{Re}(t) \le \frac{1}{2}\beta$. Then

$$w-1 = \exp(t/(\sigma\sqrt{c}))-1 = \frac{t}{\sigma\sqrt{c}}\left(1+O\Big(\frac{1}{\sqrt{c}}\Big)\right).$$

With $z = z(x) = z_{sp} + ix + O(x^2)$, we have

$$\frac{w-1}{(z-1)(w-z)} = \frac{t/(\sigma\sqrt{c})}{\big(z_{sp}-1+ix\big)\big(t/(\sigma\sqrt{c})-(z_{sp}-1)-ix\big)}\left(1+O\Big(|x|+\frac{1}{\sqrt{c}}\Big)\right). \tag{3.49}$$

The factor $1+O\Big(|x|+\frac{1}{\sqrt{c}}\Big)$ follows from Equation (3.44) and $\mathrm{Re}(t) \le \frac{1}{2}\beta$, so that

$$z_{sp}-1-\mathrm{Re}\left(\frac{t}{\sigma\sqrt{c}}\right) \ge \frac{\beta}{2\sigma\sqrt{c}}\left(1+O\Big(\frac{1}{\sqrt{c}}\Big)\right).$$

We next substitute $v = x\sqrt{gh''(z_{sp})}$. Writing

$$\begin{aligned}\gamma &= \frac{\sqrt{gh''(z_{sp})}}{\sigma\sqrt{c}} = 1+O\Big(\frac{1}{\sqrt{c}}\Big), \\ \eta &= (z_{sp}-1)\sqrt{gh''(z_{sp})} = \beta\left(1+O\Big(\frac{1}{\sqrt{c}}\Big)\right),\end{aligned}$$

where we use Equations (3.46) and (3.47), we have uniformly in $x \in \mathbb{R}$:

$$\begin{aligned}&\frac{t/(\sigma\sqrt{c})\,\mathrm{d}x}{\big(z_{sp}-1+ix\big)\big(t/(\sigma\sqrt{c})-(z_{sp}-1)-ix\big)} \\ &= \frac{\gamma t\,\mathrm{d}v}{(\eta+iv)(\gamma t-\eta-iv)} = \frac{t\,\mathrm{d}v}{(\beta+iv)(t-\beta-iv)}\left(1+O\Big(\frac{1}{\sqrt{c}}\Big)\right).\end{aligned} \tag{3.50}$$

Step V. By Equations (3.42), (3.48), (3.49), and (3.50), we have, with $w = \exp(t/(\sigma\sqrt{c}))$,

$$I_c(w) = \frac{1}{2\pi}\int_{-\Delta}^{\Delta}\left[\frac{t}{(\beta+i\nu)(t-\beta-i\nu)}\ln\left(1-\mathrm{e}^{-\frac{1}{2}\beta^2-\frac{1}{2}\nu^2}\right)\left(1+O\Big(\frac{1+|\nu|}{\sqrt{c}}\Big)\right)\right]\mathrm{d}\nu,$$

where $\Delta = \delta\sqrt{gh''(z_{sp})}$. For this it has been used that

$$|x| = \frac{|\nu|}{\sqrt{gh''(z_{sp})}} = O\left(\frac{|\nu|}{\sqrt{c}}\right),$$
$$|z(x)-1| \le |z_{sp}-1| + O(|x|) = O\left(\frac{1+|\nu|}{\sqrt{c}}\right).$$

Finally, since

$$\ln\left(1-\mathrm{e}^{-\frac{1}{2}\beta^2-\frac{1}{2}\nu^2}\right) = O\left(\mathrm{e}^{-\frac{1}{2}\nu^2}\right)$$

when $\nu\to\infty$, while $\Delta = \delta\sigma\sqrt{c}\left(1+O\left(\frac{1}{\sqrt{c}}\right)\right)\to\infty$ like $\sqrt{c}$, we get that

$$I_c(w) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{t}{(\beta+i\nu)(t-\beta-i\nu)}\ln\left(1-\mathrm{e}^{-\frac{1}{2}\beta^2-\frac{1}{2}\nu^2}\right)\mathrm{d}\nu\left(1+O\Big(\frac{1}{\sqrt{c}}\Big)\right).$$

That is, $I_c(w) = J(t)\left(1+O\left(\frac{1}{\sqrt{c}}\right)\right)$, and this holds uniformly in t in any bounded set with $\mathrm{Re}(t) \le \frac{1}{2}\beta$, finishing the proof of Equation (3.27).

Turning to Equation (3.5) in Theorem 3.1, we have for the MGF's F_c and F in Equation (3.27)

$$F_c(t) = \sum_{k=0}^{\infty}\frac{m_k(c)}{k!}t^k,$$
$$F(t) = \sum_{k=0}^{\infty}\frac{m_k}{k!}t^k,$$

where $m_k(c)$ and m_k are the k^{th} moment of $X_g/(\sigma\sqrt{c})$ and M_β. By Cauchy's integral formula for the k^{th} derivative at 0 of an analytic function, we have

$$\frac{m_k(c)}{k!} = \frac{1}{2\pi i}\oint_{|t|=a}\frac{F_c(t)}{t^{k+1}}\mathrm{d}t,$$

where we take $a > 0$ such that the disk $|t| \le a$ is contained in the set of t's where the convergence in Equation (3.27) is uniform. Since $F_c(t) = F(t)\left(1 + O\left(\frac{1}{\sqrt{c}}\right)\right)$ uniformly on $|t| = a$, this yields $m_k(c) = m_k\left(1 + O\left(\frac{1}{\sqrt{c}}\right)\right)$, proving Equation (3.5) in Theorem 3.1.

To prove Equation (3.4) in Theorem 3.1, we must argue differently. Letting $t \to -\infty$ in Equation (3.26), we have

$$\mathbb{P}(M_\beta = 0) = \exp\left(\frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{1}{\beta + i\nu} \ln\left(1 - \mathrm{e}^{-\frac{1}{2}\beta^2 - \frac{1}{2}\nu^2}\right) \mathrm{d}\nu\right).$$

Also, setting $w = 0$ in Equation (3.25), we have that the front factor in the integral in Equation (2.2) is given by

$$\frac{Y'(z)z - Y(z)}{z - Y(z)} \cdot \frac{-1}{z} = \frac{1}{z-1}\Big(1 + O\big(|z-1|\big)\Big),$$

where $Y(0) > 0$ and Equation (3.35) have been used. We are now in a completely similar, and indeed even simpler, situation as before:

$$\mathbb{P}\left(\frac{1}{\sigma\sqrt{c}} X_g = 0\right) = \exp\left(\frac{1}{2\pi i}\oint_{|z|=1+\epsilon} \frac{1 + O(|z-1|)}{z-1} \ln\left(1 - \frac{Y^c(z)}{z^g}\right) \mathrm{d}z\right).$$

The combined effect on the front factor of the two substitutions $z = z(x)$ and $\nu = x\sqrt{g h''(z_{sp})}$ amounts to

$$\frac{1}{i}\frac{\mathrm{d}z}{z-1} = \frac{\mathrm{d}\nu}{\beta + i\nu}\left(1 + O\left(\frac{1}{\sqrt{c}}\right)\right)$$

and this yields $\mathbb{P}\left(X_g/(\sigma\sqrt{c}) = 0\right) = \mathbb{P}(M_\beta = 0)\left(1 + O\left(\frac{1}{\sqrt{c}}\right)\right)$.

Remark 3.2 *We can allow the green time g to be random as we for example do for the G_i in Remark 3.1. The randomness in G_i as introduced in Remark 3.1 has a minor impact on the proof of Theorem 3.1. We need to modify Equation* (3.31) *slightly and multiply the left-hand side of Equation* (3.31) *with $1/\big(p + (1-p)z(x)\big)$ with p as in Remark 3.1. Observe that*

$$1/\big(p + (1-p)z(x)\big) = 1 + O(z(x) - 1) \tag{3.51}$$

uniformly in p for $0 \le p \le 1$. As the right-hand side of Equation (3.31) *is smaller than* 1*, see Equation* (3.32)*, we may take the factor in Equation* (3.51) *out of the logarithm in* (3.33)*. In this way, the proof of Theorem 3.1 still works with the only further modification that Equation* (3.42) *gets an additional $O(z(x) - 1)$ term from Equation* (3.51).

3.6 Conclusion

The main technical novelty in this chapter concerns establishing heavy-traffic limits for the single-lane FCTL queue, in particular Theorem 3.1. These heavy-traffic limits follow from combining a suitable large-cycle regime (3.2) with the transform method for establishing convergence in distribution of the stationary overflow to a nondegenerate limit. We are able to use this transform method thanks to Theorem 2.1, providing an alternative expression for the PGF of the overflow queue than the one established in the existing literature. The proof that exploits this transform method is presented in Section 3.5 and is interesting in its own right. The key technical novelty, the asymptotic expansion of the complex contour integral, is likely to be of broader interest and not limited to the FCTL queue. Examples where this proof method applies include the bulk-service queue and extensions of the FCTL queue considered respectively in [146] and in Chapter 2.

The limiting heavy-traffic behavior is governed by a reflected Gaussian random walk with negative drift, a well-studied stochastic process. This gives heavy-traffic approximations that reduce the complexity of the (pre-limit) expressions for the mean overflow queue in the FCTL queue considerably. These limiting results enable us to formulate easy-to-calculate approximations and allow us to solve capacity allocation problems in the form of optimization problems that generate (close-to-optimal) green times. This adds to the literature of capacity allocation problems [112, 219] and asymptotic dimensioning of queueing systems [25, 208].

In some practical situations, it might be beneficial to have non-static signaling strategies, such as vehicle-actuated strategies. Generalizations of the results to non-deterministic cycle times and green times are possible. Under appropriate adaptations of Equation (3.2) and certain restrictions, e.g. on the standard deviation of the red and green periods, similar heavy-traffic results can be established as the ones derived in this chapter, see also Remark 3.2. Another example is vehicle-actuated signaling, where the green times depend on the queue lengths. An example would be that, instead of a fixed green time, we introduce a maximum green time and switch to the next queue as soon as either the queue empties or the maximum green time is reached. The corresponding model is multidimensional (as opposed to the one-dimensional FCTL queue) and a theoretical analysis similar to the one conducted here is therefore not possible. Nevertheless, we show, by means of simulation, that the same scaling as in rule (3.2) leads to similar asymptotic results for several vehicle-actuated strategies in the next chapter, Chapter 4.

Appendix

3.A Remaining proofs

We now provide the proofs of Proposition 3.2 and Theorem 3.4 in Subsections 3.A.1 and 3.A.2, respectively.

3.A.1 Proof of the heavy-traffic approximation for the mean queue length

We start the proof with an expression for $\mathbb{E}[X_g]$. Equation (2.11) reads

$$\mathbb{E}[X_g] = \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} \frac{Y(z) - zY'(1)}{Y(z) - z} \frac{\left(z^g - Y(z)^c\right)'}{z^g - Y(z)^c} \mathrm{d}z,$$

for some $\epsilon > 0$. We define, as before,

$$h(z) = -\ln z + \frac{c}{g} \ln Y(z).$$

Then we are able to derive (following the same steps as in the proof of Lemma 1 in [101])

$$\begin{aligned}
\mathbb{E}[X_g] &= \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} \frac{Y(z) - zY'(1)}{Y(z) - z} \frac{gz^{g-1} - cY(z)^{c-1}Y'(z)}{z^g - Y(z)^c} \mathrm{d}z \\
&= \frac{1}{2\pi i} \oint_{|z|=1+\epsilon} \left[\frac{Y(z) - zY'(1)}{Y(z) - z} \left(\frac{g}{z} - \frac{g}{z} \left(\frac{c}{g} \frac{zY'(z)}{Y(z)} - 1 \right) \frac{z^{-g} Y(z)^c}{1 - z^{-g} Y(z)^c} \right) \right] \mathrm{d}z
\end{aligned}$$

$$= \frac{g}{2\pi i} \oint_{|z|=1+\epsilon} h'(z) \frac{Y(z) - zY'(1)}{z - Y(z)} \frac{\exp(gh(z))}{1 - \exp(gh(z))} \mathrm{d}z,$$

where in the last step we use that

$$h'(z) = \frac{c}{g} \frac{Y'(z)}{Y(z)} - \frac{1}{z},$$
$$\oint_{|z|=1+\epsilon} \frac{Y(z) - zY'(1)}{Y(z) - z} \frac{g}{z} \mathrm{d}z = 0.$$

We let z_{sp} denote the unique minimum of $h(z)$ with $z \geq 1$ and we let

$$z(x) = z_{sp} + ix + \sum_{k=2}^{\infty} c_k (ix)^k$$

solve the equation

$$h(z(x)) = h(z_{sp}) - \frac{1}{2} x^2 h''(z_{sp}) =: q(x).$$

Then, following the same steps as are taken in Section 3 of [101], we get that with exponentially small error

$$\mathbb{E}[X_g] = -\frac{gh''(z_{sp})}{2\pi i} \int_{-1/2\delta}^{1/2\delta} x \frac{Y(z(x)) - z(x)Y'(1)}{z(x) - Y(z(x))} \frac{\exp(gq(x))}{1 - \exp(gq(x))} \mathrm{d}x \tag{3.52}$$

for some $\delta > 0$.

Proceeding as in the proof of Theorem 3 of [101], we obtain, since $z(-x) = \overline{z(x)}$ for real x, where $\overline{a}$ denotes the complex conjugate of a, that

$$x \frac{Y(z(x)) - z(x)Y'(1)}{z(x) - Y(z(x))} - x \frac{Y(z(-x)) - z(-x)Y'(1)}{z(-x) - Y(z(-x))} = \frac{-2ix^2 \left(1 + O(z_{sp} - 1) + x^2\right)}{(z_{sp} - 1)^2 + x^2 - 2c_2(z_{sp} - 1)x^2},$$

for $|x| \leq 1/(2\delta)$ and where $c_2 \in \mathbb{R}$. This implies that we get, with exponentially small error, using the previous result together with Equation (3.52) and extending the integration range to $(-\infty, \infty)$ while using symmetry of $q(x)$, that

$$\mathbb{E}[X_g] = \frac{gh''(z_{sp})}{\pi} \int_0^{\infty} \left[\frac{x^2 \left(1 + O(z_{sp} - 1) + x^2\right)}{(z_{sp} - 1)^2 + x^2 - 2c_2(z_{sp} - 1)x^2} \frac{\exp(gq(x))}{1 - \exp(gq(x))} \right] \mathrm{d}x,$$

so we are now exactly in the same situation as that of Sections 4 and 5.1 of [101]. Here it should be noted that the FCTL relation $g = c\mu + \beta\sigma\sqrt{c}$, see Equation (3.2), can be written in the bulk-service queue form of [101], $c/g = (1-\gamma/\sqrt{g})/\mu$ with

$$\gamma = \frac{\beta\sigma}{\sqrt{\mu}}\left(1+\frac{\beta\sigma}{\mu\sqrt{c}}\right)^{-1/2}. \tag{3.53}$$

Hence, letting

$$b_0^2 = b(\beta)^2 = \frac{\gamma^2\mu}{2\sigma^2} = \frac{1}{2}\beta^2\left(1+\frac{\beta\sigma}{\mu\sqrt{c}}\right)^{-1}, \tag{3.54}$$

see Equation (4.12) of [101], we get with an absolute error of order $1/\sqrt{c}$,

$$\begin{aligned}\mathbb{E}[X_g] = &\frac{\sigma}{\pi}\sqrt{\frac{2g}{\mu}}G_0(b_0)+\\ &\frac{\sigma}{\pi}\sqrt{\frac{2}{\mu}}\left((C_1+C_3)G_0(b_0)-(C_2+b_0^2C_3)G_3(b_0)+C_4G_4(b_0)\right),\end{aligned}$$

according to Equation (5.14) of [101], with

$$\begin{aligned}G_3(b) &= \int_0^\infty \frac{t^2}{(b^2+t^2)^2}\frac{\mathrm{e}^{-b^2-t^2}}{1-\mathrm{e}^{-b^2-t^2}}\mathrm{d}t\\ G_4(b) &= \int_0^\infty \frac{t^2}{b^2+t^2}\frac{\mathrm{e}^{-b^2-t^2}}{(1-\mathrm{e}^{-b^2-t^2})^2}\mathrm{d}t.\end{aligned}$$

We proceed by computing the C_i explicitly. From [101], Equations (5.2-5.5), (5.8), and (5.9), we get

$$\begin{aligned}C_1 &= -\frac{\gamma(\sigma^2-\mu)}{2\sigma^2},\\ C_2 &= \frac{\gamma(\sigma^2-\mu)}{\sigma^2}b_0^2.\end{aligned}$$

Furthermore, from [101], Equations (5.2-3), (5.5-6), and (5.10), we get

$$C_3 = -\frac{1}{3}\gamma a\frac{\mu^2}{\sigma^4}, \tag{3.55}$$

while from [101], Equations (5.2-3), (5.7), and (5.11), we get

$$C_4 = -\gamma \frac{\sigma^2 - \mu}{\sigma^2} b_0^2 + \frac{1}{3}\gamma a \frac{\mu^2}{\sigma^4} b_0^2. \tag{3.56}$$

In Equations (3.55) and (3.56), a is given by, see [101], Equation (5.3),

$$\begin{aligned} a &= -2 + \frac{Y'''(1)}{Y'(1)} - 3Y''(1) + 2\left(Y'(1)\right)^2 \\ &= \frac{1}{\mu}\left(\mu_3 - \mu^3 - 3(1+\mu)\sigma^2\right), \end{aligned}$$

where $\mu_3 = \mathbb{E}[Y^3]$, as in Equation (3.10).

It follows that

$$\begin{aligned} C_1 + C_3 &= \frac{1}{2b_0^2} C_4, \\ C_2 + b_0^2 C_3 &= -C_4. \end{aligned}$$

When we also use Equations (5.17) and (4.27) from [101], we get

$$\begin{aligned} G_3(b_0) + G_4(b_0) &= \frac{1}{2b_0^2} G_2(b_0), \\ G_0(b_0) + G_2(b_0) &= G_1(b_0), \end{aligned}$$

with

$$G_2(b) = \int_0^\infty \frac{b^2}{b^2 + t^2} \frac{\mathrm{e}^{-b^2 - t^2}}{1 - \mathrm{e}^{-b^2 - t^2}} \mathrm{d}t,$$

and find

$$(C_1 + C_3) G_0(b_0) - (C_2 + b_0^2 C_3) G_3(b_0) + C_4 G_4(b_0) = (C_1 + C_3) G_1(b_0).$$

Therefore we get, with an absolute error of order $1/\sqrt{c}$,

$$\mathbb{E}[X_g] = \frac{\sigma}{\pi}\sqrt{\frac{2g}{\mu}} G_0(b_0) + \frac{\sigma}{\pi}\sqrt{\frac{2}{\mu}} (C_1 + C_3) G_1(b_0).$$

Finally, we have from $g = c\mu + \beta\sigma\sqrt{c}$ and Equations (3.55) and (3.56), that

$$\frac{\sigma}{\pi}\sqrt{\frac{2g}{\mu}} = \frac{\sqrt{2}}{\pi}\sigma\sqrt{c}\left(1 + \frac{\beta\sigma}{\mu\sqrt{c}}\right)^{1/2}$$

and

$$\frac{\sigma}{\pi}\sqrt{\frac{2}{\mu}}(C_1+C_3) = \frac{\sqrt{2}}{\pi}\frac{\gamma\sigma}{2\sqrt{\mu}}\left(-\frac{\sigma^2-\mu}{\sigma^2}+\frac{1}{3}a\frac{\mu^2}{\sigma^4}\right)$$
$$= \frac{\sigma^2 b(\beta)}{\pi\mu}\left(\frac{\mu}{\sigma^2}+\frac{1}{3}a\frac{\mu^2}{\sigma^4}-1\right), \tag{3.57}$$

where in Equation (3.57) also Equations (3.53) and (3.54) have been used. Therefore, with θ as given in Equation (3.11), we get

$$\mathbb{E}[X_g] = \frac{\sqrt{2}}{\pi}\sigma\left(1+\frac{\beta\sigma}{\mu\sqrt{c}}\right)^{1/2} G_0(b(\beta)) + \frac{\sqrt{2}}{\pi}\theta b(\beta) G_1(b(\beta)) + O\left(\frac{1}{\sqrt{c}}\right).$$

The expression in Equation (3.9) is then obtained by noting that

$$\left(1+\frac{\beta\sigma}{\mu\sqrt{c}}\right)^{1/2} = 1+\frac{\beta\sigma}{2\mu\sqrt{c}}+O\left(\frac{1}{c}\right),$$
$$b(\beta) = \frac{\beta}{\sqrt{2}}+O\left(\frac{1}{\sqrt{c}}\right),$$

finishing the proof of Proposition 3.2.

3.A.2 Proof of optimal green-time allocation using Equation (3.9)

We use the Lagrange multiplier technique to prove Theorem 3.4. To start, we differentiate Equation (3.17)

$$\frac{\partial}{\partial\beta_j}\sum_{i=1}^{n}\left(\frac{\sqrt{2}}{\pi}\left(\sigma_i\sqrt{c}+\frac{1}{2}\beta_i\frac{\sigma_i^2}{\mu_i}\right)G_0\left(b_i(\beta_i)\right)+\frac{\theta_i\beta_i}{\pi}G_1\left(\frac{\beta_i}{\sqrt{2}}\right)\right) =$$
$$\frac{1}{\pi\sqrt{2}}\frac{\sigma_j^2}{\mu_j}G_0(b_j(\beta_j))+\frac{\sqrt{2}}{\pi}\left(\sigma_j\sqrt{c}+\frac{\beta_j\sigma_j^2}{2\mu_j}\right)b_j'(\beta_j)G_0'(b_j(\beta_j))+$$
$$\frac{\theta_j}{\pi}G_1\left(\frac{\beta_j}{\sqrt{2}}\right)+\frac{\theta_j\beta_j}{\pi\sqrt{2}}G_1'\left(\frac{\beta_j}{\sqrt{2}}\right) =$$
$$\frac{\sigma_j\sqrt{c}}{\pi}G_0'\left(\frac{\beta_j}{\sqrt{2}}\right)+\frac{1}{\pi}\left\{\frac{\sigma_j^2}{\sqrt{2}\mu_j}G_0\left(\frac{\beta_j}{\sqrt{2}}\right)-\frac{\beta_j\sigma_j^2}{2\mu_j}G_0'\left(\frac{\beta_j}{\sqrt{2}}\right)-\right.$$
$$\left.\frac{\beta_j^2\sigma_j^2}{2\sqrt{2}\mu_j}G_0''\left(\frac{\beta_j}{\sqrt{2}}\right)+\theta_j G_1\left(\frac{\beta_j}{\sqrt{2}}\right)+\frac{\theta_j\beta_j}{\sqrt{2}}G_1'\left(\frac{\beta_j}{\sqrt{2}}\right)\right\}+O\left(\frac{1}{\sqrt{c}}\right),$$

where we have used/approximated that

$$
\begin{aligned}
b(\beta_j) &= \frac{\beta_j}{\sqrt{2}} - \frac{\beta_j^2\sigma_j}{2\sqrt{2}\mu_j\sqrt{c}} + O\left(\frac{1}{c}\right),\\
G_0\left(b_j(\beta_j)\right) &= G_0\left(\frac{\beta_j}{\sqrt{2}}\right) + O\left(\frac{1}{\sqrt{c}}\right),\\
\left(\sigma_j\sqrt{c} + \frac{\beta_j\sigma_j^2}{2\mu_j}\right) b_j'(\beta_j) &= \frac{\sigma_j\sqrt{c}}{\sqrt{2}} - \frac{\beta_j\sigma_j^2}{2\sqrt{2}\mu_j} + O\left(\frac{1}{\sqrt{c}}\right),\\
G_0'\left(b_j(\beta_j)\right) &= G_0'\left(\frac{\beta_j}{\sqrt{2}}\right) - \frac{\beta_j^2\sigma_j}{2\sqrt{2}\mu_j\sqrt{c}} G_0''\left(\frac{\beta_j}{\sqrt{2}}\right) + O\left(\frac{1}{c}\right).
\end{aligned}
$$

So, introducing a Lagrange multiplier $\lambda_1 \in \mathbb{R}$ and ignoring the O-terms, we need to solve

$$
\begin{aligned}
\lambda_1\sigma_j\sqrt{c} = \frac{\sigma_j\sqrt{c}}{\pi} G_0'\left(\frac{\beta_j}{\sqrt{2}}\right) + \frac{1}{\pi}\Bigg\{ &\frac{\sigma_j^2}{\sqrt{2}\mu_j} G_0\left(\frac{\beta_j}{\sqrt{2}}\right) - \frac{\beta_j\sigma_j^2}{2\mu_j} G_0'\left(\frac{\beta_j}{\sqrt{2}}\right)\\
&- \frac{\beta_j^2\sigma_j^2}{2\sqrt{2}\mu_j} G_0''\left(\frac{\beta_j}{\sqrt{2}}\right) + \theta_j G_1\left(\frac{\beta_j}{\sqrt{2}}\right) + \frac{\theta_j\beta_j}{\sqrt{2}} G_1'\left(\frac{\beta_j}{\sqrt{2}}\right)\Bigg\},
\end{aligned}
$$

for $j = 1,\dots,n$. The second term on the right-hand side of the former equation is $O(1)$ and it is fair to expect that the optimal β_j in Theorem 3.4 are close to the optimal solution in Theorem 3.3 in the sense that $\beta_j = \beta_* + O(1/\sqrt{c})$. Therefore, we approximate K_j as in Equation (3.19). After rewriting, we then get

$$
\frac{1}{\pi} G_0'\left(\frac{\beta_j}{\sqrt{2}}\right) = \lambda_1 - \frac{K_j}{\pi\sigma_j\sqrt{c}}.
$$

We develop, using Equation (3.16),

$$
\begin{aligned}
\frac{1}{\pi} G_0'\left(\frac{\beta_j}{\sqrt{2}}\right) &= \frac{1}{\pi} G_0'\left(\frac{\beta_*}{\sqrt{2}}\right) + \frac{1}{\pi\sqrt{2}}(\beta_j - \beta_*) G_0''\left(\frac{\beta_*}{\sqrt{2}}\right) + O\left(\frac{1}{c}\right)\\
&= \lambda_0 + \frac{1}{\pi\sqrt{2}}(\beta_j - \beta_*) G_0''\left(\frac{\beta_*}{\sqrt{2}}\right) + O\left(\frac{1}{c}\right).
\end{aligned}
$$

Combining the last two results, we get

$$
\frac{1}{\pi\sqrt{2}}(\beta_j - \beta_*) G_0''\left(\frac{\beta_*}{\sqrt{2}}\right) = \lambda_1 - \lambda_0 - \frac{K_j}{\pi\sigma_j\sqrt{c}} + O\left(\frac{1}{c}\right).
$$

Deleting the $O(1/c)$ term, we find that

$$\beta_j = \beta_* + \frac{\lambda_1 - \lambda_0 - \frac{K_j}{\pi\sigma_j\sqrt{c}}}{\frac{1}{\pi\sqrt{2}} G_0''\left(\frac{\beta_*}{\sqrt{2}}\right)}.$$

Using the equality constraint, we readily see that the following should hold

$$\sum_{j=1}^{n} \sigma_j \left(\lambda_1 - \lambda_0 - \frac{K_j}{\pi\sigma_j\sqrt{c}}\right) = 0,$$

implying that

$$\lambda_1 - \lambda_0 = \frac{1}{\pi\sqrt{c}} \frac{\sum_{j=1}^{n} K_j}{\sum_{j=1}^{n} \sigma_j}.$$

We thus obtain

$$\beta_i = \beta_* + \sqrt{\frac{2}{c}} \frac{1}{G_0''\left(\frac{\beta_*}{\sqrt{2}}\right)} \left(\frac{\sum_{j=1}^{n} K_j}{\sum_{j=1}^{n} \sigma_j} - \frac{K_i}{\sigma_i}\right),$$

concluding the proof.

Chapter 4

Heavy-traffic scaling of vehicle-actuated traffic lights

4.1 Introduction

The heavy-traffic results in the previous chapter point towards several generalizations. There is a large set of models that allow for a similar heavy-traffic scaling as the green-time allocation rule for the FCTL queue, see Equation (3.2). Heavy-traffic results for the models contained in the general set of queueing models described in Theorem 2.2 can be derived from the results in Chapter 3 and Theorem 2.2. However, in this chapter we look beyond this set of models and consider vehicle-actuated access control of intersections.

Control mechanisms such as semi-actuated or fully vehicle-actuated control are able to adapt to (time-)varying circumstances. Commonly, as a control parameter, a maximum length of the cycle is introduced and hence each lane receives a limited amount of time for vehicle crossing, reducing the probability of excessive waiting times for e.g. lanes on which only few vehicles drive. However, the queueing models for such traffic-light settings are not well understood even in the most basic case of an isolated intersection. Indeed, an example of the type of queueing models that we are dealing with, are polling models with a k-limited type of service discipline which seem mathematically intractable, see also Subsections 1.2.2 and 1.3.2.

We take an approach to find good settings for vehicle-actuated controlled traffic lights where we gain inspiration from the established heavy-traffic re-

sults for the FCTL queue in Chapter 3. For the FCTL queue, we have e.g. proven that if the capacity and demand are balanced (or scaled) in the right way, there exists an allocation of the access times for each of the lanes such that the probability of facing an empty queue at the end of the green period is strictly positive even when the vehicle-to-capacity ratio approaches 1. The heavy-traffic scaling is based on the Central Limit Theorem (CLT), see e.g. [86, Section 5.10], intuitively meaning that the capacity for each lane should be chosen as the mean amount of "work" arriving during a cycle (time needed for all arriving vehicles to cross the intersection), where a variability hedge based on the square root of the variance of the amount of "work" is added. This approach has been applied in various settings and has proven its merits there, for more details see Section 4.3.

In this chapter, we extend the results for the FCTL queue to a fully actuated setting for traffic lights, where the length of the green period or access period is random instead of being fixed as in the FCTL queue. We use the same scaling as for the FCTL queue in Chapter 3 with appropriate modifications and show that the same set of properties holds for more general settings of the access period. This enables us to gain insight in close-to-optimal settings for adaptive traffic lights. Moreover, it is easy to compute those settings and to explain *why* they are performing well. From a mathematical point of view, those results for k-limited polling models and, more generally, models with multiple queues and multiple servers are very scarce and our results thus aid in the understanding of such complicated systems. The gained insights might also pave the way to obtain similar results for networks of intersections with an adaptive control, although such an extension is not straightforward.

In summary, our main contributions are as follows:

(i) We propose a new way of finding a good length of the access periods at intersections. We extend the results of Chapter 3 to a more general distribution of the length of the access period. Instead of a fixed length of the access period, we allow for an actuated control, thus having randomness and dependencies among different cycles in the access period.

(ii) Our approach enables us to gain insights into close-to-optimal settings for adaptive traffic lights. Instead of the need for e.g. difficult optimization schemes or computationally expensive simulations, we are able to find the close-to-optimal settings on the basis of one-line calculations. Another advantage is that our scheme is easy to explain, which adds to its practical value.

Chapter outline

This chapter is organized as follows. In Section 4.2 we present a detailed description of the model under consideration in this chapter. Section 4.3 is devoted to sketching the theoretical background from which we take our inspiration. In Section 4.4 we present simulation results that provide valuable insights and we wrap up in Section 4.5 with a conclusion.

4.2 Model description

As mentioned before, we focus on isolated intersections, with a certain number of lanes, N, leading towards it, which we number from $i = 1,\dots,N$. We number the phases from $j = 1,\dots,M$ and each phase J_j, representing a subset of all lanes that always receives the same color of the traffic light, satisfies $J_j \subseteq \{1,\dots,N\}$. We assume that each phase consists of one or more non-conflicting traffic flows, so that each lane in the same phase can receive access to the intersection at the same time. We also assume that each lane belongs to at least one phase.

We will model such an intersection as a queueing model in order to apply and extend the machinery developed in Chapter 3. For each lane we divide time into slots of fixed length, as in Chapter 3, but instead of assuming a fixed length for the access period (or green time) for each lane as in the FCTL queue, we assume a vehicle-actuated control for each of the access periods in a cycle. The actuation mechanism is as follows: when all lanes in the current phase do not have any vehicles in the queue anymore, the cycle immediately continues with the next phase, or we switch to the next phase after a fixed maximum time if the queue(s) is (are) not yet dissolved. We thus have a limited type of vehicle-actuated control. We assume that a lane remains empty as soon as a lane gets empty, as is for example also done in [17]. This makes sense as, when the queue is dissolved during the access period, a new queue does not build during the remaining access period. We choose the access times as follows: initially, we start with the maximum length of the cycle, denoted with c, based on which we allocate the access times to each of the lanes in a similar way as we did in the FCTL queue in Chapter 3. Any remaining part of c is assumed to correspond to clearance times which results in a red traffic light for all phases (scaling linearly with c), see also Remark 4.1. For completeness, we mention that we incur the entire red time at the end of the cycle. More details can be found below in Section 4.3.

We introduce some further notation: μ_i is the arrival rate of vehicles at lane

i in a single slot and the standard deviation of the number of arrivals at lane i in each slot is denoted with σ_i. We assume that the numbers of arrivals during any slot for any lane are identically distributed and behave independently of one another. Further, with $g_{i,c}$ we denote the access period allocated to lane i when the cycle length is c and with β_i we denote a positive number, which might depend on the lane i. One could use these β_i as a control parameter (as we did in Chapter 3), with which we are able to steer the performance of the lanes: a higher β_i implies a longer access period.

Remark 4.1 *In most settings, the sum of times during which no vehicles are allowed to cross the intersection, the all-red time, will be fixed in length. However, in our model we assume that the all-red time scales linearly with the total cycle length. We do so to show the type of properties we are after, such as the probability of an empty queue at the end of the green period converging to a value strictly between* 0 *and* 1*. When the red time is fixed, the performance will* improve*, while already having very good performance under our assumptions, as we will show in Section 4.4. There we also briefly comment on the implications when the all-red time is fixed.*

Remark 4.2 *The inspiration for this model is current-day traffic. However, exactly analogous results hold for vehicles that are autonomous. Different examples of such intersection access algorithms are discussed in Chapter 7. The vehicle-actuated strategy described in this chapter might also be applied there.*

4.3 Theoretical background

As indicated before, we apply the ideas developed in Chapter 3. First of all, we require stability of the underlying queueing model that we study, or equivalently, the vehicle-to-capacity ratio should be less than 1. This boils down to

$$\mu_i c < g_{i,c},$$

which intuitively makes sense: the capacity at lane i (the right-hand side) should be higher than the average number of arrivals during a cycle (the left-hand side). This is a necessary and sufficient condition as it also is for the FCTL queue, see e.g. [206]: when heavily loaded, the actuated control will behave similarly to the FCTL queue and therefore the stability condition is the same.

The stochastic process that we consider is as follows: we study a k-limited type of polling model with deterministic service times and switchover times. We

then propose a new way to allocate the service limits at queue i, or the access times to lane i, where we exploit the heavy-traffic scaling result obtained in Chapter 3,

$$g_{i,c} = \mu_i c + \beta_i \sigma_i \sqrt{c}, \tag{4.1}$$

where we note that $g_{i,c}$ might have to be rounded up if $g_{i,c}$ is non-integer (to ensure stability), as $g_{i,c}$ is a *number* of slots. The scaling thus relates to finding the right access time or scale of the capacity for lane i, based on the maximum cycle length c. We can choose $\beta_i > 0$ arbitrarily to steer the performance of the system as we did in Chapter 3: a low β_i means that the vehicle-to-capacity ratio for lane i is close to 1 which e.g. implies a relatively high mean queue length for lane i. If β_i is higher, then we are further away from criticality and the queue length for lane i tends to be smaller. As a last note on the model that we consider, we incur the entire all-red time, or the switchover time, at the end of the cycle.

Even though the scaling rule as in Equation (4.1) is meant for the case when c gets large and when the intersection is close to oversaturation, we stress that also for small c good performance is obtained, see Section 4.4. In this sense, the model considered in this chapter is comparable to the FCTL queue studied in Chapter 3, even though the underlying stochastic processes are fundamentally different: we are studying a multidimensional queueing model in the present chapter rather than the one-dimensional FCTL queue.

The scaling rule as in (4.1) has been applied in many settings and in various guises. It yields very desirable properties in many respects: the limiting process, when c grows large, is generally a well-understood process with good system performance. Examples (in our setting) are that the probability of an empty queue at the end of the access period is strictly between 0 and 1, instead of converging to 0 or 1, and that the mean queue length for each lane (which we measure in *number of vehicles*) at the end of the access period scales with the cycle length c as $\sqrt{c}$. Note that the additional capacity needed, compared to the minimum of $\mu_i c$ required for stability, is only of order $\sqrt{c}$, so it increases very slowly compared to the leading order term. In [197, Chapter 6], a similar type of green-time allocation is proposed for vehicle-actuated traffic lights, but the allocation is based on some optimization function which contains approximations for the mean delay at each lane. Using Lagrange multiplier techniques, the optimal green-time allocation is found and yields a similar scaling rule, as the additional capacity is also of the order $\sqrt{c}$ multiplied with the standard deviation of the number of arrivals per slot, see e.g. [197, Section 6.4.2].

Moreover, the limiting process, when known, usually yields good and easy-to-use approximations, which can then be used for further purposes like optimization of certain performance characteristics. Examples of the latter can be found in e.g. call centers [25], communication systems [186], and traffic engineering, see Chapter 3. In [25], it is shown that an approximation based on the limiting process yields very good results even when "far from the limit" (this relates to small c in actuated-access control): an optimization based on the approximation, yields an optimal allocation for many parameter settings. Moreover, when the parameter settings are not the optimal ones, they are only off by a small amount [25]. In Chapter 3 a similar approach is taken and qualitatively similar results are obtained. Based on these observations, we expect that the scaling that we propose results in similar *optimality* results for the actuated access control, yet we do not study this in-depth. For further background on the scaling rule, we refer to the tutorial and review paper [208].

The scaling in Equation (4.1) is inspired by the CLT, a fundamental tool in probability theory. A sum of random variables (under some conditions on independence and similarity) can be scaled by subtracting the mean and dividing by the standard deviation after which the distribution of the sum can be approximated by a normal distribution. Even though we scale the capacity for each queue (the length of the access period) and the demand (the cars aiming to pass the intersection), we have the same structure as in the CLT. Indeed, the demand is a sum of random variables and the capacity is the appropriate scaling: the mean of the demand is $\mu_i c$ and to ensure stability, we add a small term, namely $\beta_i \sigma_i \sqrt{c}$. Then, after multiplying with a factor $1/(\sigma_i \sqrt{c})$, and letting c tend to infinity, we see that the right approximation is a normal distribution with mean $-\beta_i$ (ensuring stability) and standard deviation 1. We also take this approach in Chapter 3, where this intuition is shown to be the exact outcome of the scaling for the FCTL queue.

Due to the vehicle-actuated control considered in this chapter, we are not able to show analytical results. In contrast with the FCTL queue considered in Chapter 3, we do not have an expression for the PGF of the steady-state overflow queue. We are thus not able to study the model in this chapter analytically. One of the difficulties when considering actuated-access control is that dependencies between queues carry over: when a queue empties early in this cycle, the other queues are likely to be short as well. For this dependence we cannot account in exact computations. From the literature on (k-limited) polling systems, it is known that these types of vehicle-control strategies offer little or no hope on an exact solution, see e.g. [24]. However, our simulation results indicate that our scaling rule is achieving what we desire: the probability of an empty queue at

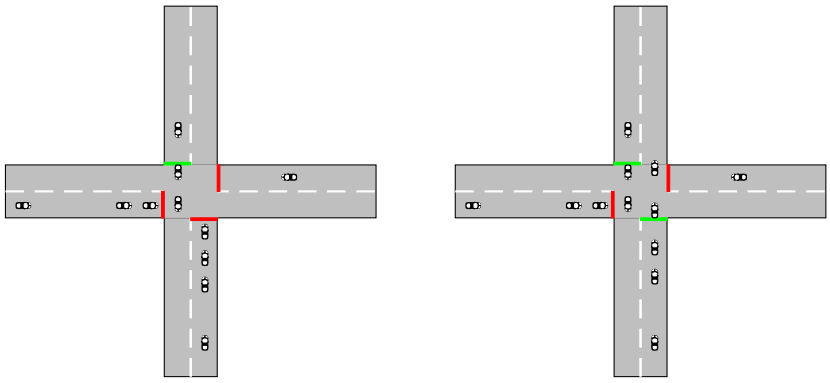
the end of the access period is strictly between 0 and 1 and the queue length just before switching to the next queue is of order $\sqrt{c}$.

4.4 Simulation results

In this section, we show the desirable properties of an actuated traffic control with a scaling rule between demand and capacity as in Equation (4.1). We employ (discrete-event) simulations in order to gain those insights. We consider various settings and discuss the differences and similarities with the results for the FCTL queue obtained in Chapter 3. We also validate (part of) our results with the microscopic traffic simulator SUMO [129], which captures more realistic features such as interactions between vehicles.

4.4.1 Single-lane access control

Example 1a



(a) (b)

Figure 4.1: Graphical representations of (a) the single-lane access control examples considered in Subsection 4.4.1 and (b) the multiple-lane access control examples in Subsection 4.4.2.

This example consists of four lanes, so $N = 4$, where each lane has its own dedicated phase, i.e. cars from only one lane are allowed to cross the intersection, so $J_j = \{j\}$ for $j = 1,\ldots,4$, see Figure 4.1(a) for a graphical represen-

tation. We assume that all vehicles are going straight and that the number of arrivals per time slot is Poisson distributed, with means $\mu_i = i/11$, for lane i with $i = 1,\ldots,4$. In this way, we are able to assign appropriate $g_{i,c}$ for any c sufficiently big (ensuring stability). We choose $\beta_i = 0.1$, so the β_i are the same for each lane, to study the behavior of the system when each lane is receiving the same amount of additional capacity (scaled with the standard deviation of the number of arrivals per cycle). Further, choosing $\beta_i = 0.1$ in this example, turns out to yield clearly non-degenerate behavior as opposed to the cases where $\beta \downarrow 0$ and $\beta \to \infty$. If Equation (4.1) results in non-integer values for $g_{i,c}$, we round them up to the nearest integer.

To investigate the influence of the cycle length c (given in the number of slots unless otherwise specified) on the mean queue length at the end of the access period, $\mathbb{E}[X_{g_{i,c}}]$, and on the probability of having an empty queue at the end of the access period, $\mathbb{P}(X_{g_{i,c}} = 0)$, we perform discrete-event simulations. We perform 8 independent runs with a length of 10^7 cycles in order to reduce simulation variability and obtain accurate simulation results (we also take this number of cycles and runs for the other examples we present unless otherwise specified). The results are shown in Figures 4.2(a) and 4.2(b). Note that the dashed black line (as in the other figures) represents the weighted sum of the vehicle-to-capacity ratios of each individual lane, ρ, with its value on the right axis.

The mean queue length at the end of the access period scales with $\sqrt{c}$, which is shown by the results in Figure 4.2(a). The mean queue length grows as a constant times $\sqrt{c}$, as after dividing by $\sqrt{c}$, the mean queue length seems to converge to a constant (modulo the rounding effect of the $g_{i,c}$ and the simulation uncertainty). The higher the arrival rate on a lane, the higher the limiting constant seems to be, as can be observed in Figure 4.2(a). There is no influence of β_i visible in this example, as the value of β_i is the same for all lanes. The vehicle-to-capacity ratio in Figure 4.2(a) is well above 0.9 for all values of c and often above 0.99, which makes the relative low mean queue lengths at the end of the access period for all lanes quite remarkable.

The predicted behavior of the probability that a queue is empty at the end of the access period is observed in Figure 4.2(b): the probability converges to a value between 0 and 1. This value depends on the β_i only and not on the arrival rate at the lane, as is the case for the FCTL queue under the same type of scaling as in Chapter 3.

The actuated access control mechanism clearly outperforms the fixed control mechanism. This is visible in Figure 4.2(b), as the limiting probability for the fixed control scheme is below 0.2 (this value is in accordance with Theo-

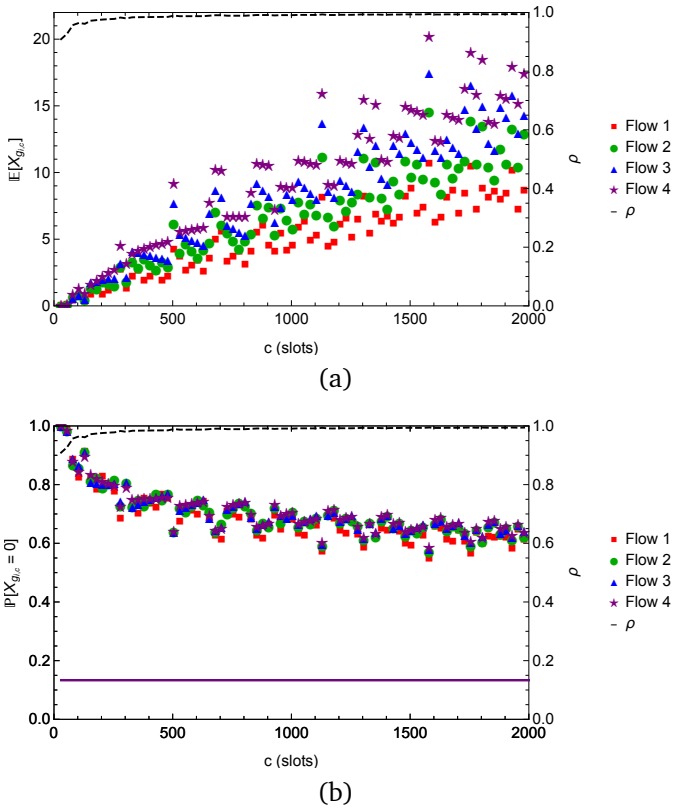


Figure 4.2: The simulated mean queue length at the end of the access period for the four traffic flows in Example 1a in (a); and the simulated probability of an empty queue at the end of the access period for the four traffic flows in Example 1a and the limiting value of that probability for the FCTL queue (solid lines) in (b). In both subfigures we added the vehicle-to-capacity ratio ρ on the right axis (dashed line).

rem 3.1), whereas the simulated probabilities for the actuated access control are well above 0.5. The expected queue length at the end of the green period for the FCTL queue is different for each of the lanes (see e.g. Table 3.1), as is the case for the actuated control setting, which can clearly be seen in Figure 4.2(a). The mean queue length for the FCTL queue would be much higher than the

values in Figure 4.2(a), which is why we did not plot these results.

If we would assume a fixed length for the all-red period and actuated control, Figures 4.2(a) and 4.2(b) would change considerably. The mean queue length at the end of the access period converges to 0 and the probability of an empty queue at that same moment converges to 1. This shows that, as long as the vehicle-to-capacity ratio is below 1 and under some independence assumptions on the arrivals, there exist settings for vehicle-actuated traffic control that are capable of dealing with all traffic efficiently and that have empty queues at the end of the green period.

Example 1b

In Figures 4.3(a) and 4.3(b) we have adapted the values of μ_i and β_i. We choose $\mu_i = 5/22$ and $\beta_i = i/10$ for $i = 1,...,4$. Qualitatively, we observe the same behavior, yet some interesting differences are also present. The mean queue length at the end of the access period depends on the value of β_i, which makes sense: a higher β_i results in a longer maximal access period, and thus in a smaller queue length.

Surprisingly, the probability of an empty queue at the end of the access period seems to converge to the same value for each of the lanes, even though the values of the β_i differ, see Figure 4.3(b). In the fixed access control setting there is a differentiation, see e.g. Tables 3.1 and 3.2 and the limiting probabilities for the fixed-control case in Figure 4.3(b). It might be that an empty queue implies an early switch to the next phase. This is then more likely to result in empty queues at the end of the access period in that phase, because the vehicles had a shorter time to accumulate on these lanes. This effect seems to strengthen over cycles and to cause the probability to be the same for each of the lanes.

SUMO Example

As a proof of concept, we also present an example performed in the microscopic traffic simulator SUMO employing the so-called vehicle-actuated control mechanism of SUMO based on time gaps. Our purpose is to show that also in this simulator, which is generally considered to be excellent in capturing real-world traffic dynamics, we are able to define an actuated control with the desirable properties as are obtained in the other examples. For simplicity, we assume that we have two lanes, so $N = 2$. As arrival distribution we choose a Bernoulli distribution with parameter $\mu_i = 0.15$ (these arrivals correspond to a single simulation step in SUMO) and $\beta_i = 0.1$ for $i = 1,2$. We choose to do a single simulation run

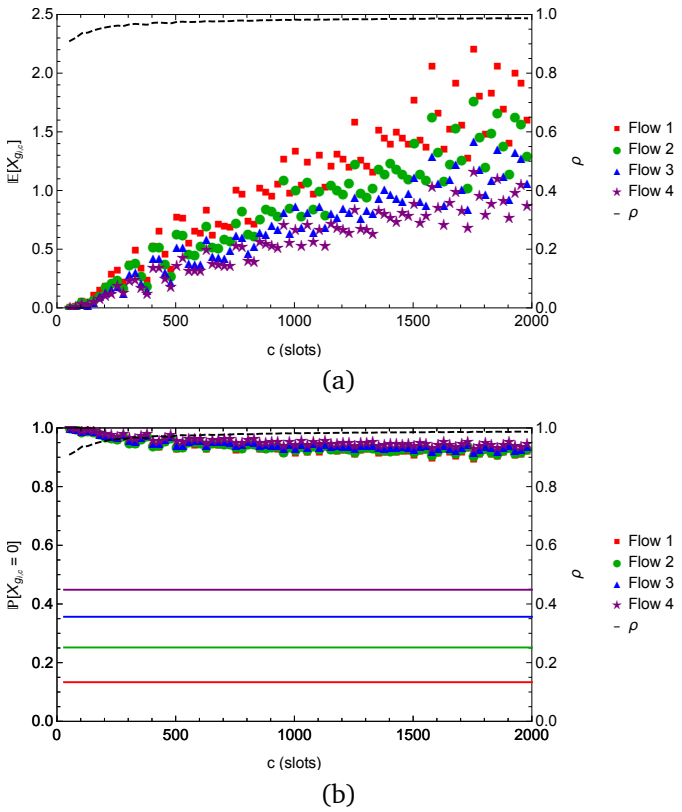


Figure 4.3: The simulated mean queue length at the end of the access period for the four traffic flows in Example 1b in (a); and the simulated probability of an empty queue at the end of the access period for the four traffic flows in Example 1b and the limiting value of that probability for the FCTL queue (solid lines) in (b). In both subfigures we added the vehicle-to-capacity ratio ρ on the right axis (dashed line).

of 3,600,000 steps, because this gives results that are (more than) sufficiently accurate for our purposes.

One of the main difficulties in this example is to define a slot as the period between departures is not constant in SUMO. Partly because of this, it is also difficult to compute the vehicle-to-capacity ratio and to determine whether we

are close to oversaturation. We obtain a measure for this ratio by dividing the average effective vehicle access time per cycle by the maximum specified access period. This ratio is close to one, because we are close to oversaturation.

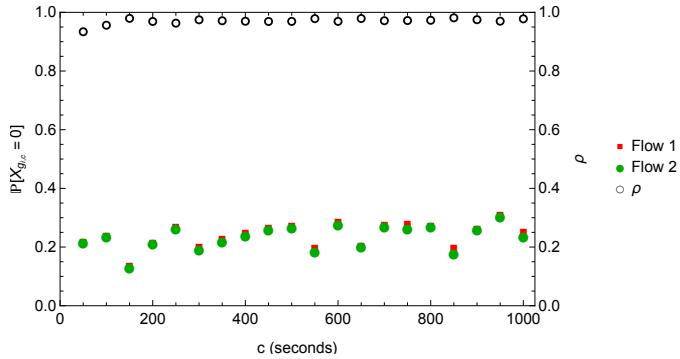


Figure 4.4: The simulated probability of an empty queue at the end of the access period for each of the two traffic flows. The black circles represent the vehicle-to-capacity ratio ρ as displayed on the right axis.

In Figure 4.4 we see the probability of an empty queue at the end of the access period (as determined by the actuated control mechanism in SUMO) and the vehicle-to-capacity ratio. Qualitatively, we observe similar behavior as in Examples 1a and 1b. The vehicle-to-capacity ratio approaches 1 quickly, yet the probability of an empty queue remains between 0 and 1, which shows that our discrete-event simulations are able to capture the essential queueing behavior of vehicles at intersections with an actuated control sufficiently well.

In Example 1b we saw that the empty-queue probabilities seem to converge to the same value for each of the lanes. We clarified this by arguing that an empty queue implies an early switch to the next queue which in turn results in a relatively large probability of that queue being empty at the end of the phase too. This seems to be confirmed by the SUMO simulation. When looking in detail at the moments that an early switch occurs in this example, those moments seem to be clustered. This points in the same direction as the argument that we gave for the observed behavior of the probability that a queue at the end of the access period is empty.

4.4.2 Multiple-lane access control

Example 2a

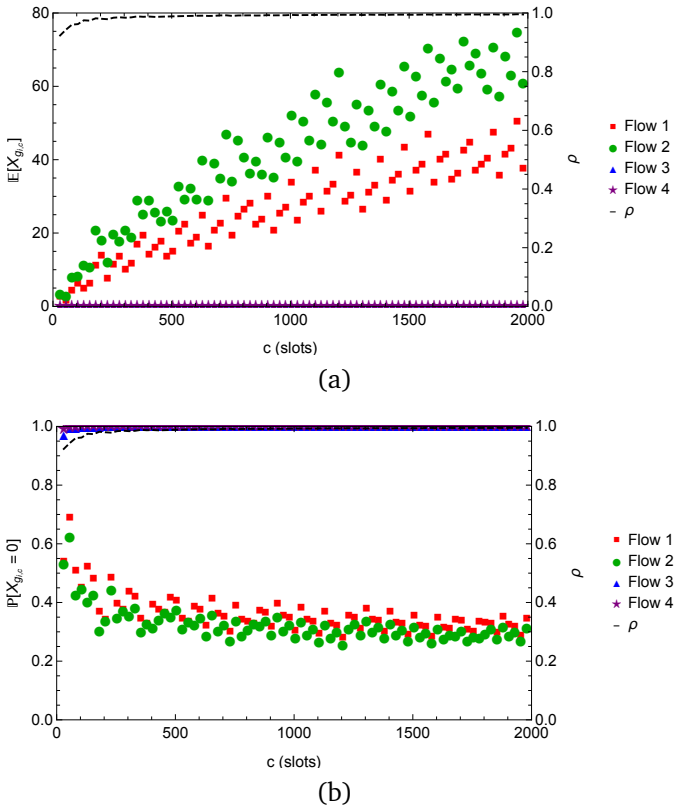


Figure 4.5: The simulated mean queue length at the end of the access period for the four traffic flows in Example 2a in (a); and the simulated probability of an empty queue at the end of the access period for the four traffic flows in Example 2a in (b). In both subfigures we added the vehicle-to-capacity ratio ρ on the right axis (dashed line).

Also in this example we assume that the intersection has four lanes and only straight-going traffic, yet here we combine the two opposing non-conflicting lanes in a single phase, i.e. $J_1 = \{1,3\}$ and $J_2 = \{2,4\}$, see Figure 4.1(b) for a

graphical representation. This allows for a higher load on the intersection, as twice as many vehicles are allowed to depart at the same time in comparison with the Examples 1a and 1b. We choose $\mu_1 = 0.25$, $\mu_2 = 0.5$, $\mu_3 = 0.15$, $\mu_4 = 0.3$, all arrival distributions to be Poisson, and $\beta_i = 0.1$. We present results for the mean queue length and the probability of an empty queue, both at the end of the access period, see Figures 4.5(a) and 4.5(b).

Lanes 1 and 2, the lanes with the highest load, show similar behavior as in Examples 1a and 1b. However, different behavior is observed for lanes 3 and 4. The access period is too long, because the length of the access period is dominated by lanes 1 and 2 that face more traffic. This implies that the queue is (very often) empty at the end of the access period for both lanes 3 and 4, as can be observed in Figures 4.5(a) and 4.5(b) (the purple stars are on top of the blue triangles). This stresses the fact that the additional part of the access period, compared to what is needed to ensure stability, is of the wrong order. Indeed, the additional part is of order c, whereas the right order is $\sqrt{c}$.

Example 2b

This example is the same as Example 2a, except that $\mu_3 = 0.25$ and $\mu_4 = 0.5$. In this example, the load on the lanes in each of the phases is the same. In Figures 4.6(a) and 4.6(b) we see that both lanes inside a phase behave similarly and we observe the same desirable properties as in the other examples. From a mathematical point of view, this is an interesting result, because the model at hand is notoriously difficult to study. Already the queueing model in case of single-lane access control is intractable, but the case of a queueing model with multiple-lane access control is possibly even more complex, as it relates to a polling model with multiple servers [24].

When comparing Figures 4.5(a) and 4.6(a), we see that the limiting value of the mean queue length is considerably higher in Example 2b. This is the result of having longer access periods (on average) for both phases, as we only switch to the next phase when *both* queues are empty, while both queues are on average equal in length. So, usually we switch later to the next phase in Example 2b, which causes the queues at other lanes to be longer, resulting in a higher mean queue length. The same intuition seems to hold for the decrease in the probability of an empty queue, see Figure 4.5(b) and 4.6(b).

Examples 2a and 2b do not immediately indicate that a convenient setup of each phase is one in which each of the lanes has more or less the same load. The lane with the highest load is dominating the length of the phase in Example 2a (which is favorable), but some capacity is “lost” for the lanes with

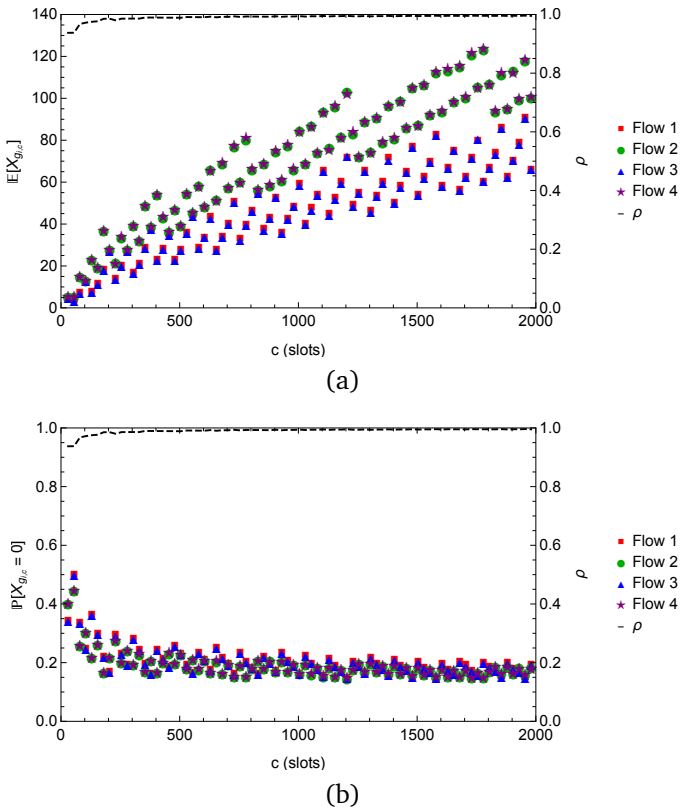


Figure 4.6: The simulated mean queue length at the end of the access period for the four traffic flows in Example 2b in (a) and the simulated probability of an empty queue at the end of the access period for the four traffic flows in Example 2b in (b). In both subfigures we added the vehicle-to-capacity ratio ρ on the right axis (dashed line).

a lower load in that phase. When there is a queue, the outflow is a single car per slot. However, if there is no queue at a lane, we have a lower outflow equal to the arrival rate, which is strictly smaller than a single car per slot. On the other hand, the longer access periods in Example 2b cause some negative effects as well, due to longer time periods in which the queues at other lanes can accumulate. Based on this, it is not clear which is the best option. Note that

a direct comparison of the mean queue lengths in Examples 2a and 2b is not fair, as the total load on the intersection is not the same in the two examples.

SUMO Example

We also perform a SUMO simulation for the double-lane access control strategy considered in this subsection. We do so for the same reasons as in the single-lane access control setting: to serve as a proof of concept that an actuated control with the desirable properties, as seen in Examples 2a and 2b, is achievable. We choose $N = 4$, the arrival distributions to be Bernoulli distributions with parameters $\mu_1 = \mu_2 = 0.15$, $\mu_3 = \mu_4 = 0.1$, and $\beta_i = 0.1$. We take the same approach as in Subsection 4.4.1, e.g. performing a single simulation run of $3{,}600{,}000$ steps. We note that the maximum load on phases 1 and 2 in this example is the same as the load on each lane in the example studied in Subsection 4.4.1. We obtain Figure 4.7.

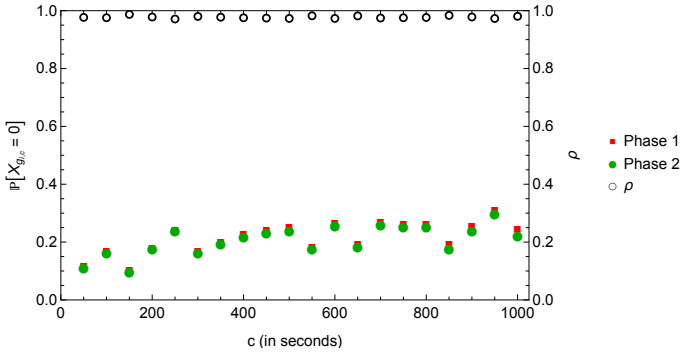


Figure 4.7: The simulated probability of both queues in the same phase being empty at the end of the access period for each of the two phases. The black circles represent the vehicle-to-capacity ratio ρ as displayed on the right axis.

In Figure 4.7, we plot the probability that *both* lanes at the end of the access period are empty. We do not distinguish between the individual lanes inside a phase being empty, because that would be more difficult to simulate in SUMO. As a result, we display only two probabilities and not four empty-queue probabilities as in Examples 2a and 2b. Nevertheless, we observe a similar pattern as in Figure 4.4 (this also seems to be in line with the similarities between Example 2a and Example 1a). The empty-queue probabilities in Figure 4.7 are

(slightly) below the probabilities in Figure 4.4, which makes sense: the probability that *both* lanes are empty in the double-lane access control setting is lower than the probability of a *single* queue being empty in the single-lane access control scenario studied in Figure 4.4. This is more prominent in case of smaller cycle lengths, because the within-cycle variability is relatively large. This causes the probability that the lane with the lower load is non-empty at the end of the access period to be relatively high when compared with larger cycle lengths.

We also studied a double-lance access example where the loads on both queues in a phase are the same (mimicking the setting in Example 2b) and we obtained qualitatively similar results as in Figure 4.7. The probabilities that the green periods terminate early are lower than the corresponding probabilities in Figure 4.7, similarly to the differences observed between Examples 2a and 2b.

4.5 Conclusion

We have shown, with the aid of simulation, that desirable properties are achievable for actuated traffic control of isolated intersections when using a scaling rule such as Equation (4.1). Those properties are similar to the ones established for the FCTL queue in Chapter 3. We have investigated several setups and in each of those, we have observed those desirable properties. One such property is that the limiting probability of an empty queue at the end of the access period is strictly between 0 and 1. We also observed this in the simulation experiments that we performed in SUMO, indicating that our results seem to be qualitatively reliable for real traffic (remember that the other simulations are discrete-event simulations). Another desirable property is that the mean queue length at the end of the access period grows only with order $\sqrt{c}$.

Based on our experiments, it is not clear whether lanes with the same load should be combined into the same phase, something which was already observed by Newell and Osuna in 1969 [143]. If lanes in the same phase have different loads, the lane with the highest load dominates the length of the access period. This is favorable, because vehicles at other lanes accumulate over a relatively short amount of time when compared to a case where lanes with the same load are combined into a single phase (as such a setup results in longer access periods, which in turn increases the queue length at other lanes). On the other hand, when lanes with different loads are combined into a single phase, the lanes with a lower load structurally receive an access period that is too long, which results in a drop of the average outflow for those lanes after they become empty.

A possible extension is to obtain an exact *limiting process* as is obtained in Chapter 3 for the FCTL queue. It would add to the understanding of the model at hand and would be very interesting from a mathematical point of view: the models that we consider relate to notoriously hard types of polling models, for which hardly any exact results (being in heavy traffic or not) have been obtained. A clear path to achieve this is absent, unfortunately. In Chapter 3, we used a counter-integral expression for the PGF of the overflow queue, for which we do not have an alternative in the case of vehicle-actuated traffic-light control.

Chapter 5

Fixed-Cycle Traffic-Light queue with multiple lanes and blockages

5.1 Introduction

The FCTL queue studied in Chapters 2 and 3 cannot always be applied as an accurate model to study the queue-length distribution in front of a traffic light. Take, for example, an intersection where vehicles from a single stream are spread onto two lanes which are both heading straight and where both lanes are governed by the same traffic light. Then one could analyze each lane separately as in the FCTL queue, but that is not entirely realistic, see also Figure 5.1(a). Indeed, since there are two parallel lanes in each direction, two vehicles can cross the intersection simultaneously and vehicles will in general switch lanes (if needed) to join the lane with the shorter queue. Moreover, it might be the case that the vehicles are blocked during the green period, e.g. because of a pedestrian crossing the intersection (receiving a green light at the same time as the stream of vehicles that we model), see Figure 5.1(b) for a visualization. Such blockages also occur in a multi-lane scenario (where all lanes are going in the same direction) as visualized in Figure 5.1(c). The study in this chapter provides an extension of the FCTL queue, which we call the blocked Fixed-Cycle Traffic-Light (bFCTL) queue with multiple lanes, to account for such situations.

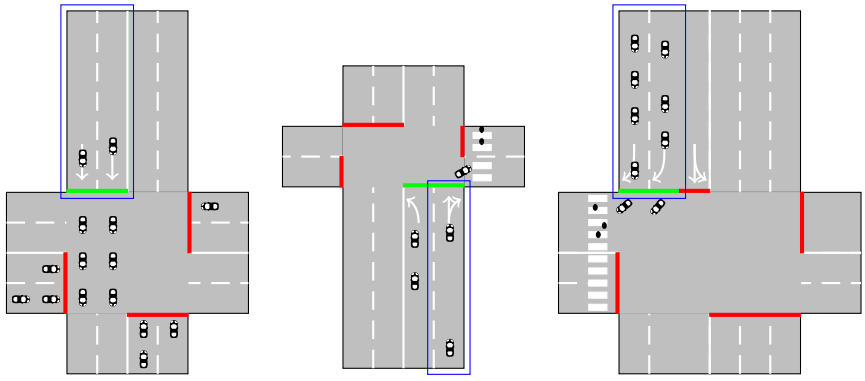


Figure 5.1: A visualization of three intersections that can be modeled by the bFCTL queue with multiple lanes. In (a), the blue rectangle indicates a combination of lanes which can be analyzed as a bFCTL queue with two lanes. The other lanes at the intersection, the complement of the blue rectangle, can be considered separately because of the fixed settings. In (b), the blue rectangle indicates a lane that can be modeled as a bFCTL queue with a single lane with blockages that mimics the setting with pedestrians in [96]. In (c), the blue rectangle indicates two lanes that we can model as a bFCTL queue with *two* lanes where vehicles are potentially blocked by pedestrians.

A shared right-turn lane as in Figure 5.1(b), that is a lane with vehicles that are either turning right or are heading straight, has been studied before. However, to the best of our knowledge, there are no papers with a rigorous analysis taking stochastic effects into account to compute e.g. the mean queue length for such lanes. Shared right-turn lanes where vehicles are blocked by pedestrians crossing immediately after the right turn have been considered in e.g. [8, 44–46, 135, 171, 172]. Several case studies, such as [45, 171] indicate that there is a potentially severe impact by pedestrians blocking vehicles. This is for example also reflected in the Highway Capacity Manual (HCM) [192], where the focus is on capacity estimation. Most papers have also focused on the estimation of the so-called saturation flow rate, or capacity, of shared lanes where turning vehicles are possibly blocked by pedestrians, see e.g. [46, 135, 172]. In [44], it is stated that the used functions for the capacity estimation for turning lanes (such as those in the HCM) might have to be extended to account

for stochastic behavior. In a small case study, the authors in [44] confirm that the capacity estimation by the HCM [192] yields an overestimation in various cases. The overestimation of the capacity by the HCM is also observed in several other papers, such as in [45, 46, 96], and is probably due to random/stochastic effects.

As mentioned before, we call the model that we consider in this chapter the bFCTL queue with multiple lanes. On the one hand we thus allow for the modeling of vehicle streams that are spread over multiple lanes and on the other hand we allow for vehicles to be (temporarily) blocked during the green phase. The key observation to constructing the mathematical model is that we can model multiple parallel (say m) lanes as *one* single queue where batches of (up to) m delayed vehicles can depart in one time slot, for more details see Section 5.2. The resulting queueing model is one-dimensional just like the standard FCTL queue, which allows us to obtain the PGF of the steady-state queue-length distribution of the bFCTL queue with multiple lanes. A slightly different version of the bFCTL queue with a single lane has been studied by means of simulation in a recent paper by Huang et al. [96], which has been the inspiration for the study in the present chapter.

The model that is studied in [96] is thus a potential application of the bFCTL queue with a single lane as depicted in Figure 5.1(b). A description of the model in [96] is as follows, where we replace the left-turn assumption for left-driving traffic to a right-turn assumption for the more standard case of right-driving traffic. We have a shared lane with straight-going and right-turning traffic controlled by a traffic light, where immediately after the right turn there is a crossing for pedestrians. The pedestrians may block the right-turning vehicles as the vehicles and pedestrians may receive a green light simultaneously. The right-turning vehicles that are blocked, immediately block all vehicles behind them.

In most traffic-light models (such as the FCTL queue), such situations are not considered at all, which makes them less suitable for intersection modeling where conflicts may arise due to multiple traffic flows receiving a green light simultaneously. Another potential application of the bFCTL queue is to account for bike lanes. Bikes might make use of a dedicated lane or mix with other traffic and in both cases a turning vehicle might be (temporarily) blocked by bicycles because the bicycles happen to be in between the vehicle and the direction that the vehicle is going. As such, blockages have an influence on the performance measures of the traffic light. It is important to take such influences into account in order to find good traffic-light settings. Several papers studying the impact of bikes can be found in [9, 42, 47, 87]. Also other types of blocking might occur, such as by a shared-left turn lane and opposing traffic receiving a green light

simultaneously, see e.g. [38, 122, 127, 128, 131, 224–226]. As such, the bFCTL queue (either with multiple lanes or not) is a relevant addition to the literature because it enables a more suitable modeling of traffic lights at intersections with crossing pedestrians and bikes, which leads to traffic-light control strategies for more realistic situations. In order to model a situation where two opposing streams of vehicles potentially block one another as in e.g. [225], the bFCTL queue would have to be extended. For more references on the topics discussed in this paragraph see also the review paper [49].

One of the studies that is close to ours is Oblakova et al. [146]. In Section 4.4 of [146], the standard FCTL model is supplemented with the possibility that drivers are "distracted". At each moment that a driver is allowed to depart from the queue, it departs with a probability p. If the vehicle does not depart from the queue while it is actually allowed to do so, we might view this as the vehicle being blocked. The difference between the bFCTL queue with a single lane and the model of Oblakova et al. is that every blockage at a departure moment in [146] occurs independently. This is not the case in the bFCTL queue as once a right-turning vehicle is blocked, it will (most likely) be blocked for a longer period than a single departure moment. This introduces subtle dependencies in the model which lead to an additional dimension in the state space of the underlying Markov chain.

The study by Huang et al. in [96] is closely related to ours although the used techniques are different. The bFCTL queue is based on the model described in Huang et al. and the models are quite similar. A contribution compared to their study lies in the possibility for exact computations instead of the need to rely on simulation experiments. We alleviate some of the assumptions in [96]. For example, in the model studied by Huang et al. it is assumed that there are always pedestrians crossing if the pedestrians have a green light. We allow for the random presence of pedestrians if the pedestrians have a green light. On the other hand, we put some additional constraints on the bFCTL queue compared to the model studied in [96]. We e.g. do not consider start-up delays.

In summary, our main contributions are as follows:

(i) We extend the general applicability of the FCTL queue. We allow for traffic streams with multiple lanes and for vehicles to be blocked during the green phase. We refer to this model variation as the blocked Fixed-Cycle Traffic-Light (bFCTL) queue with multiple lanes.

(ii) We provide a way to compute the PGF of the steady-state queue-length distribution of the bFCTL queue and show that it can be used to obtain several performance measures of interest.

(iii) We provide a queueing-theoretic framework for the study of shared lanes with potential blockages by pedestrians. This e.g. allows for the study of several performance measures and allows us to model the impact of randomness on the performance measures.

Chapter outline

The remainder of this chapter is organized as follows. In Section 5.2, we give a detailed model description. This is followed by a derivation of the PGF of the steady-state queue-length distribution and a derivation of some of the main performance measures in Section 5.3. In Section 5.4, we provide an overview of relevant performance measures for some numerical examples and point out various interesting results. We wrap up with a conclusion and some suggestions for future research in Section 5.5.

5.2 Detailed model description

In this section we provide a detailed model description of the bFCTL queue with multiple lanes.

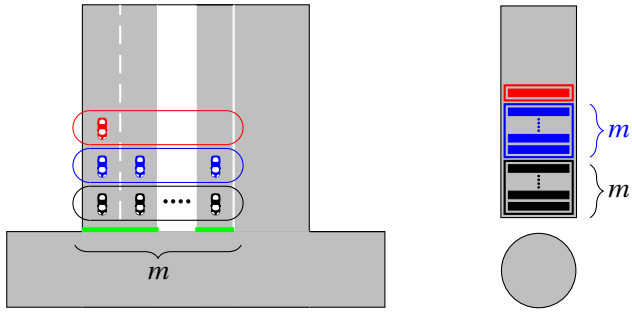


Figure 5.2: Visualization of (a) the bFCTL model in terms of an intersection with a traffic stream spread over m lanes and (b) the corresponding queueing model, where the server takes batches of m vehicles into service simultaneously unless there are less than m vehicles present: in that case all vehicles are taken into service.

We assume that there are multiple lanes for a traffic stream, that is a group of vehicles coming from the same road and heading into one (or several) di-

rection(s), governed by a *single* traffic light. A visualization can be found in Figure 5.2(a). As can be seen in Figure 5.2(a), we assume that there are m lanes and that vehicles spread themselves among the available lanes in such a way that m vehicles can depart if there are at least m vehicles. In practice, this assumption makes sense as drivers gladly minimize their delay by choosing free lanes. The traffic-light model is then turned into a queueing model with a *single* queue with batch services of vehicles, see Figure 5.2(b). The batches generally consist of m delayed vehicles (we consider delayed vehicles as we did in the FCTL queue), except if less than m delayed vehicles are present at the moment that a batch is taken into service: then all vehicles are taken into service. We further assume that the time axis is divided into time intervals of constant length, where each interval corresponds to the time it takes for a batch of delayed vehicles to depart from the queue. We will refer to these intervals as slots.

We now turn to discuss two concrete examples that fit the framework of the bFCTL queue with multiple lanes. After that, we describe the assumptions of the bFCTL queue more formally.

Example 5.1 (Shared right-turn lane) *In this example we consider the scenario as in Figure 5.1(b). We have batches of vehicles of size* 1*, i.e. batches are individual vehicles.*

We distinguish between vehicles that are going straight ahead and vehicles that turn right. We do so because only right-turning vehicles can be blocked by crossing pedestrians. The probability that an arbitrary vehicle at the head of the queue is a turning vehicle is p. Such a turning vehicle is blocked by a pedestrian in slot i with probability q_i, i.e. a pedestrian is present on the crossing with probability q_i. If a turning vehicle is blocked, all vehicles behind it are also blocked. Then, we proceed to the next slot, $i+1$, and check whether there are any pedestrians crossing (with probability q_{i+1}): if there are pedestrians crossing, all vehicles in the queue keep being blocked and otherwise, the turning vehicle at the head of the queue may depart and the blockage of all other vehicles is removed.

Moreover, if the queue becomes empty during the green period, it will in general not start building again (cf. the FCTL assumption, see Assumption 1.1), except *if there arrives a turning vehicle and there is a crossing pedestrian. The turning vehicle is then blocked and any vehicles arriving in the same slot behind this vehicle are also blocked.*

Example 5.2 (Two turning lanes) *In this example we consider the scenario as in Figure 5.1(c). We have batches of vehicles of size* 2*.*

In this example, there is no need to make a distinction between vehicles: each vehicle is a turning vehicle with probability 1*, i.e.* $p = 1$*. During each slot* i*, there are pedestrians on the crossing with probability* q_i *and if there is a pedestrian, all vehicles in the batch are blocked, as are all other vehicles in the queue: there are no vehicles that can complete the right turn. All vehicles in the queue keep being blocked until there are no pedestrians crossing anymore.*

Also in this example, the queue of vehicles might dissolve entirely during the green period. If that happens, it only starts building again if there are vehicles arriving and *if there are pedestrians crossing. In such cases, all arriving vehicles get blocked and remain blocked until there are no pedestrians anymore.*

We are now set to formalize the assumptions for the bFCTL queue with multiple lanes. We number them for clarity and provide additional remarks if necessary. We start with a standard assumption for FCTL queues and a standard assumption on the independence of arriving vehicles, see, e.g., [206].

Assumption 5.1 (Discrete-time assumption) *We divide time into discrete slots. The red and green times,* r *and* g *respectively, are fixed multiples of those discrete slots and the total cycle length,* $c = g + r$*, thus consists of an integer number of slots. Each slot corresponds to the duration of the departure of a batch of maximally* m *delayed vehicles, where* m *is the maximum number of vehicles that can cross the intersection simultaneously. Any arriving vehicle that finds at least* m *other vehicles waiting in front of the traffic light is delayed and joins the queue.*

Assumption 5.2 (Independence of arrivals) *All arrivals are assumed to be independent. In particular, the arrivals during slot* i *do not affect the arrivals in slot* j *when* $i \neq j$*.*

The next three assumptions, Assumptions 5.3, 5.4, and 5.5, relate to the blockages and are a generalization of the assumptions discussed in Section III.B.1 of [96].

Assumption 5.3 (Green period division) *For the green period we distinguish between two parts,* g_1 *and* g_2*, with* $g = g_1 + g_2$*. During the first part of the green period, blockages might occur (see also Assumption 5.4 below). During the second part of the green period there are no blockages at all. We further assume that* $g_2 > 0$ *for technical reasons.*

We make a division of the green period into two parts, because such a division is often present in reality and because it slightly eases the computations later on. We note that if $g_1 = 0$ (and $m = 1$), we obtain the standard FCTL queue.

Further, we assume that the second part of the green period is strictly positive, mainly for technical reasons. This basically implies that at least one batch of vehicles can depart from the queue during each cycle and that there is *no* batch of vehicles in the queue at the end of the cycle that has caused a blockage before. If g_2 would be zero and if a batch of vehicles is blocked at the end of slot g_1, this would allow for a blockage to carry over to the next cycle, leading to a slightly more complex model.

Next, we make an assumption about the blocking of batches of vehicles during the first part of the green period. We take into account that (i) not all batches of vehicles at the head of the queue are potentially blocked (e.g. because only turning batches of vehicles can be blocked); that (ii) if a batch of vehicles is blocked, all vehicles behind it are blocked as well; that (iii) once a blockage occurs, it carries over to the next slot; and that (iv) blockages occur only in the combined event of having a right-turning batch at the head of the queue *and* pedestrians crossing the road.

Assumption 5.4 (Potential blocking of batches) *A batch of vehicles, arriving at the head of the queue in time slot i, turns right with probability p_i. Independently, in time slot j, pedestrians cross the road with probability q_j, blocking right-turning traffic from the main road. As a consequence, whenever a new batch arrives at the head of the queue, this batch will be served in that particular time slot if (i) the batch goes straight ahead,* or *(ii) the batch turns right but there are no crossing pedestrians. Once a batch (of right-turning vehicles) is blocked, it will remain blocked until the next time slot when no pedestrians cross the road. Note that this will be time slot $g_1 + 1$ at the latest. If the batch at the head of the queue is blocked, it will also block all the other batches in the queue, including those that would go straight.*

Remark 5.1 *We make a couple of remarks on the values of the p_i. First, whether a batch of vehicles is a right-turning batch or not, in general does not depend on the slot in which the batch gets to the head of the queue. This would imply that $p_i = p$ (see, e.g., Example 5.2) and that we could drop the subscript i. However, we are able to let p_i depend on the slot in the derivation of the formulas and opt to provide the general case where p_i is allowed to depend on i.*

Moreover, in the case that $m > 1$, we will in practice often have that either $p_i = 0$, as is the case in Figure 5.1(a), or $p_i = 1$, as is the case in Figure 5.1(c). This is mainly due to the fact that all *vehicles in a batch have to be treated similarly: the framework of the bFCTL queue does not allow for batches consisting of one right-turning vehicle that is blocked and one straight-going vehicle that is allowed*

to depart because it is not blocked. I.e. a case with mixed traffic and multiple *lanes, such as the shared right-turn lane example in Figure 5.1(b) but with $m > 1$, is not modeled by the bFCTL queue.*

Remark 5.2 *We would like to stress that the blockage of a batch of vehicles carries over to the next slot. E.g. if a vehicle is a right-turning vehicle in Figure 5.1(b) and is blocked, it is still at the head of the queue in the next slot. So, as soon as a blockage actually takes place, we are essentially in a different state of the system than in the case where there is no blockage: if there is a blockage in time slot i then we are sure that there is a right-turning batch at the head of the queue in time slot $i+1$. This is why we have two mechanisms for the blocking: on the one hand we have the p_i to check whether batches are right turning and on the other hand we have the pedestrians crossing in slot i accounted for by the q_i.*

We need one final assumption which is a slightly adapted version of the standard FCTL assumption. We require a slight change because of the potential blocking of vehicles during the first part of the green phase and because of the possibility that there is more than one delayed vehicle departing in a single slot during the green period because of the batch-service structure.

Assumption 5.5 (bFCTL assumption) *We assume that any vehicle arriving during a slot where $m-1$ or less vehicles are in the queue, may depart from the queue immediately together with the $m-1$ or less delayed vehicles. There are two exceptions: (i) if this batch of $m-1$ or less vehicles is blocked or (ii) if the queue was empty and there is an arriving vehicle that gets blocked, then that vehicle gets blocked together with any arriving vehicles after that vehicle. In the former case, all arriving vehicles together with the delayed vehicles remain at the queue, whereas in the latter case, the first blocked vehicle is delayed and any arriving vehicles behind it (if any) are also delayed and blocked. For the latter case we restrict ourselves to the situation where the queue is empty: if the queue was not empty, then we assume that either all arriving vehicles in that slot are blocked and delayed (because the batch at the head of the queue is blocked) or that all arriving vehicles are allowed to depart along with the batch of delayed vehicles (because the batch at the head of the queue is not blocked).*

Remark 5.3 *The bFCTL assumption allows one to model a situation where arriving vehicles get blocked if the queue was already empty before the start of the slot. Although, in principle, one can use any distribution for the number of arriving vehicles that are blocked, there are only few logical choices in practice. For example, in the case of Figure 5.1(b), the number of (potentially) blocked vehicles that*

arrive at the queue during slot i would correspond to the number of vehicles counting from the first right-turning vehicle among all vehicles arriving in slot i: these vehicles will be blocked if there is a crossing pedestrian in slot i. In Figure 5.1(c), any arriving vehicle is a turning vehicle. So, if there is a crossing pedestrian, all arriving vehicles in slot i are blocked.

The combination of all the above assumptions enables us to view the process as a discrete-time Markov chain, which in turn allows us to obtain the PGF of the steady-state queue-length distribution of the bFCTL queue with multiple lanes. We derive this PGF implicitly by means of a recursion in the next section.

5.3 PGFs and performance measures for the bFCTL queue

In this section we provide the derivation of the steady-state queue-length distribution in terms of PGFs in Subsection 5.3.1, after which we turn to the most important performance measures in Subsection 5.3.2.

5.3.1 Derivation of the PGFs for the bFCTL queue

First, we need to introduce some further concepts and notation before we continue our quest to obtain the relevant PGFs of the queue-length distribution. We introduce two states, one corresponding to a situation where the queue is blocked and one where this is not the case, cf. Assumption 5.4 and Remark 5.2. We denote the random variable of being in either of the two states with S and S takes the values b (blocked) and u (unblocked). By definition, blocked states only occur during the first part of the green period and if there are vehicles in the queue. We define S to be equal to u if the queue is empty. We denote the joint steady-state queue length (measured in number of vehicles) and the state S at the end of slot $i = 1,\dots,g_1$ with the tuple (X_i, S) and we denote its PGF with $X_{i,j}(z)$ where $i = 1,\dots,g_1$ and $j = u, b$. We note that $X_{i,b}(z)$ and $X_{i,u}(z)$ are partial generating functions: we e.g. have $X_{i,b}(z) = \mathbb{E}[z^{X_i}\mathbb{1}\{S = b\}]$, where $\mathbb{1}\{S = b\} = 1$ if $S = b$ and 0 otherwise. For the slots $i = 1,\dots,c$ we denote the steady-state queue length with X_i and its PGF with $X_i(z)$, so for $i = 1,\dots,g_1$ we have that $X_i(z) = X_{i,u}(z) + X_{i,b}(z)$.

We note that, as we are looking at the steady-state distribution of the number of vehicles in the queue, we need to require stability of the queueing model (i.e. on average there are fewer arrivals in a cycle than delayed vehicles departing

from queue during a cycle). We refrain from giving the stability condition in the general case because of its complicated expression. However, in Appendix 5.A, we present an algorithm to check whether the stability condition is satisfied.

We further denote with Y_i the number of arrivals during slot i and with $Y_{i,b}$ we denote the total number of arrivals of potentially blocked vehicles during slot i, see also Assumption 5.5. We denote their PGFs respectively with $Y_i(z)$ and $Y_{i,b}(z)$. Later in this subsection, we provide $Y_{i,b}(z)$ for several examples.

As a last remark, we note that we will refer to $X_{g_1+g_2}$, or alternatively X_g, as the overflow queue as this is the queue length at the end of the green period (similar to the overflow queue in the FCTL queue).

In the next part of this subsection, we provide the recursion between the $X_{i,j}(z)$, $i = 1,\dots,g_1$ and $j = u,b$, and the $X_i(z)$, $i = g_1+1,\dots,c$. Afterwards, we wrap up with some technicalities that need to be overcome to obtain a full characterization of all the PGFs.

Recursion for the $X_{i,j}(z)$

We start with the relation between $X_{1,b}(z)$ and $X_c(z)$. We distinguish several cases while making a transition from slot c to a blocked state in slot 1. We get

$$\begin{aligned} X_{1,b}(z) =& p_1 q_1 \mathbb{E}[z^{X_c+Y_1}\mathbb{1}\{X_c>0\}] + q_1\mathbb{E}[z^{Y_{1,b}}\mathbb{1}\{X_c=0\}\mathbb{1}\{Y_{1,b}>0\}] + \\ & 0\cdot\mathbb{E}[\mathbb{1}\{X_c=0\}\mathbb{1}\{Y_{1,b}=0\}] \\ =& p_1 q_1 X_c(z)Y_1(z) + q_1\mathbb{P}(X_c=0)\left(Y_{1,b}(z) - Y_{1,b}(0) - p_1 Y_1(z)\right). \end{aligned} \tag{5.1}$$

We explain this relation as follows: if the queue is nonempty at the end of slot c, we need both a right-turning batch of vehicles and a crossing pedestrian in slot 1 to get a blockage, which happens with probability $p_1 q_1$. The queue length at the end of slot 1 is then $X_c + Y_1$. The second term can be understood as follows: if $X_c = 0$, the queue at the end of slot c is empty and then we get to a blocked state if there is a pedestrian crossing (which happens with probability q_1) and if $Y_{1,b} > 0$, in which case the queue length is $Y_{1,b}$. Note that we further have that the case $X_{1,b} = 0$ cannot occur (by definition) as indicated by the term on the second line of Equation (5.1).

Similarly, we derive $X_{1,u}(z)$:

$$\begin{aligned} X_{1,u}(z) =& (1-p_1q_1)\mathbb{E}[z^{X_c+Y_1-m}\mathbb{1}\{X_c\geq m\}] + (1-p_1q_1)\mathbb{E}[z^0\mathbb{1}\{1\leq X_c\leq m-1\}] + \\ & (1-q_1)\mathbb{E}[z^0\mathbb{1}\{X_c=0\}] + q_1\mathbb{E}[z^0\mathbb{1}\{X_c=0\}\mathbb{1}\{Y_{1,b}=0\}] \\ =& (1-p_1q_1)X_c(z)\frac{Y_1(z)}{z^m} + (1-p_1q_1)\sum_{l=1}^{m-1}\mathbb{P}(X_c=l)\left(1-\frac{Y_1(z)}{z^{m-l}}\right) + \end{aligned} \tag{5.2}$$

$$\mathbb{P}(X_c = 0)\left(1 - q_1 + q_1 Y_{1,b}(0) - (1 - p_1 q_1)\frac{Y_1(z)}{z^m}\right).$$

This relation can be understood in the following way: first, if there are at least m vehicles at the end of slot c and if there is no blockage (which occurs with probability $1 - p_1 q_1$, i.e. the complement of a blockage occurring), then the queue length at the end of slot 1 is $X_c + Y_1 - m$. Secondly, if there is at least 1 but at most $m-1$ vehicles at the end of slot c, we have an empty queue at the end of slot 1 if there is no blockage (which is the case with probability $1 - p_1 q_1$). Thirdly, if the queue is empty at the end of slot c, then the queue remains empty if there are no pedestrians crossing (occurring with probability $1 - q_1$) or if there is a pedestrian crossing (occurring with probability q_1) while $Y_{1,b} = 0$. This fully explains Equation (5.2).

In a similar way, we obtain the following relations for slots $i = 2, \ldots, g_1$:

$$\begin{aligned} X_{i,b}(z) =& p_i q_i \mathbb{E}[z^{X_{i-1}+Y_i}\mathbb{1}\{S = u\}] + q_i \mathbb{E}[z^{X_{i-1}+Y_i}\mathbb{1}\{S = b\}] + \\ & q_i \mathbb{E}[z^{Y_{i,b}}\mathbb{1}\{X_{i-1} = 0\}\mathbb{1}\{S = u\}\mathbb{1}\{Y_{i,b} > 0\}] \\ =& p_i q_i X_{i-1,u}(z) Y_i(z) + q_i X_{i-1,b}(z) Y_i(z) + \\ & q_i \mathbb{P}(X_{i-1} = 0, S = u)\left(Y_{i,b}(z) - Y_{i,b}(0) - p_i Y_i(z)\right), \end{aligned} \tag{5.3}$$

where we have to take both transitions from slot $i-1$ while being blocked (the case $S = b$) and not being blocked (the case $S = u$) into account, and

$$\begin{aligned} X_{i,u}(z) =& (1 - p_i q_i)\mathbb{E}[z^{X_{i-1}+Y_i-m}\mathbb{1}\{X_{i-1} \geq m\}\mathbb{1}\{S = u\}] + \\ & (1 - q_i)\mathbb{E}[z^{X_{i-1}+Y_i-m}\mathbb{1}\{X_{i-1} \geq m\}\mathbb{1}\{S = b\}] + \\ & (1 - p_i q_i)\mathbb{E}[z^0\mathbb{1}\{1 \leq X_{i-1} \leq m-1\}\mathbb{1}\{S = u\}] + \\ & (1 - q_i)\mathbb{E}[z^0\mathbb{1}\{1 \leq X_{i-1} \leq m-1\}\mathbb{1}\{S = b\}] + \\ & (1 - q_i)\mathbb{E}[z^0\mathbb{1}\{X_{i-1} = 0\}\mathbb{1}\{S = u\}] + \\ & q_i \mathbb{E}[z^0\mathbb{1}\{X_{i-1} = 0\}\mathbb{1}\{S = u\}\mathbb{1}\{Y_{i-1,b} = 0\}] \\ =& (1 - p_i q_i) X_{i-1,u}(z)\frac{Y_i(z)}{z^m} + (1 - q_i) X_{i-1,b}(z)\frac{Y_i(z)}{z^m} + \\ & (1 - p_i q_i)\sum_{l=1}^{m-1} \mathbb{P}(X_{i-1} = l, S = u)\left(1 - \frac{Y_i(z)}{z^{m-l}}\right) + \\ & (1 - q_i)\sum_{l=1}^{m-1} \mathbb{P}(X_{i-1} = l, S = b)\left(1 - \frac{Y_i(z)}{z^{m-l}}\right) + \\ & \mathbb{P}(X_{i-1} = 0, S = u)\left(1 - q_i + q_i Y_{i,b}(0) - (1 - p_i q_i)\frac{Y_i(z)}{z^m}\right). \end{aligned} \tag{5.4}$$

In order to derive $X_{g_1+1}(z)$, we note that we need to take the cases into account where the queue was blocked or not during slot g_1. We then get

$$\begin{aligned} X_{g_1+1}(z) =& \mathbb{E}[z^{X_{g_1}+Y_{g_1+1}-m}\mathbb{1}\{X_{g_1} \geq m\}\mathbb{1}\{S=u\}] + \\ & \mathbb{E}[z^{X_{g_1}+Y_{g_1+1}-m}\mathbb{1}\{X_{g_1} \geq m\}\mathbb{1}\{S=b\}] + \\ & \mathbb{E}[z^{0}\mathbb{1}\{X_{g_1} \leq m-1\}\mathbb{1}\{S=u\}] + \mathbb{E}[z^{0}\mathbb{1}\{X_{g_1} \leq m-1\}\mathbb{1}\{S=b\}] \\ =& \left(X_{g_1,u}(z) + X_{g_1,b}(z)\right)\frac{Y_{g_1+1}(z)}{z^m} + \\ & \sum_{l=0}^{m-1} \mathbb{P}(X_{g_1}=l, S=u)\left(1-\frac{Y_{g_1+1}(z)}{z^{m-l}}\right) + \\ & \sum_{l=1}^{m-1} \mathbb{P}(X_{g_1}=l, S=b)\left(1-\frac{Y_{g_1+1}(z)}{z^{m-l}}\right). \end{aligned} \tag{5.5}$$

For $i = g_1+2,\ldots,g_1+g_2$, we obtain the following

$$\begin{aligned} X_i(z) =& \mathbb{E}[z^{X_{i-1}+Y_i-m}\mathbb{1}\{X_{i-1} \geq m\}] + \mathbb{E}[z^{0}\mathbb{1}\{X_{i-1} \leq m-1\}] \\ =& X_{i-1}(z)\frac{Y_i(z)}{z^m} + \sum_{l=0}^{m-1} \mathbb{P}(X_{i-1}=l)\left(1-\frac{Y_i(z)}{z^{m-l}}\right), \end{aligned} \tag{5.6}$$

while for slots $i = g_1+g_2+1,\ldots,c$ we get

$$X_i(z) = \mathbb{E}[z^{X_{i-1}+Y_i}] = X_{i-1}(z)Y_i(z). \tag{5.7}$$

The combination of all equations above, provides us with a recursion with which we can express $X_{g_1+g_2}(z)$ in terms of $Y_i(z)$, $Y_{i,b}(z)$, $\mathbb{P}(X_i=l,S=u)$ and $\mathbb{P}(X_i=l,S=b)$ for $i=1,\ldots,g_1$ and $l=0,\ldots,m-1$, and $\mathbb{P}(X_i=l)$ for $i=g_1+1,\ldots,g_1+g_2-1$, $i=c$, and $l=0,\ldots,m-1$, with the following general form:

$$X_{g_1+g_2}(z) = \frac{X_n(z)}{X_d(z)}, \tag{5.8}$$

with known $X_n(z)$ and $X_d(z)$. We refrain from giving $X_n(z)$ and $X_d(z)$ in the general case because of their complexity and only provide them under simplifying assumptions later in this subsection. The $Y_i(z)$ are known, but we still need to obtain the $Y_{i,b}(z)$, the $\mathbb{P}(X_i=l,S=u)$ and $\mathbb{P}(X_i=l,S=b)$ for $i=1,\ldots,g_1$ and $l=0,\ldots,m-1$, and the $\mathbb{P}(X_i=l)$ for $i=g_1+1,\ldots,g_1+g_2-1$, $i=c$, and $l=0,\ldots,m-1$. We start with the $Y_{i,b}(z)$ and then come back to the unknown probabilities.

The occurrence of the PGF $Y_{i,b}(z)$ directly relates to Assumption 5.5. As mentioned before in Remark 5.3, one could, a priori, use any positively distributed, discrete random variable. However, when we have a specific example in mind, there is usually one logical definition, see also Remark 5.4 below.

Remark 5.4 *In general, we define $Y_{i,b}$ to be the random variable of the total number of arrivals of potentially blocked vehicles during slot i, cf. Assumption 5.5. In case $m = 1$, such as in Figure 5.1(b), the interpretation of the $Y_{i,b}(z)$ is straightforward. We simply count the number of arriving vehicles starting from the first vehicle that is a turning vehicle. We get the following expression for $Y_{i,b}(z)$:*

$$\begin{aligned}
Y_{i,b}(z) &= \sum_{k=0}^{\infty} \mathbb{P}(Y_{i,b} = k) z^k \\
&= \sum_{j=0}^{\infty} \mathbb{P}(Y_i = j)(1-p_i)^j + \sum_{k=1}^{\infty} \sum_{j=k}^{\infty} \mathbb{P}(Y_i = j)(1-p_i)^{j-k} p_i z^k \\
&= Y_i(1-p_i) + \sum_{j=1}^{\infty} p_i \mathbb{P}(Y_i = j)(1-p_i)^j \sum_{k=1}^{j} \left(\frac{z}{1-p_i}\right)^k \\
&= Y_i(1-p_i) + \sum_{j=1}^{\infty} p_i \mathbb{P}(Y_i = j)(1-p_i)^j z \frac{1 - \left(\frac{z}{1-p_i}\right)^j}{1-p_i-z} \\
&= Y_i(1-p_i) + \frac{p_i z}{1-p_i-z} \sum_{j=1}^{\infty} \mathbb{P}(Y_i = j) \left((1-p_i)^j - z^j\right) \\
&= Y_i(1-p_i) + \frac{p_i z}{1-p_i-z} \left(Y_i(1-p_i) - Y_i(z)\right),
\end{aligned}$$

where in the second step we condition on the total number of arrivals and take into account how we can get to k blocked vehicles; in the third step we interchange the order of the summation; and in the fourth step we compute a geometric series. The remainder is straightforward bookkeeping.

If $m > 1$, the interpretation as above for the case $m = 1$ is not necessarily meaningful. It is more difficult to compute the $Y_{i,b}$ in a logical and consistent way. This has to do with the fact that if $m > 1$ we consider batches of vehicles that are either all blocked or not, whereas the $Y_{i,b}$'s are about individual vehicles. As mentioned before in Remark 5.1, if $m > 1$ we often have that either $p_i = 0$ or $p_i = 1$. If $p_i = 0$, the general expression for $Y_{i,b}(z)$ reduces to:

$$Y_{i,b}(z) = Y_i(1) + 0 \cdot (Y_i(1) - Y_i(z)) = Y_i(1) = 1,$$

which makes sense as there are no turning vehicles in case $p_i = 0$ *and indeed* $Y_{i,b} = 0$ *with probability* 1. *If* $p_i = 1$, *we have that:*

$$Y_{i,b}(z) = Y_i(0) - (Y_i(0) - Y_i(z)) = Y_i(z),$$

which is also logical: every arriving vehicle is a turning vehicle if $p_i = 1$, *so we have that* $Y_{i,b}(z) = Y_i(z)$.

Except for the constants $\mathbb{P}(X_i = l, S = u)$ and $\mathbb{P}(X_i = l, S = b)$ for $i = 1, \ldots, g_1$ and $l = 0, \ldots, m-1$, and $\mathbb{P}(X_i = l)$ for $i = g_1 + 1, \ldots, g_1 + g_2 - 1$, $i = c$, and $l = 0, \ldots, m-1$, we are now done. We explain how to find the (so far) unknown constants in the next part of this subsection. We close this part with several special cases of the bFCTL queue and a couple of further remarks.

Special cases of the bFCTL queue

We study several special cases of the bFCTL queue, e.g. cases where the bFCTL queue reduces to the FCTL queue.

If $q_i = 1$, an explicit expression for the PGF of the distribution of the overflow queue can be written down relatively easily. When it is further assumed, for the ease of exposition, that all $p_i = p$, $Y_i \overset{d}{=} Y$, $Y_{i,b} \overset{d}{=} Y_b$ and $m = 1$, the following expression for $X_{g_1+g_2}(z)$ is obtained:

$$X_{g_1+g_2}(z) = \frac{X_n(z)}{X_d(z)}, \tag{5.9}$$

with

$$\begin{aligned} X_n(z) = z^{g_1+g_2} \sum_{i=0}^{g_2-1} \left(\frac{Y(z)}{z}\right)^{g_2-i-1} \left(1 - \frac{Y(z)}{z}\right) \mathbb{P}(X_{g_1+i} = 0) + z^{g_1} Y(z)^{g_2} \cdot \\ \sum_{i=0}^{g_1-1} \left\{ \mathbb{P}(X_i = 0, S = u) \left[\left(Y_b(0) - (1-p)\frac{Y(z)}{z}\right) \left((1-p)\frac{Y(z)}{z}\right)^{g_1-i-1} + \right.\right. \\ \left.\left. \left(Y_b(z) - Y_b(0) - pY(z)\right) Y(z)^{g_1-i-1} \right] + pY(z)^{g_1-i} \cdot \right. \\ \left. \sum_{j=0}^{i-1} \mathbb{P}(X_j = 0, S = u) \left(Y_b(0) - (1-p)\frac{Y(z)}{z}\right) \left((1-p)\frac{Y(z)}{z}\right)^{i-j-1} \right\}, \end{aligned} \tag{5.10}$$

where $\mathbb{P}(X_0 = 0, S = u)$ is to be interpreted as $\mathbb{P}(X_c = 0)$, and

$$X_d(z) = z^{g_1+g_2} - \left((1-p)^{g_1} + pz^{g_1} \sum_{i=0}^{g_1-1} \left(\frac{1-p}{z} \right)^i \right) Y(z)^c. \tag{5.11}$$

The reason that we provide an explicit formula for this particular case is that this formula is significantly easier than the formula in the case where $q_i < 1$ for one or more $i = 1, \ldots, g_1$. The stability condition (cf. Algorithm 5.1 in Appendix 5.A) for this example is relatively easy to derive and reads as follows:

$$\begin{cases} \mu c < g_1 + g_2, & \text{if } p = 0, \\ \mu c < g_2, & \text{if } p = 1, \\ \mu c < g_2 + \left(1 - (1-p)^{g_1}\right) \frac{1-p}{p}, & \text{otherwise,} \end{cases}$$

where μ is the mean arrival rate per slot, i.e. $\mu = \mathbb{E}[Y]$. This can be understood as follows: if $p = 0$ there are no turning vehicles and we obtain the regular FCTL queue with green period $g_1 + g_2$. If $p = 1$ all vehicles are turning vehicles and there are no departures during the first part of the green period because $q_i = 1$, so we obtain the FCTL queue with green period g_2. The other case can be understood as follows: on the left-hand side we have the average number of arrivals per cycle whereas on the right-hand side we have the average number of slots available for delayed vehicles to depart. Indeed, on the right-hand side we have g_2, the number of green slots during the second part of the green period which are all available for vehicles to depart, and the number of green slots available for departures during the first green period:

$$\sum_{i=1}^{g_1} (1-p)^i = \left(1 - (1-p)^{g_1}\right) \frac{1-p}{p}.$$

If $p_i = 0$ for all i, i.e. there are no blockages occurring at all (regardless of the q_i), the FCTL queue with multiple lanes (with green period $g = g_1 + g_2$) is obtained. Note that we do not have to include the state S, because there are no blockages of batches of vehicles. The established recursion reduces to the recursion as in Subsection 1.3.1 if $m = 1$ and therefore the steady-state distributions and the PGFs coincide. This can e.g. be observed when putting $p_i = 0$ and $m = 1$ in Equations (5.9),

(5.10), and (5.11). The expression for $X_{g_1+g_2}(z)$ or, alternatively, $X_g(z)$ is (after rewriting):

$$X_g(z) = \frac{(z - Y(z))z^{g-1}\sum_{i=0}^{g-1}\mathbb{P}(X_i = 0)\left(\frac{Y(z)}{z}\right)^{g-i-1}}{z^g - Y(z)^c}, \tag{5.12}$$

where $\mathbb{P}(X_0 = 0)$ is to be interpreted as $\mathbb{P}(X_c = 0)$. For general m, we have the following formula:

$$X_g(z) = \frac{z^{mg}\sum_{i=0}^{g-1}\sum_{l=0}^{m-1}\mathbb{P}(X_i = l)\left(1 - \frac{Y(z)}{z^{m-l}}\right)\left(\frac{Y(z)}{z}\right)^{g-i-1}}{z^{mg} - Y(z)^c}, \tag{5.13}$$

where the $\mathbb{P}(X_0 = l)$, $l = 0, \ldots, m-1$, are to be interpreted as $\mathbb{P}(X_c = l)$. The stability condition for this case can be verified to be

$$\mu c < mg$$

which is in accordance with Algorithm 5.1 described in Appendix 5.A.

It can also be verified that the bFCTL queue reduces to the regular FCTL queue with green time $g = g_2$ and red time $r + g_1$, if $p_i = 1$ and $q_i = 1$.

Remark 5.5 *For the FCTL queue with a single lane and no blockages (i.e. $p_i = 0$ or $p_i = 1$ and $q_i = 1$) there is an alternative characterization of the PGF in terms of a complex contour integral, see Chapter 2. It remains an open question whether such a contour-integral representation exists for the bFCTL with multiple lanes, as the polynomial structure in terms of $Y(z)/z$ as present in Equation* (5.12) *is not present in the general bFCTL queue. This feature of the FCTL queue seems essential to obtain a contour-integral expression as is done in Chapter 2.*

Remark 5.6 *In Chapter 2, a decomposition result is presented in Theorem 2.2. It shows that several related queueing processes can in fact be decomposed in the independent sum of the FCTL queue and some other queueing process. It is likely that the bFCTL queue with multiple lanes allows for some of those generalizations as well (for more details on those models see Subsection 2.3.2). We mention randomness in the green and red time distributions as a relevant potential extension.*

Finding the unknowns in $X_{g_1+g_2}(z)$

As mentioned before, we still need to find several unknowns before the expression for $X_{g_1+g_2}(z)$ is complete. How this is done for the standard FCTL model, is explained in Subsection 1.3.1. The standard framework for the FCTL queue as described before is also applicable to the bFCTL queue with multiple lanes with some minor differences. Although we are dealing with more complex formulas, the key ideas are identical. We have $m(g_1+g_2)+(m-1)g_1$ unknowns in the numerator $X_n(z)$ of $X_{g_1+g_2}(z)$ in Equation (5.8) and we have $m(g_1+g_2)$ roots with $|z| \leq 1$ for the denominator $X_d(z)$ of $X_{g_1+g_2}(z)$, assuming stability of the queueing model. For more details on how to compute the stability condition, we refer the reader to Appendix 5.A. An application of Rouché's theorem, see e.g. [6], shows that $X_d(z)$ indeed has $m(g_1+g_2)$ roots on or within the unit circle assuming stability. One root is $z=1$, which leads to a trivial equation and as a substitute for this root, we put in the additional requirement that $X_{g_1+g_2}(1)=1$. The remaining $(m-1)g_1$ equations are implicitly given in Equations (5.1) and (5.3). We give them here separately for completeness. We have for $k=1,...,m-1$

$$\mathbb{P}(X_1=k,S=b)=p_1q_1\sum_{l=1}^{k}\mathbb{P}(X_c=l)\mathbb{P}(Y_1=k-l)+q_1\mathbb{P}(X_c=0)\mathbb{P}(Y_{1,b}=k),$$

and for $i=2,\ldots,g_1$ and $k=1,\ldots,m-1$

$$\begin{aligned}\mathbb{P}(X_i=k,S=b)=\sum_{l=1}^{k}\left\{p_iq_i\mathbb{P}(X_{i-1}=l,S=u)+q_i\mathbb{P}(X_{i-1}=l,S=b)\right\}\mathbb{P}(Y_i=k-l)\\ +q_i\mathbb{P}(X_{i-1}=0,S=u)\mathbb{P}(Y_{i,b}=k),\end{aligned}$$

which provides us with the $(m-1)g_1$ additional equations. In total, we obtain a set of $m(g_1+g_2)+(m-1)g_1$ linear equations with $m(g_1+g_2)+(m-1)g_1$ unknowns, which we can solve to find the unknown $\mathbb{P}(X_i=l,S=u)$, for $i=1,\ldots,g_1$ and $l=0,\ldots,m-1$, the unknown $\mathbb{P}(X_i=l,S=b)$, for $i=1,\ldots,g_1$ and $l=1,\ldots,m-1$, and the unknown $\mathbb{P}(X_i=l)$, for $i=g_1+1,\ldots,g_1+g_2-1$, $i=c$, and $l=0,\ldots,m-1$. Due to the complicated structure of our formulas, we do not obtain a similar, easy-to-compute Vandermonde system as for the standard FCTL queue (see [206]), but a linear solver is in general able to find the unknowns (we did not encounter any numerical issues/problems in the examples that we studied).

There are several ways to obtain the roots of $X_d(z)$ in Equation (5.8). Because those roots are subsequently used in solving a system of linear equations, we need to find the required roots with a sufficiently high precision, certainly if

$m(g_1+g_2)+(m-1)g_1$ is large. In some cases, Mathematica [221] is able to find the roots analytically (using the function Solve), e.g. in case the number of arrivals per slot has a Poisson or geometric distribution. In other cases, one could use the function NSolve in Mathematica, which is able to compute roots with any precision. There are several alternatives to using functions of Mathematica: such root-finding procedures have in fact attracted quite some attention in the research on similar queueing models. We also discussed an algorithm to find roots of certain equations, see Algorithm 2.1 described in Appendix 2.A, while two other methods, one based on a Fourier series representation and one based on a fixed point iteration, are described in [98].

5.3.2 Performance measures

Now that we have a complete characterization of $X_{g_1+g_2}(z)$, we can find the PGFs of the queue-length distribution at the end of the other slots by employing Equations (5.1) up to (5.7). This basically implies that we can find any type of performance measure related to the queue-length distribution. As an example we find the PGF of the queue-length distribution at the end of an arbitrary slot. We denote this PGF with $X(z)$ and obtain the following expression:

$$X(z) = \frac{1}{c}\sum_{i=1}^{c} X_i(z).$$

Another important performance measure is the delay distribution. The mean of the delay distribution, $\mathbb{E}[D]$, can easily be derived from the mean queue length at the end of an arbitrary slot by means of Little's law with a time-varying arrival rate (for a proof of Little's law in this setting see e.g. [180]):

$$\mathbb{E}[D] = \frac{X'(1)}{\frac{1}{c}\sum_{i=1}^{c} Y_i'(1)}.$$

The PGF of the delay distribution can be derived (as is done for the FCTL queue in [206]), but such a derivation is more difficult. In the regular FCTL queue, the number of slots an arriving car has to wait is deterministic when conditioned on the number of cars in the queue and the time slot in which the car arrives. This is not the case for the bFCTL queue as the occurrence of blockages is random. By proper conditioning on the various blocked slots and queue lengths, one should be able to directly obtain the delay distribution from the distribution of the queue length. We do not pursue this here.

If we want to obtain probabilities and moments from a PGF, we need to differentiate the PGF and respectively put $z = 0$ or $z = 1$. In our experience, this has not proven to be a problem. However, differentiation might become prohibitive in various settings, e.g. when $m(g_1 + g_2) + (m-1)g_1$ becomes large or if we want to obtain tail probabilities. There are ways to circumvent such problems. If we are pursuing probabilities and do not want to rely on differentiation, we might use the algorithm developed by Abate and Whitt in [2] to numerically obtain probabilities from a PGF. For obtaining moments of random variables from a PGF, an algorithm was developed in [51] which finds the first N moments of a PGF numerically. Essentially, this shows that, from the PGF, we can obtain any type of quantity related to the steady-state distribution of the queue length, in the form of a numerical approximation.

5.4 Examples

We investigate the influence of several parameters on the performance measures. We consider performance measures like the mean and variance of the steady-state queue-length distribution, both at specific moments and at the end of an arbitrary slot, the mean delay, and several interesting queue-length probabilities. We start with studying the influence of the p_i and q_i in Subsection 5.4.1. In Subsection 5.4.2, we compare the case of turning and straight-going traffic on a single lane, as present in the bFCTL queue where blockages of all vehicles might occur, and cases where we have dedicated lanes for the right-turning and straight-going traffic where only turning vehicles are blocked. Note that we will consider each lane separately in those examples. Afterwards, we investigate the bFCTL queue with multiple lanes without any blockages, so we study a direct extension of the regular FCTL queue to a model with multiple lanes. We do this in Subsection 5.4.3.

5.4.1 The bFCTL queue with turning vehicles and pedestrians

In this subsection, we study the bFCTL queue with a single lane, so $m = 1$. The setting in this subsection is as depicted in Figure 5.1(b). We mainly focus on the distribution of $X_{g_1+g_2}$, the overflow queue, as this is the distribution from which some interesting performance measures can be derived. This distribution reflects the probability distribution of the queue size at the moment that the green light switches to a red light. We also briefly consider some other performance measures.

Influence of the number of turning vehicles

First, we vary the fraction of right-turning vehicles p_i and study its influence on $X_{g_1+g_2}$. We choose the p_i to be the same for each i, so we have $p_i = p$, and we vary p. We choose the value of the $q_i = q$ to be 1, so there are always pedestrians on the pedestrian crossing during the first part of the green period with length g_1. In this way, we can effectuate the influence of the fraction of turning vehicles on the performance measures. Further, we choose g_1 to be either 2 or 10 and we choose $g_2 = r = 2g_1$. The arrival process is taken to be Poisson with mean 0.395. We display results for $\mathbb{P}(X_{g_1+g_2} \leq j)$ for $j = 0, \ldots, 10$ in Figure 5.3.

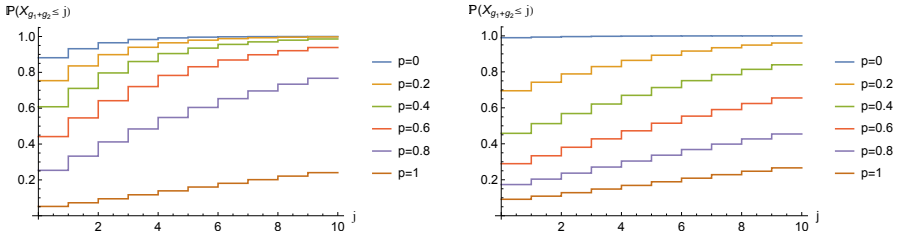
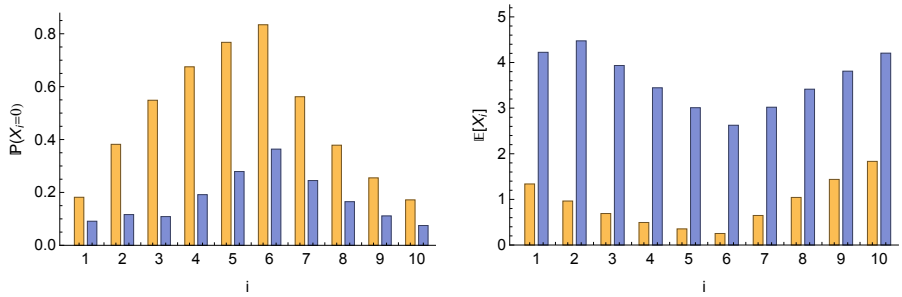


Figure 5.3: Cumulative Distribution Function (CDF) of the overflow queue for various values of $p_i = p$, $q_i = q = 1$, and Poisson arrivals with mean 0.395. In (a) we have $g_2 = r = 2g_1 = 4$ and in (b) we have $g_2 = r = 2g_1 = 20$.

As can be observed from Figure 5.3, the fraction of turning vehicles may dramatically influence the number of queueing vehicles. There is virtually no queue at the end of the green period when there are no turning vehicles ($p = 0$), whereas in more than 70% of the cases there is a queue of at least 10 vehicles at the end of the green period when all vehicles are turning vehicles ($p = 1$). The blockages of the turning vehicles in the latter case effectively reduce the green period by a factor 1/3 in our examples (as $q = 1$), which causes the huge difference in performance. We note that the distribution of $X_{g_1+g_2}$ coincides with the overflow queue distribution in the FCTL queue when $p = 0$ (when we take $g_1 + g_2$ as the green period and r as the red period in the FCTL queue) and when $p = 1$ and $q = 1$ (with g_2 the green period and $r + g_1$ the red period).

When comparing Figures 5.3(a) and 5.3(b), we see that the influence of p is not uniform across the two examples. In case $p = 0$ or $p = 1$, the probability of a large overflow queue is larger for the case where $g_1 = 2$. This might be clarified by noting that a larger cycle reduces the amount of within-cycle variance which

reduces the probabilities of a large queue length. If $0 < p < 1$ this does not seem to be the case. This might be due to the fact that a relatively big part of the first green period is eaten away by turning vehicles that are blocked when $g_1 = 10$. For example, when $p > 0$ and the first vehicle is a turning vehicle, immediately the entire period g_1 is wasted because $q = 1$. This is of course also the case when $g_1 = 2$, but the blockage is resolved sooner and during the second part of the green period the blocked vehicle may depart relatively soon in comparison with the case where $g_1 = 10$.



(a) (b)

Figure 5.4: In (a) $\mathbb{P}(X_i = 0)$ for slot number $i = 1,\ldots,10$ is displayed for two different values of p_i, where orange corresponds to $p_i = p = 0$ and blue to $p_i = p = 0.6$, with $2g_1 = g_2 = r = 4$, $q_i = q = 1$, and with Poisson arrivals with mean 0.395. In (b) the same two examples are studied, but the mean queue length $\mathbb{E}[X_i]$ at the end of slot i is shown.

In Figure 5.4(a), we see the probability of an empty queue after slot i, where $i = 1,2,\ldots,c$, for two different values of p. For the case $p = 0$ (in orange) we have a monotone increasing sequence of probabilities during the green period as one would expect: this setup corresponds to a regular FCTL queue and once the queue empties during the green period, it stays empty. We see that for the case $p = 0.6$ (in blue) the probabilities of an empty queue after slot i are much lower (as there are more turning vehicles which might be blocked and hence cause the queue to be non-empty). In fact, the probability of an empty queue even decreases when going from slot 2 to slot 3. This can be clarified by the fact that the queue might start building again even when the queue is (almost) empty: e.g. if the queue is empty during the first green period and there is an arrival of a turning vehicle, that vehicle will be blocked as $q = 1$ in which case the queue is no longer empty.

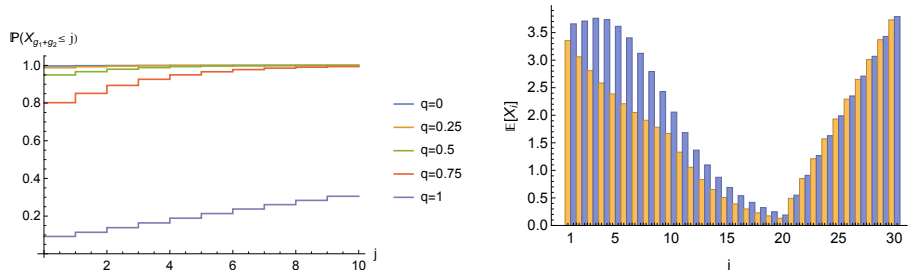
The same type of behavior is reflected in the mean queue length at the end of a slot, as can be observed in Figure 5.4(b). Even though the green period already started, the queue in the example with $p = 0.6$ still grows (in expected value) during the first part of the green period. This is caused by the fact that vehicles might be blocked, which demonstrates the possibly severe impact of blocked vehicles on the performance of the system.

Influence of the pedestrians

Secondly, we investigate the influence of the presence of pedestrians by studying various values for the q_i. A high value of the q_i corresponds to a high density of pedestrians as q_i corresponds to the probability that a turning vehicle is not allowed to depart during the first green period. Conversely, a low value of the q_i corresponds to a low density of pedestrians and a relatively high probability of a turning vehicle departing during the first green period. We choose $p_i = p = 0.5$ and take $g_1 = g_2 = r = 10$. We take Poisson arrivals with mean 0.36. We study one set of examples where the q_i are constant over the various slots, see Figure 5.5(a). We also study the influence of the dependence of the q_i on i by investigating two cases with all parameters as before in Figure 5.5(b). In one case we take $q_i = 0.5$ for all i, but in the other case we take $q_i = 1 - (i-1)/g_1$. The latter case reflects a decreasing number of pedestrians blocking the turning flow of vehicles during the first part of the green period.

We note that it is important to get the right q_i if one wants to investigate the queue-length distribution in front of the traffic light, as the q_i have an impact on the performance measures. In Figure 5.5(a), we clearly see that the more pedestrians, the longer the queue length at the end of the green period is. Indeed, if there are more pedestrians, there are relatively many blockages of vehicles which subsequently causes the queue to be relatively large.

Moreover, it is important to capture the dependence of the q_i on the slot i in the right way, see Figure 5.5(b). Even though, on average over all slots, the mean number of pedestrians present is similar in the two cases, we see a clear difference between the two examples. In the case with decreasing q_i (in blue), we see an initial increase of the mean queue length during the first green slots of the cycle, caused by a relatively large fraction of turning vehicles ($p = 0.5$) *and* a high value of q_i. This is not the case in the other example where $q_i = 0.5$ for all i. After some slots of the first green period, the decrease in the mean queue length is quicker for the example where the q_i decrease when i increases, which can (at least partly) be explained by the decreasing q_i. There might thus be an influence of the q_i when focusing on the first part of the green period. During



(a) (b)

Figure 5.5: In (a) the CDF of the overflow queue is displayed for various values of the q_i with all $q_i = q$ the same, $p_i = p = 0.5$, Poisson arrivals with mean 0.36, and $g_1 = g_2 = r = 10$. In (b) the $\mathbb{E}[X_i]$ are compared for slot number $i = 1,\ldots,30$ with in orange $q_i = 0.5$ and in blue $q_i = 1-(i-1)/g_1$ for $i = 1,\ldots,g_1$. Further, it is assumed that $p_i = p = 0.5$, that the number of arrivals in each slot follows a Poisson distribution with mean 0.36, and that $g_1 = g_2 = r = 10$.

the remaining part of the cycle, the queue in front of the traffic light behaves more or less the same in both examples and even the mean overflow queue, $\mathbb{E}[X_{g_1+g_2}]$, is not that much different for the two examples. This implies, as can also be observed in Figure 5.5(b), that the mean queue length during the red period is comparable as well for our setting. This does not hold for the mean queue length at the end of an arbitrary slot and the mean delay, because of the differences in the queue length during the first part of the green period.

5.4.2 Shared right-turn lanes and dedicated lanes

We continue with a study of several numerical examples that focus on the differences between shared right-turn lanes and dedicated lanes for turning traffic. We do so in order to provide relevant insights in the benefit of splitting the vehicles in different streams. Firstly, we study the difference between a single shared right-turn lane (as visualized in Figure 5.6(a)) and a case where the straight-going and turning vehicles are spread over two different lanes. In the latter case, we thus have two lanes, one for the straight-going traffic and one for the turning traffic (as visualized in Figure 5.6(b)) which we can analyze as two *separate* bFCTL queues.

Secondly, we compare two two-lane settings. The first is visualized in Fig-

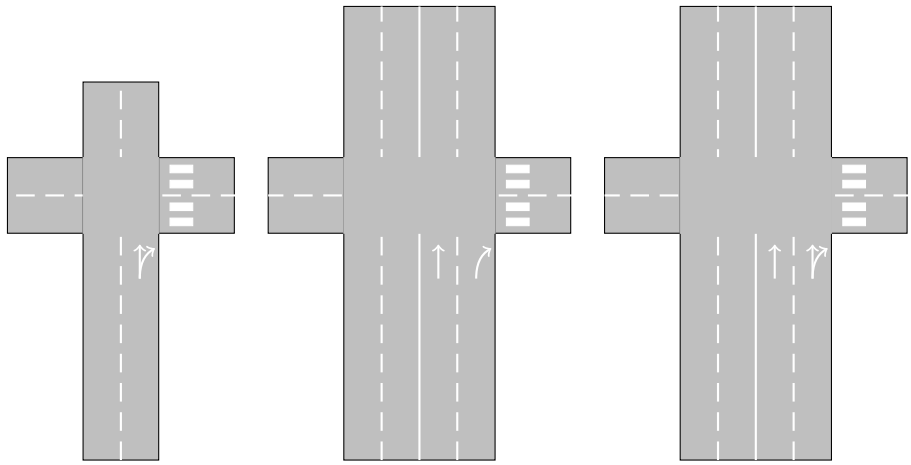


Figure 5.6: The various lane configurations considered in Subsection 5.4.2. In (a) we have a single lane with a shared right-turn lane. In (b) we have two dedicated lanes: one for straight-going vehicles and one for right-turning traffic, whereas in (c) we have a two-lane setup with one lane for straight-going vehicles only and a shared right turn.

ure 5.6(b), while the other is a two-lane scenario where one lane is a dedicated lane for straight-going traffic and the other is a shared right-turn lane as depicted in Figure 5.6(c). We thus allow for straight-going traffic to mix with the right-turning vehicles in the latter case. We do so in order to make sure that the shared right-turn lane together with the lane for vehicles heading straight has the same capacity as the two lanes where the two streams of vehicles are split (as opposed to the first example in this subsection, where there is a difference in capacity between the two cases). In both two-lane scenarios we, again, analyze the two lanes as two separate bFCTL queues.

One lane for the shared right-turn

We start with comparing the traffic performance of a single shared right-turn lane as in Figure 5.6(a), case (1), and a two-lane scenario where the turning vehicles and the straight-going vehicles are split as in Figure 5.6(b), case (2). We refer in the latter case to the lane which has right-turning vehicles as lane 1 and to the other lane we refer as lane 2. We assume that the arrival process

is Poisson and that the arrival rate of turning vehicles, μ_1, and straight-going vehicles, μ_2, are the same in both cases. The total arrival rate of vehicles is $\mu = \mu_1 + \mu_2$ in case (1). We choose $p_i = 0.3$ for the shared right-turn lane, whereas in the two-lane case we have $p_i = 1$ for lane 1 and $p_i = 0$ for lane 2 and arrival rates $\mu_1 = 0.3\mu$ at lane 1 and $\mu_2 = 0.7\mu$ at lane 2. Further, we choose $q_i = 1$, $g_1 = 8$, $g_2 = 20$, and $r = 20$. We compute the mean queue length at the end of an arbitrary time slot for both lanes in case (2), denoted with $\mathbb{E}[X^{(i)}]$ for lane i, and the total mean queue length at the end of an arbitrary time slot, denoted with $\mathbb{E}[X^t]$, and which equals $\mathbb{E}[X^{(1)}] + \mathbb{E}[X^{(2)}]$. For case (1) we denote the mean queue length at the end of an arbitrary time slot with $\mathbb{E}[X^t]$. The delay of an arbitrary car is denoted with $\mathbb{E}[D]$ for both cases (1) and (2). We study various values of μ in Table 5.1.

Table 5.1: The total Poisson arrival rate, μ, the mean queue length at the end of an arbitrary time slot, $\mathbb{E}[X^t]$, and the mean delay, $\mathbb{E}[D]$, for case (1), and for case (2) the mean queue length at the end of an arbitrary time slot at lanes 1 and 2, $\mathbb{E}[X^{(1)}]$ and $\mathbb{E}[X^{(2)}]$ respectively, the total mean queue length at the end of an arbitrary time slot, $\mathbb{E}[X^t]$, and the mean delay of an arbitrary car, $\mathbb{E}[D]$.

	Case (1)		Case (2)			
μ	$\mathbb{E}[X^t]$	$\mathbb{E}[D]$	$\mathbb{E}[X^{(1)}]$	$\mathbb{E}[X^{(2)}]$	$\mathbb{E}[X^t]$	$\mathbb{E}[D]$
0.08	0.542	6.771	0.208	0.260	0.468	5.855
0.16	1.262	7.889	0.427	0.555	0.982	6.140
0.24	2.179	9.080	0.658	0.892	1.550	6.458
0.32	3.451	10.79	0.902	1.280	2.182	6.818
0.40	6.496	16.23	1.159	1.733	2.892	7.230

In Table 5.1, we can clearly see that the total mean queue length at the two lanes in case (2) is lower than the mean queue length at the single lane in case (1). This makes sense from various points of view: in case (2), we have twice as much capacity as in case (1), so we would expect a smaller total mean queue length in case (2). Moreover, in case (1), it might happen that straight-going vehicles are blocked. Such blockages cannot occur in case (2), as all turning traffic is on lane 1 and all vehicles that go straight are on lane 2. These two reasons are the main drivers for the performance difference between cases (1) and (2). From the point of view of the traffic performance, it thus makes sense to split the traffic on a shared right-turn lane into two separate streams of vehicles on two lanes. Note that this holds while assuming that the capacity for the two

separate lanes is twice as large as for the single-lane case. We observe similar results when looking at the mean delay and comparing cases (1) and (2).

Two lanes for the shared right-turn

Now we turn to an example where we still have two dedicated lanes as in case (2) of the previous example, one for turning traffic and one for straight-going traffic, see Figure 5.6(b), but we compare it with a two-lane example where the vehicles mix, see Figure 5.6(c). All turning vehicles will be on lane 1, but we also allow some straight-going traffic to be present on lane 1 too. Lane 1 is thus a shared right-turn lane. On lane 2, we only have vehicles that are heading straight. In order to make a comparison that is as fair as possible we assume the following: the total arrival rate and the fraction of turning vehicles are the same.

We assume that the probability that an arbitrary vehicle is a turning vehicle is 0.3 and we vary the total Poisson arrival rate μ to study the influence of the strict splitting of the turning vehicles. In case (1), we thus have an arrival rate at the right-turning lane that satisfies $\mu_1 = 0.3\mu$, whereas on the other lane we have an arrival rate $\mu_2 = 0.7\mu$. At lane 1 we have $p_i = 1$ and at lane 2 we have $p_i = 0$. In case (2) we distinguish between two subcases. In subcase (2a) we assume that the total arrival rate at both lanes is the same and thus $\mu_1 = \mu_2 = 0.5\mu$. In subcase (2b), we assume that the arrival rate is split in the ratio $2:3$, so $\mu_1 = 0.4\mu$ and $\mu_2 = 0.6\mu$. This implies that in subcase (2a) we choose $p_i = 0.6$ (the fraction of turning vehicles is then $p\mu_1 = 0.6 \cdot 0.5\mu = 0.3\mu$) and in subcase (2b) we choose $p_i = 0.75$ (the fraction of turning vehicles is then $p\mu_1 = 0.75 \cdot 0.4\mu = 0.3\mu$), to make sure that we match the number of turning vehicles in case (1). Further, we choose $q_i = 1$, $g_1 = 8$, $g_2 = 16$, and $r = 16$. Then, we study the mean queue length at the end of an arbitrary time slot of both lanes, $\mathbb{E}[X^{(1)}]$ and $\mathbb{E}[X^{(2)}]$, and the total average mean queue length at the end of an arbitrary time slot, denoted with $\mathbb{E}[X^t]$. We obtain Table 5.2.

In Table 5.2, we see only small differences in the total mean queue lengths at the end of an arbitrary time slot for low arrival rates. At both lanes, there are few vehicles in the queue. This is different for the examples in Table 5.2 with a higher arrival rate. In all examples for case (1) we see that the mean queue length at lane 2, the straight-going traffic lane, is higher than for lane 1. This is due to the relatively high fraction of vehicles that *have* to use lane 2 due to the strict splitting between turning and straight-going vehicles. In some sense, lane 1, which only has turning vehicles, has overcapacity that cannot be used for the busier lane 2 with only straight-going traffic. This is different for the other two

Table 5.2: The total Poisson arrival rate, μ, the mean queue length at the end of an arbitrary time slot at lanes 1 and 2, $\mathbb{E}[X^{(1)}]$ and $\mathbb{E}[X^{(2)}]$, and the total mean queue length at the end of an arbitrary time slot, $\mathbb{E}[X^t]$, for cases (1), (2a) and (2b).

	Case (1)			Case (2a)			Case (2b)		
μ	$\mathbb{E}[X^{(1)}]$	$\mathbb{E}[X^{(2)}]$	$\mathbb{E}[X^t]$	$\mathbb{E}[X^{(1)}]$	$\mathbb{E}[X^{(2)}]$	$\mathbb{E}[X^t]$	$\mathbb{E}[X^{(1)}]$	$\mathbb{E}[X^{(2)}]$	$\mathbb{E}[X^t]$
0.08	0.185	0.202	0.387	0.258	0.142	0.400	0.221	0.172	0.393
0.16	0.379	0.432	0.811	0.558	0.297	0.855	0.467	0.363	0.831
0.24	0.584	0.695	1.278	0.899	0.467	1.367	0.738	0.578	1.315
0.32	0.800	0.998	1.798	1.281	0.655	1.936	1.033	0.819	1.852
0.40	1.028	1.353	2.381	1.711	0.863	2.573	1.353	1.094	2.447
0.48	1.270	1.775	3.045	2.206	1.094	3.299	1.704	1.408	3.113
0.56	1.527	2.301	3.829	2.821	1.353	4.173	2.094	1.775	3.869
0.64	1.802	3.037	4.839	3.718	1.646	5.364	2.546	2.217	4.763
0.72	2.100	4.366	6.465	5.541	1.984	7.525	3.112	2.793	5.905
0.80	2.430	8.878	11.31	15.13	2.390	17.52	3.928	3.669	7.597

cases, where the traffic is split more evenly across the two lanes. As one would expect, the longest queue in subcase (2a) is present at lane 1, as the arrival rate at both lanes is the same and because vehicles are only blocked at lane 1, the shared right-turn lane. This points towards another potential improvement and this is found in subcase (2b) where we balance the arrival rate differently, which leads to a more economic use of both lanes and, hence, also the best performance in this example when looking at $\mathbb{E}[X^t]$.

The results in Tables 5.1 and 5.2 might seem conflicting at a first glance, but they are not. In the case of a single, shared right-turn lane as in Table 5.1, we see a higher mean queue length than for the two dedicated lanes case in Table 5.1. This is the other way around in Table 5.2 (considering case (2b)). This is mainly explained by the fact that in case (2b) in Table 5.2, we have two lanes and thus twice as much capacity as in case (1) in Table 5.1. This is one of the main factors in the explanation of the differences in the mean performance between the examples studied in Tables 5.1 and 5.2.

The two examples in this subsection tell us that a separate or dedicated lane for turning traffic does not necessarily improve the traffic flow. An in-depth study is needed to obtain the best layout of the intersection and the best traffic-light control. As a side-remark, we surpass the possibility here that in Table 5.2, case (1), we might control the two lanes in a different way, e.g. by prolonging the green period for one of the lanes. This is not possible in cases (2a) and

(2b). This is also something one should take into account when looking for good traffic-light settings.

5.4.3 FCTL queue with multiple lanes

The regular FCTL queue has only a single lane from which vehicles might depart, yet at bigger intersections, this is not realistic. There might be several lanes for, e.g., straight-going traffic which all receive green simultaneously. For a visualization, see Figure 5.1(a). Our framework for the bFCTL queue with multiple lanes allows us to model such examples, which we demonstrate in this subsection. We study both the case of a Poisson distributed number of arrivals and the case of a geometrically distributed number of arrivals studied in [206]. We thus study a case where $g = g_1 + g_2 = 5$, $p_i = 0$ for all i, $r = 5$, and with Poisson or geometrically distributed arrivals in each slot with mean μ. We study various cases of μ and analyze the overflow queue, denoted with X_g, the mean queue length at the end of an arbitrary time slot $\mathbb{E}[X^t]$, and the mean delay $\mathbb{E}[D]$. We also vary m to study the influence of having multiple lanes in the FCTL queue. In order to make a comparison between the various cases with different m, we scale the arrival rate proportionally with m so that the load or vehicle-to-capacity ratio, $\rho = (c\mu)/(mg)$, is fixed for different values of m. Then, we obtain Tables 5.3 (for Poisson arrivals) and 5.4 (for geometrically distributed arrivals).

We note that there is a difference between analyzing m FCTL queues separately and the joint analysis of the m lanes as presented here. It is thus important to perform an analysis that accounts for the number of lanes that vehicles from a single stream can use. This can most prominently be observed by fixing ρ and considering various values of m: the mean and variance of the overflow queue (measured in number of vehicles) then decrease if we have Poisson arrivals in each slot. This is not the case for some examples with geometrically distributed arrivals if ρ is sufficiently high. However, when taking into account that vehicles are spread out over the different lanes, the physical length of the queue still decreases. The different behavior probably relates to the geometric distribution being more variable than the Poisson distribution. When m increases, the squared coefficient of variation for the number of arrivals per slot is decreasing for the Poisson case and increasing for the geometric case, which probably is (part of) the explanation for the observed behavior. Indeed, a larger variability in the number of arrivals in general tends to lead to an increase in the mean steady-state queue length in queueing models. This indicates that having more lanes at a single intersection while ρ is fixed, is not necessarily beneficial when looking at the total number of vehicles in the queue: a high variability

Table 5.3: The bFCTL queue with m lanes, $g = 5$, $r = 5$, Poisson arrivals, and no blockages. The load ρ, the number of lanes m, the mean arrival rate μ, and several performance measures are displayed.

ρ	m	μ	$\mathbb{E}[X_g]$	$\mathrm{Var}[X_g]$	$\mathbb{P}(X_g \geq 10)$	$\mathbb{E}[X^t]$	$\mathbb{E}[D]$
0.2	1	0.1	0.000583	0.000788	< 0.00001	0.170	1.701
	2	0.2	< 0.00001	0.000010	< 0.00001	0.317	1.587
	5	0.5	< 0.00001	< 0.00001	< 0.00001	0.762	1.523
	10	1.0	< 0.00001	< 0.00001	< 0.00001	1.505	1.505
	15	1.5	< 0.00001	< 0.00001	< 0.00001	2.252	1.502
	20	2.0	< 0.00001	< 0.00001	< 0.00001	3.001	1.500
0.4	1	0.2	0.0217	0.0384	< 0.00001	0.404	2.021
	2	0.4	0.00324	0.00663	< 0.00001	0.711	1.778
	5	1.0	0.000013	0.000033	< 0.00001	1.661	1.661
	10	2.0	< 0.00001	< 0.00001	< 0.00001	3.240	1.620
	15	3.0	< 0.00001	< 0.00001	< 0.00001	4.816	1.605
	20	4.0	< 0.00001	< 0.00001	< 0.00001	6.390	1.598
0.6	1	0.3	0.180	0.429	0.000029	0.817	2.724
	2	0.6	0.0770	0.215	0.000019	1.279	2.131
	5	1.5	0.00788	0.0298	< 0.00001	2.834	1.890
	10	3.0	0.00019	0.00101	< 0.00001	5.505	1.835
	15	4.5	< 0.00001	0.000030	< 0.00001	8.181	1.818
	20	6.0	< 0.00001	< 0.00001	< 0.00001	10.85	1.809
0.8	1	0.4	1.097	4.181	0.00842	2.025	5.063
	2	0.8	0.795	3.465	0.00662	2.598	3.247
	5	2.0	0.359	2.038	0.00417	4.707	2.354
	10	4.0	0.109	0.836	0.00242	8.621	2.155
	15	6.0	0.0343	0.332	0.00130	12.68	2.113
	20	8.0	0.0109	0.127	0.00057	16.79	2.099
0.98	1	0.49	23.22	614.8	0.638	24.44	49.88
	2	0.98	22.59	613.1	0.621	25.02	25.53
	5	2.45	21.02	606.9	0.580	27.06	11.04
	10	4.90	18.47	589.0	0.517	30.51	6.227
	15	7.35	15.90	558.9	0.451	33.93	4.616
	20	9.80	13.45	517.4	0.381	37.44	3.820

Table 5.4: The bFCTL queue with m lanes, $g = 5$, $r = 5$, geometrically distributed arrivals, and no blockages. The load ρ, the number of lanes m, the mean arrival rate μ, and several performance measures are displayed.

ρ	m	μ	$\mathbb{E}[X_g]$	$\mathrm{Var}[X_g]$	$\mathbb{P}(X_g \geq 10)$	$\mathbb{E}[X^t]$	$\mathbb{E}[D]$
0.2	1	0.1	0.00135	0.00210	< 0.00001	0.174	1.736
	2	0.2	0.000098	0.00019	< 0.00001	0.323	1.614
	5	0.5	< 0.00001	< 0.00001	< 0.00001	0.773	1.547
	10	1.0	< 0.00001	< 0.00001	< 0.00001	1.526	1.526
	15	1.5	< 0.00001	< 0.00001	< 0.00001	2.279	1.519
	20	2.0	< 0.00001	< 0.00001	< 0.00001	3.032	1.516
0.4	1	0.2	0.0407	0.0903	< 0.00001	0.432	2.158
	2	0.4	0.0176	0.0532	< 0.00001	0.749	1.874
	5	1.0	0.00551	0.0292	0.000049	1.736	1.736
	10	2.0	0.00292	0.0263	0.000085	3.388	1.694
	15	3.0	0.00239	0.0305	0.000096	5.040	1.680
	20	4.0	0.00226	0.0371	0.000098	6.691	1.673
0.6	1	0.3	0.300	0.951	0.000469	0.949	3.163
	2	0.6	0.245	1.100	0.00147	1.486	2.477
	5	1.5	0.224	1.859	0.00602	3.200	2.133
	10	3.0	0.261	3.812	0.0111	6.106	2.035
	15	4.5	0.313	6.544	0.0134	9.022	2.005
	20	6.0	0.369	10.04	0.0144	11.94	1.990
0.8	1	0.4	1.709	9.176	0.0323	2.646	6.615
	2	0.8	1.890	14.40	0.0549	3.726	4.657
	5	2.0	2.633	37.43	0.109	7.129	3.564
	10	4.0	3.982	100.1	0.151	12.89	3.223
	15	6.0	5.358	193.2	0.167	18.68	3.113
	20	8.0	6.741	316.7	0.174	24.47	3.059
0.98	1	0.49	34.93	$1.38 \cdot 10^4$	0.728	36.15	73.78
	2	0.98	45.83	$2.44 \cdot 10^4$	0.765	48.26	49.24
	5	2.45	78.64	$7.44 \cdot 10^4$	0.814	84.71	34.57
	10	4.90	133.4	$2.18 \cdot 10^5$	0.839	145.5	29.70
	15	7.35	188.2	$4.37 \cdot 10^5$	0.846	206.3	28.07
	20	9.80	242.9	$7.31 \cdot 10^5$	0.849	267.1	27.26

in the number of arrivals per slot might result in an increase of the number of vehicles in the queue when the number of lanes is increased. However, in all cases the mean delay decreases if ρ is fixed and m increases.

5.5 Conclusion

In this chapter, we have established a recursion for the PGFs of the queue-length distribution at the end of each slot which can be used to provide a full queue-length analysis of the bFCTL queue with multiple lanes. This is an extension of the regular FCTL queue so that we can account for temporal blockages of vehicles receiving a green light, for example because of a crossing pedestrian at the turning lane or because of a (separate) bike lane, and to account for a vehicle stream that is spread over multiple lanes. These features might impact the traffic-light performance as we have shown by means of various numerical examples. The blocking of turning vehicles and the number of lanes corresponding to a vehicle stream therefore has to be taken into account when choosing the settings for a traffic light.

We briefly touched upon how one should design the layout of an intersection. Interestingly, it might be suboptimal to have a dedicated lane for turning traffic. It seems that mixing turning and straight-going traffic has benefits over a strict separation of those two traffic streams when there are two lanes for this turning and straight-going traffic. We advocate a further investigation into the influence of separating or mixing different streams of vehicles in front of traffic lights. It might be possible to find the optimal division of straight-going and turning vehicles over the various lanes, e.g. by enumerating several possibilities. A more structured optimization seems difficult because of the intricate expressions involved, but would definitely be worthwhile to investigate. Some research on the splitting of different traffic streams has already been done in e.g. [109, 187, 223, 229] and the present study can be seen as an alternative way of modeling the situation at hand.

The work in [96], in which a simulation study of a similar model is performed, has been our source of inspiration for the study in this chapter. There are some extensions possible when comparing our work with [96]. We e.g. did not study the influence of start-up delays as is done in [96]. Investigating such start-up delays at the beginning of the green period is easily done in our framework: we simply need to adjust the Y_i for the first few slots. Another approach to deal with start-up delays is presented in [132]. Start-up delays which depend on the blocking of vehicles and different slot lengths for different combinations

of turning/straight going vehicles, are harder to tackle. One could e.g. introduce additional states (besides states u and b) to deal with this. Although the developed recursion does not directly allow for such a generalization, it seems possible to account for this at the expense of a more complex recursion. For the ease of exposition, we have refrained from doing so and we leave a full study on this topic for future research.

We are able to provide an exact calculation for numerous performance measures by means of our queueing-theoretic approach, which before often either had to be approximated or simulated. The authors in [44] indicate that the stochastic behavior of the model needs to be taken into account and our framework allows us to do so. Our results indicate that the stochastic behavior indeed plays a role in various performance measures.

A possible extension of the results on the bFCTL queue is a study of (the PGF of) the delay distribution. We have refrained from deriving the delay distribution because of its (notational) complexity. Using proper conditioning, one should be able to obtain (the PGF of) the delay distribution for the bFCTL queue.

The bFCTL queue calls for further generalizations. For example, instead of two full lanes, e.g. one for straight-going and one for turning traffic, we could also consider a single lane which splits into two lanes just before the intersection in such a way that some, say N, turning vehicles may accumulate on a separate lane. Such a small separate lane is often referred to as a turning bay. The N vehicles on the turning bay would not block straight-going traffic in any way (because they are on a separate lane), but if there would be N vehicles at the turning bay and another turning vehicle arrives, also the vehicles heading straight will be blocked. It would be interesting to study such a model and gain insight into the benefits of such a turning bay.

Another topic for future research is to modify the bFCTL queue in such a way that it enables a *joint* analysis of two lanes with one dedicated lane for vehicles heading straight and one shared lane with both turning and straight-going vehicles. Such a case is not covered by the bFCTL with multiple lanes, as it seems that in this extension one needs to take into account how many vehicles of both types (i.e. turning and straight-going vehicles) there are. This might lead to a two-dimensional queueing model rather than the one-dimensional one considered in the current chapter.

A further possible extension of the bFCTL queue would be to consider more realistic blocking behaviors: instead of e.g. a fixed probability q_i for each slot i, a more general blocking process might be considered. For example, if there are no pedestrians during slot i for the model depicted in Figure 5.1(b), then the probability that there are also no pedestrians in slot $i+1$, might be rela-

tively high. In other words, there might be *dependence* between the various slots when considering the presence of pedestrians. It is worthwhile to investigate generalizations of the blocking process in order to further increase the general applicability of the bFCTL queue with multiple lanes.

Another generalization for the blocking mechanism, is to block only a *part* of the m vehicles that are at the head of the queue. Indeed, we restrict ourselves to the cases where either all vehicles in a batch of size m are blocked (or not). In various real-life examples, it might be the case that only part of the m vehicles are blocked. It would be interesting to investigate whether such a model can be analyzed.

Finally, we also advocate an investigation whether the bFCTL queue with a vehicle-actuated mechanism (rather than the fixed green and red times that we consider) results in a tractable model. Although perhaps not directly visible in the derivation, we use the fact that the traffic lights have a fixed setting to our advantage. This namely implies that we can study each queue in isolation, which then enables us to turn the bFCTL queue with multiple lanes into a model with one dimension. This does not seem to be possible in an actuated setting, because the green time in this cycle depends on all queue lengths in the previous cycle and if we have n vehicle streams, we would have to study an n-dimensional PGF. Currently, a fully exact analysis seems to be beyond reach. If we could find the PGF for an actuated bFCTL type of model, it would perhaps provide a way to study other n-dimensional queueing models as well. A study whether this extension is tractable is therefore interesting, both in view of the bFCTL queue with multiple lanes and in view of queueing theory more generally.

Appendix

5.A Stability condition for the bFCTL queue

In this section, we formulate an algorithm to check whether the bFCTL queue renders a stable queueing model. In order to derive the stability condition, we note that the mean number of departures of delayed vehicles per cycle should be larger than the mean number of arriving vehicles per cycle. The latter is easy to compute: it is simply $\sum_{i=1}^{c} \mathbb{E}[Y_i]$, the sum of the mean number of arrivals per slot. It is more difficult to derive the mean number of delayed vehicles departing from the queue during a cycle. A Markov reward model is one way to obtain this mean number. The Markov chain with the associated transition probabilities that we use is depicted in Figure 5.7. The states and transitions of this Markov chain are similar to those of the Markov chain in Section 5.3, but now we are no longer interested in the queue length. Instead, we are interested in the number of departures of delayed vehicles in each time slot. For this reason, this Markov chain only has states (i,s) for $i = 1,\ldots,g_1$ and $s = u,b$, and states i for $i = g_1+1,\ldots,g_1+g_2+r$. Finally, we create an artificial state 0 to gather the rewards from states $(1,b)$ and $(1,u)$. The long-term mean number of departures of delayed vehicles can now be determined through a reward structure.

The rewards that we assign to each transition are as follows: if we make a transition to a state (i,u) for $i = 1,\ldots,g_1$, we receive a reward m reflecting the maximum of m delayed vehicles departing from the queue. We also get a reward m if we make a transition from state g_1+i to state g_1+i+1 for $i = 1,\ldots,g_2-1$. For all other transitions, we receive no reward. We denote the received reward up to state (i,s) with $r_{i,s}$ with $i = 1,\ldots,g_1$ and $s = u,b$ and the received reward

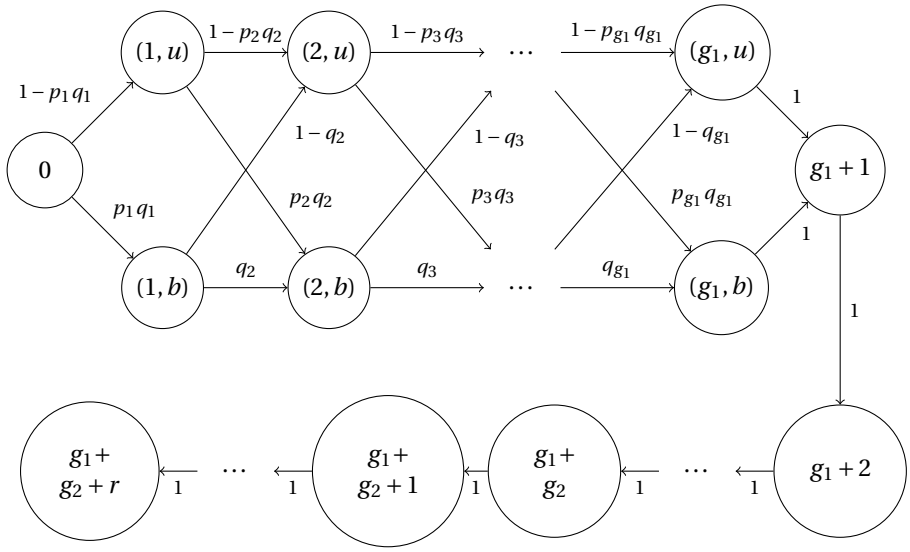


Figure 5.7: Markov chain used to study the stability condition.

up to state i with r_i for $i = 0$ and $i = g_1+1,\ldots,g_1+g_2+r$. Then, if we traverse the Markov chain once, we get the following relations between the rewards in the various states. We work backwards from state g_1+g_2+r to obtain the reward in state 0 (as usual in Markov reward theory). We start with defining the total reward in state g_1+g_2+r to be 0 (there are no vehicle departures while being in state g_1+g_2+r), i.e.

$$r_{g_1+g_2+r} = 0. \tag{5.14}$$

For states $i = g_1+g_2,\ldots,g_1+g_2+r-1$, we obtain

$$r_i = r_{i+1}, \tag{5.15}$$

as there are no departures during the red period. However, for states $i = g_1+1,\ldots,g_1+g_2-1$, we have

$$r_i = m + r_{i+1} \tag{5.16}$$

as there are (potentially) m departures of delayed vehicles. For state (g_1, b) we have that

$$r_{g_1,b} = r_{g_1+1}, \tag{5.17}$$

as there are no departures when the vehicles are blocked. For state (g_1, u) we obtain

$$r_{g_1,u} = m + r_{g_1+1} \tag{5.18}$$

as there are, at most, m delayed vehicles departing from the queue when we make a transition from state (g_1, u) to $g_1 + 1$. Similarly, for states (i, b) with $i = 1, \ldots, g_1 - 1$, we get

$$r_{i,b} = q_{i+1} r_{i+1,b} + (1 - q_{i+1}) r_{i+1,u} \tag{5.19}$$

and for states (i, u) with $i = 1, \ldots, g_1 - 1$, we get

$$r_{i,u} = m + p_{i+1} q_{i+1} r_{i+1,b} + (1 - p_{i+1} q_{i+1}) r_{i+1,u}. \tag{5.20}$$

Finally, for state 0, we get

$$r_0 = p_1 q_1 r_{1,b} + (1 - p_1 q_1) r_{1,u}. \tag{5.21}$$

Then we have that r_0 is the average reward received when traversing the Markov chain as depicted in Figure 5.7, where the average reward translates to the mean number of delayed vehicles that are able to depart from the queue during a cycle. Together with the mean number of vehicles arriving per cycle, we can check whether or not a specific choice of input parameters results in a stable bFCTL queue. The procedure is summarized in Algorithm 5.1.

Algorithm 5.1 Algorithm to check for stability of the bFCTL queue.

1: Input: μ, g_1, g_2, c, p_i for $i = 1, \ldots, g_1$, and q_i for $i = 1, \ldots, g_1$.
2: Use Equations (5.14) up to (5.21) to determine r_0.
3: **if** $\sum_{i=1}^{c} \mathbb{E}[Y_i] < r_0$ **then**
4: The bFCTL queue is stable.
5: **else**
6: The bFCTL queue is not stable.
7: **end if**

Chapter 6

A novel approximation scheme for multidimensional queueing models

6.1 Introduction

Traffic lights with a vehicle-actuated control strategy relate to queueing models with multiple queues, as we argued in Subsections 1.2.2 and 1.3.2. Such queueing models with multiple dimensions are notoriously difficult to analyze. The complicated interactions within such queueing models typically do not allow for a general approach to obtain exact results for performance measures. Even approximation schemes that apply to a large set of multidimensional queueing models are often difficult to establish. The need to obtain performance measures for traffic lights with a vehicle-actuated control strategy has motivated us to develop an approximation procedure. This procedure appears to be applicable to a quite large class of multidimensional queueing models. This chapter is devoted to that procedure.

Multidimensional queueing models arise in many situations and applications. Canonical examples include queues that can be modeled as a random walk in the positive quadrant (for two dimensions), fork-join queues, polling models, and queueing networks. The area of application of such models is very wide and ranges from e.g. production systems, applications in the medical area,

traffic engineering, and numerous applications in communication and computer (supporting) systems, see e.g. the review paper [18]. A sound understanding of the underlying queueing models for each of the aforementioned application areas could lead to service level improvements and/or cost reductions. The method in this chapter contributes to this.

In recent years, only few papers have appeared with an exact analysis for multidimensional queueing models. Of course, there have been some advances in the realm of methodological and computational aspects, but those are mostly for quite specific queueing models. Explicit results for performance measures are still mostly lacking. One exception is formed by so-called product-form networks, which allow for an explicit characterization of the joint queue-length distribution, like Jackson and Kelly networks [97, 104]. For a general treatment see [26, Chapters 1-7], and for recent advances in the general applicability of product forms for multidimensional queueing networks, see e.g. [7, 59]. Another exact method is the boundary value method, which focuses on two-dimensional queueing models. This method heavily relies on complex analysis, see e.g. [56, 57, 73]. Another class of multidimensional queueing models for which explicit results have been obtained, are polling models which satisfy the branching property. Besides these methods, there are some numerical-analytical methods such as the matrix-geometric and matrix-analytic techniques, the compensation method, and the power-series algorithm, see e.g. [4]. Next to this, a few exceptional models with specific characteristics exist that make it possible to find explicit results, oftentimes only for two-queue models (see e.g. [4]).

Some new approximation and numerical-analytical methods have been developed over the years (for a brief overview of the state-of-the-art, see Subsection 6.1.1). In this chapter, we introduce a new approximation scheme that may be used to obtain approximations for a far larger set of queueing models and we continue with the idea behind our approximation scheme.

Our method is designed around the PGF for the joint steady-state queue-length distribution. Typically, a functional equation for this PGF as in Equation (6.1) below is relatively easy to derive for a queueing model and a more general version of Equation (6.1) in fact holds for a broad set of multidimensional queueing models. For now, let $P(x, y)$ be the PGF of the joint steady-state queue-length distribution for a two-dimensional queueing model (higher dimensions will be discussed later). Then we typically have a relation like the following:

$$K(x,y)P(x,y) = f_1(x,y)P(0,y) + f_2(x,y)P(x,0) + f_3(x,y)P(0,0), \tag{6.1}$$

or alternatively

$$P(x,y) = \frac{f_1(x,y)P(0,y) + f_2(x,y)P(x,0) + f_3(x,y)P(0,0)}{K(x,y)}, \tag{6.2}$$

for certain known functions $f_1(x,y)$, $f_2(x,y)$, $f_3(x,y)$, and $K(x,y)$. We refer to $K(x,y)$ as the kernel. The function $P(x,y)$ is *unknown* (and hence $P(0,0)$, $P(0,y)$, and $P(x,0)$ are unknown as well). The difficulty in finding an exact expression for $P(x,y)$ lies in these latter, unknown functions $P(0,y)$ and $P(x,0)$. In some particular queueing models these unknown functions can be found (such as the model with two $M/M/1$ queues in series discussed below). Another approach is to use e.g. the boundary-value technique, see [57, 73]. This technique relies on zeros of the function $K(x,y)$ on and within the unit circle and the fact that $P(x,y)$ is analytic within and continuous up to the unit circle (as it is a PGF). Indeed, if $K(x,y) = 0$, the numerator on the right-hand side of Equation (6.2) has to be zero as well. This typically leads to some kind of boundary value problem, which can then sometimes be used to obtain (complicated) expressions for $P(0,y)$ and $P(x,0)$. We discuss in Example 6.1 a concrete example where $P(x,y)$ is derived (and for which we happen to have an explicit expression).

Example 6.1 (Two $M/M/1$ queues in series) *In this example a model with two $M/M/1$ queues in series is considered. We have an arrival rate μ at the first queue, a service rate ν_1 at queue 1, and a service rate ν_2 at queue 2. Customers arrive at queue 1 and upon service completion move to queue 2. After completing a service at queue 2, a customer leaves the system. We assume the interarrival times and service times to be independent of one another and we assume that $\rho_1 := \mu/\nu_1 < 1$ and $\rho_2 := \mu/\nu_2 < 1$. Then we have the following balance equations, where p_{n_1,n_2} denotes the steady-state probability that there are n_1 customers at queue 1 and n_2 at queue 2:*

$$\begin{aligned}(\mu + \nu_1 \mathbb{1}\{n_1 > 0\} + \nu_2 \mathbb{1}\{n_2 > 0\}) p_{n_1,n_2} = \\ \mu \mathbb{1}\{n_1 > 0\} p_{n_1-1,n_2} + \nu_1 \mathbb{1}\{n_2 > 0\} p_{n_1+1,n_2-1} + \nu_2 p_{n_1,n_2+1},\end{aligned}$$

where $\mathbb{1}\{n_i > 0\} = 1$ if $n_i > 0$ and 0 otherwise. Then, the PGF of the joint steady-state queue-length distribution, $P(x,y) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} p_{n_1,n_2} x^{n_1} y^{n_2}$, can be shown to satisfy:

$$K(x,y)P(x,y) = \nu_1(xy - y^2)P(0,y) + \nu_2(xy - x)P(x,0), \tag{6.3}$$

where

$$K(x,y) = xy\left(\mu(1-x) + \nu_1(1 - y/x) + \nu_2(1 - 1/y)\right).$$

One can easily verify that the following equation for $P(x,y)$ satisfies Equation (6.3)*:*

$$P(x,y) = \frac{1}{1-\rho_1 x}\frac{1}{1-\rho_2 y}. \tag{6.4}$$

One way to verify this is by substituting Equation (6.4) *into Equation* (6.3)*. However, finding Equation* (6.4) *is, in general, not an easy task. In the current example, one can use the concept of detailed balance to, directly, derive that $p_{n_1,n_2} = (1-\rho_1)\rho_1^{n_1}(1-\rho_2)\rho_2^{n_2}$, from which Equation* (6.4) *readily follows (see e.g. [103]). Another example where this is possible, is the $M/M/1$ queue in series with an $M/G/1$ queue (one can use Burke's theorem [36] to show that the output process of an $M/M/1$ is, in equilibrium, a Poisson process with rate μ). In case of an $M/G/1$ queue followed by an $M/M/1$ queue, the analysis already complicates severely as is shown in e.g. [14, 27]. In general, however, solving an equation like Equation* (6.3) *is often not possible, so we are left with the question of how to obtain Equation* (6.4) *from an equation like Equation* (6.3)*.*

Instead of the intricate boundary value problem analysis or other methods to obtain $P(x,y)$ explicitly, we propose an approximation based upon an idea stemming from one-dimensional queueing models. In several one-dimensional queueing models, the queue-length PGF has a finite number of unknowns, which can be found using certain zeros of a one-dimensional kernel $K(x)$ and a normalization, which together lead to a finite-sized system of linear equations. Examples are the bulk-service queue [98], the FCTL queue (see e.g. Subsection 1.3.1), and a single-server vacation queue with customer-limited service [120]. In multidimensional settings, this is often not possible, because we would be dealing with a system of linear equations of *infinite* size as the number of unknowns in the functions $P(0,y)$ and $P(x,0)$ is often infinite. Therefore, we propose, as an approximation, to replace the functions $P(0,y)$ and $P(x,0)$ with Taylor series, $\tilde{P}(0,y)$ and $\tilde{P}(x,0)$, with only a *finite* number of coefficients. Then, using some roots of $K(x,y)$ and a normalization equation, we are able to approximate $P(x,y)$. This allows us to derive performance measures from the approximated PGF. For more details on the procedure, we refer to Section 6.2.

Although our method can be used to obtain approximations for queueing models in any dimension, it slows down when approximating models with an increasing dimension, because of an increasing number of unknowns that needs to be estimated. This relates to the curse of dimensionality which appears often in multidimensional (queueing) models and which for example relates to quickly increasing computation times because of an exploding state space. In

our examples, we see an increase in the size of the system of linear equations that we need to solve, which in turn implies an increase in computation time. Our method therefore works best for two-dimensional models, but can be used for higher-dimensional queueing models as well which we also demonstrate in this chapter.

Our method enables one to study models that can be described by a functional equation like Equation (6.1). An example of such a class of models is the class of k-limited polling models. In the beginning, we developed our method specifically for k-limited polling models (e.g. because they directly relate to vehicle-actuated controlled traffic lights). For k-limited polling models, the joint distribution of the queue length at service and switchover beginnings can be described by (a generalization of) Equation (6.1) and our method can be leveraged to find approximations for various performance measures. Other models that we have successfully approximated, include a two queue model with an alternating service discipline [68] (see Section 6.4), a tandem queue model with coupled processors [209], a fork-join queue [74, 75], and the 2x2 switch [31] (this is by no means an exhaustive list). Our method also allows us to obtain approximations for several models which can be considered to be variants of k-limited polling models, such as a polling model with multiple servers. Those might be used to model a slightly different version of vehicle-actuated controlled traffic lights than the standard approach using k-limited polling models, see also Section 6.5.

The general applicability of our method is a definite and clear advantage of our approximation scheme: we are able to obtain approximations for many more queueing models (in a practical way) than the currently existing methods are capable of. We describe some of the existing methods and provide some pros and cons of them in the next subsection after giving the main contributions of this chapter.

Our main contributions can be summarized as follows:

(i) We develop a novel approximation scheme for multidimensional queueing models that focuses on approximations for PGFs.

(ii) We demonstrate the use of our approximation scheme by studying various multidimensional queueing models showing that our scheme in general leads to qualitatively good approximations.

6.1.1 Some numerical-analytical and approximation schemes

In the last 60 to 70 years many approximation schemes have been developed for queueing models. They range from specific formulas like Kingman's approximation for the mean waiting time in a $G/G/1$ queue to complex methods involving e.g. linear optimization. We provide an overview of the most relevant numerical-analytical and approximation methods that can be used for (part of the) models to which our novel scheme can be applied as well. We focus on methods for two-dimensional queueing models, although most approximations below can be used for queueing models with three dimensions or more as well.

Balance equations and state-space truncation. A standard approximation scheme is the use of balance equations and an appropriate truncation of the state space, i.e. the probabilities for the states that have a larger queue (in any direction of the two dimensions) than a certain bound are estimated to be zero. This bound should be sufficiently high. Then, the transition probabilities from one state to another can be found, which leads to a set of linear equations which (together with a normalization equation) can be solved to find an estimation of the probabilities in the original queueing model. Examples can be found in [184, Supporting Information, Section 1.7.2] and [137].

It might be possible (depending on the exact model) to formulate error bounds for the performance measures of interest, which is a benefit of this approach. On the other hand, a disadvantage is that the number of probabilities that needs to be estimated is quadratic in the truncation parameter. Moreover, the balance equations are typically easy to derive when the service times are exponential. Generalizations to phase-type distributions are possible, which slightly complicates the derivation of the balance equations. As the phase-type distributions are dense in the space of all distributions of non-negative random variables (see e.g. [174]), approximations for any service-time distribution are possible. However, such approximations typically tend to lead to a relatively quick increase in the number of states.

The matrix-geometric and matrix-analytic approach. The matrix-geometric method [5, 117] and the matrix-analytic method [5, 140] are methods that are used to study Quasi Birth-and-Death (QBD) processes. These methods use the transition rates directly together with a truncation of the state-space in a *single* dimension. The states are then ordered in such a way that they represent levels and phases. There are infinitely many levels (which correspond to the

queue for which there is no truncation) and there is a finite number of phases in each level (corresponding to the queue with the truncation). Except for level zero, the number of phases in each level should be the same and the transition probabilities between phases in each level should be the same. On top of that, the outgoing transitions from level m are restricted to the neighboring levels $m-1$ or $m+1$. Then, the transition matrix has a block tri-diagonal structure. In both the matrix-geometric and the matrix-analytic method, this structure can be leveraged to find a recursion for the steady-state probabilities. In both methods, a solution for a matrix-quadratic equation has to be found, which can then be exploited to find the steady-state probabilities. In Chapter 6 of [5] a recursion is given to solve this matrix-quadratic equation.

There exist several extensions compared to the standard formulation described above. There are ways to allow for an infinite number of phases, see e.g. [185], and specific structures within the process might allow for an easy solution of the aforementioned matrix-quadratic equation, see e.g. [161]. Moreover, there are extensions that allow for transitions between non-neighboring levels, which relate to so-called Quasi Skip-Free processes, see e.g. [5].

When the truncation parameter increases, these methods tend to be quicker than the previously discussed method that makes direct use of the balance equations. Matrix-geometric and matrix-analytic methods are most easy to apply to queueing models if the service times are exponential. Phase-type service-time distributions can also be handled, but that is at the expense of an increase in the state space.

The compensation approach. The compensation approach has been developed in a series of papers by Adan et al., see e.g. the PhD thesis of Adan and references therein [3]. Originally, the compensation approach was designed to tackle random walks in the positive quarter plane with homogeneous nearest-neighbor transitions in the interior of the positive quarter plane and homogeneous transitions along the boundary of the quarter plane. With some restrictions on the transitions (in the interior there should be *none* to the north, north-east and east), one can show that the steady-state probabilities can be computed as a sum of product-form terms. The product-form terms can be found as follows: start with a product-form solution for the steady-state probabilities in the interior of the quarter plane, and then add, alternately, compensation terms to account for the different transitions on the two boundaries. Extensions of the original compensation approach to higher dimensions exist (with additional constraints on the allowed transitions), see e.g. [204], and to queueing models

with transitions to non-nearest-neighbor states, see e.g. [173].

The explicit expressions obtained from this method are a clear advantage. However, its applicability is limited by the restrictions on the transitions between various states.

The power series algorithm. The next method to study multidimensional queueing processes that we discuss, is the power series algorithm, see e.g. [13, 94]. The power series algorithm requires a Markov representation of the queueing process and then a power-series expansion of the steady-state probabilities is formed, often in terms of the load on the system. This power-series expansion is recursively computed on the basis of global balance equations [13] until a desired level of precision is achieved (and there is, thus, a truncation of the state space). The radius of convergence and the convergence rate of the algorithm are limiting factors in the original implementation of the power-series algorithm. Several enhancements are available, see e.g. [199].

A positive element of this approach is its general applicability as there are no strict requirements on the transitions between the various states as in the previously discussed methods.

Markov reward approach. The Markov reward approach is an approximation scheme with a different approach than the ones discussed so far. Instead of a direct approximation, error bounds for performance measures are obtained. If these error bounds are sufficiently tight, we might see those bounds as an approximation. One of the first works in this line of research is by Van Dijk, see e.g. [201]. In this approach, one often seeks for a slight modification of the original queueing model, where the modified version allows for an exact analysis (often in terms of a product-form solution). Then, using Markov reward analysis techniques, bounds for the original queueing model can be obtained, see e.g. [26, Chapter 9]. Recently, this line of research has been extended with a linear programming approach [85], which has been further developed in [11, 48].

The latter method is quite flexible and alleviates some of the technical verification steps, but remains mainly limited to exponential service-time distributions, as is noted in [26, Section 9.7.1]. Moreover, Boucherie and Van Dijk state in [26, Section 9.7.3] that further developments of the approximation scheme are of substantial interest. One of the reasons is that the verification steps in the approach become more difficult in case of discrete-time queueing models and

another one is that it would be more difficult to find analytic expressions for the modified system.

Precedence relations. In a similar spirit, using a Markov reward structure, the precedence-relation method has been developed by Van Houtum et al., see e.g. the PhD thesis of Van Houtum [203, Chapters 5 to 7] and [205]. Also in this approach, error bounds for performance measures are constructed, but now by means of so-called precedence relations. Precedence relations between states m and n are formulated using the cost incurred over a period t, denoted with $v_t(m)$ and $v_t(n)$. Then state m has precedence over state n if $v_t(m) \leq v_t(n)$ for all $t \geq 0$. Such precedence relations can then be used to obtain lower or upper bounds for performance measures for the original queueing model by modifying the original queueing model to an alternative queueing model (which is usually easier to analyze, either explicitly or numerically).

The precedence-relation method is especially useful in cases where the modified queueing model is easy to analyze. Random walks with only nearest-neighbor transitions, like shortest-queue systems as studied in [203, Chapters 5 and 6], seem to be among the easiest models to which the method can be applied. For other types of models it is usually more difficult to find useful modifications to obtain lower and upper bounds for the original model.

Light- and heavy-traffic approximations. This approach obtains a closed-form approximating formula for several performance measures of interest, like the mean queue length or the mean waiting time. This idea was developed in [166] for a single-queue model with Poisson input and was generalized later on, see [22] for an application to polling models. The idea is that, e.g., the mean waiting time can be described as a function of the load, ρ, which is approximated by a relatively simple function. Such a function could be a polynomial in terms of ρ divided by $1-\rho$. The light- and heavy-traffic limits (ρ going to 0 and 1 respectively) of the mean waiting time are then used to obtain the coefficients in the aforementioned polynomial to form an interpolation for the other values of ρ. Even if one of the limits is unknown, such an approximation can be constructed (which is then usually less accurate).

A definite advantage is that a closed-form expression is obtained, because such expressions e.g. allow for optimization purposes. A disadvantage is that a clear understanding of the model at hand is needed, in particular the light- and heavy-traffic limit of the model need to be known. Sometimes those are known, but quite often it is not straightforward to find such results (in particular the

heavy-traffic limit seems to be difficult to obtain in general). Next to this, we remark that the approximation works especially well in light- and heavy-traffic conditions, because the method ensures that the light- and heavy-traffic limits are met. As a consequence, approximations for examples with a medium load are typically most off under this scheme.

Chapter outline

We give a brief overview of the remainder of this chapter. In Section 6.2 we give a detailed description of our approximation scheme. We discuss several examples of queueing models to which we apply our method in Sections 6.3, 6.4, and 6.5. In those three sections we discuss the specific model at hand and demonstrate our method by means of studying one or more numerical examples. In Section 6.6 we wrap up with a general conclusion.

6.2 Approximation scheme

We start this section with Subsection 6.2.1 where we describe our novel approximation scheme in detail. For the ease of exposition we present the approach for two-dimensional queueing models. We also provide a list of sufficient assumptions that ensure that our approximation scheme can be used. Subsequently, in Subsection 6.2.2, we give the algorithm which we generally use to obtain root pairs of the kernel function as present in Equation (6.1), which we need for our approximation scheme. In Subsection 6.2.3, we give some further intuition and background on the root pairs. We also show that there might be a significant influence of the used roots pairs on the quality of the obtained approximation.

6.2.1 Detailed description of approximation scheme

We start the description of our approximation scheme with defining the queueing process under consideration. We provide the description in two dimensions, which can easily be generalized in several ways (upon which we briefly return at the end of this subsection). Let (Q_1, Q_2) be the joint steady-state queue length with $\mathbb{N}^2$ as underlying state space and let $p_{i,j} = \mathbb{P}\big((Q_1, Q_2) = (i, j)\big)$. Define the PGF of the joint steady-state queue-length distribution as $P(x, y) = \mathbb{E}[x^{Q_1} y^{Q_2}]$.

In general we have a functional equation like Equation (6.1). The functions $K(x, y)$, $f_1(x, y)$, $f_2(x, y)$, and $f_3(x, y)$ are known and are generally non-zero. As

can be seen from the definitions, we have that

$$P(x,0) = \mathbb{E}[x^{Q_1}\mathbb{1}\{Q_2 = 0\}] = \sum_{i=0}^{\infty} p_{i,0}x^i,$$
$$P(0,y) = \mathbb{E}[y^{Q_2}\mathbb{1}\{Q_1 = 0\}] = \sum_{j=0}^{\infty} p_{0,j}y^j,$$
$$P(0,0) = \mathbb{P}(Q_1 = 0, Q_2 = 0) = p_{0,0},$$

where $\mathbb{1}\{Q_i = 0\} = 1$ if $Q_i = 0$ and 0 otherwise. As mentioned before, the difficulty lies in obtaining $P(x,0)$ and $P(0,y)$.

Instead of an exact computation scheme, we replace $P(x,0)$ and $P(0,y)$ by functions $\tilde{P}(x,0)$ and $\tilde{P}(0,y)$, which are defined as follows:

$$\tilde{P}(x,0) = \sum_{i=0}^{M_1} \tilde{p}_{i,0}x^i,$$
$$\tilde{P}(0,y) = \sum_{j=0}^{M_2} \tilde{p}_{0,j}y^j,$$

where the $\tilde{p}_{i,j}$ are approximations for the $p_{i,j}$. To obtain the $\tilde{p}_{i,j}$ we require that the $\tilde{p}_{i,j}$ satisfy,

$$f_1(x_i,y_i)\tilde{P}(0,y_i) + f_2(x_i,y_i)\tilde{P}(x_i,0) + f_3(x_i,y_i)\tilde{p}_{0,0} = 0, \tag{6.5}$$

for a root pair (x_i,y_i) with $|x_i| < 1$, $|y_i| < 1$, and $K(x_i,y_i) = 0$. This resembles the requirement that $P(x,y)$ should be zero if $K(x,y) = 0$ and x and y are within the unit circle, because $P(x,y)$ is analytic within the unit circle.

If we have sufficiently many root pairs, we can form a set of linear equations in terms of the $\tilde{p}_{i,0}$, $i = 0,\ldots,M_1$, and the $\tilde{p}_{0,j}$, $j = 1,\ldots,M_2$. We supplement those equations with a normalization equation. We e.g. require

$$\lim_{x\to 1} \frac{f_1(x,1)\tilde{P}(0,1) + f_2(x,1)\tilde{P}(x,0) + f_3(x,1)\tilde{p}_{0,0}}{K(x,1)} = 1, \tag{6.6}$$

reflecting that $P(1,1) = 1$ is a property that any PGF should satisfy. Then, we can form a system of $M_1 + M_2 + 1$ linear equations, which we can solve for the various $\tilde{p}_{i,0}$ and $\tilde{p}_{0,j}$. Those can then be used to approximate $P(x,y)$.We summarize the approximation scheme in pseudocode in Algorithm 6.1.

This completes our approximation scheme. The only thing left to do, is to obtain (at least) $M_1 + M_2$ roots of $K(x,y)$. This is the topic of Subsection 6.2.2.

Algorithm 6.1 Approximation scheme for $P(x, y)$.

1: Find (the implicit function for) the PGF of the joint steady-state queue-length distribution as in Equation (6.1).
2: Choose M_1 and M_2 and find $M_1 + M_2$ root pairs of $K(x, y)$ within the unit circle, see also Subsection 6.2.2, in particular Algorithm 6.2.
3: Form a set of linear equations based upon Equation (6.5) and a normalization equation like Equation (6.6).
4: Solve the resulting set of equations and plug in the $\tilde{p}_{i,0}$ and $\tilde{p}_{0,j}$ to obtain an approximation for $P(x, y)$.

We remark that several variants of Algorithm 6.1 are possible. Obviously, instead of using the normalization as in Equation (6.6), we might opt for different normalization equations or use several normalization equations (in which case we should leave out some of the other equations as we need a system of linear equations with size $M_1 + M_2 + 1$). Based on numerical experiments, using several normalization equations could be beneficial for the quality of the approximation scheme.

For completeness, we provide a list of *sufficient* assumptions for the basic version of our approximation scheme that ensures that we can use Algorithm 6.1.

- A functional equation for the PGF of the joint queue-length distribution, $P(x, y)$, like in Equation (6.1) needs to be available. We require $P(x, y)$ to be analytical for $|x| < 1$ and $|y| < 1$. Moreover, $P(x, y)$ should be continuous for $|x| \leq 1$ and $|y| \leq 1$.
- We require the model to be stable, i.e. $\mathbb{E}[Q_1] < \infty$ and $\mathbb{E}[Q_2] < \infty$.
- We need to be able to obtain sufficiently many root pairs (x_i, y_i) of the function $K(x, y)$ in Equation (6.1) for which both $|x_i| < 1$ and $|y_i| < 1$, e.g. using Algorithm 6.2 introduced below in Subsection 6.2.2.

Generalizations. There are several generalizations to other types of queueing models. One generalization is that we can approximate n-dimensional queueing models. Instead of Equation (6.1), we are then dealing with an implicit n-dimensional function for $P(x_1, x_2, \ldots, x_n)$ on the left-hand side which is multiplied with an n-dimensional kernel $K(x_1, x_2, \ldots, x_n)$. On the right-hand side we have $2^n - 1$ unknown functions which all relate to $P(x_1, x_2, \ldots, x_n)$, where one or several of the x_i are zero. This means that we have the following general form.

Let $\mathbf{n} \in \{0,1\}^n$, i.e. let $\mathbf{n}$ be a vector of length n with zeros and ones, and let $\mathcal{N}$ be the set of all such vectors of length n. Then

$$P(x_1,x_2,\ldots,x_n)K(x_1,x_2,\ldots,x_n) = \sum_{\mathbf{n}\in\mathcal{N}\setminus\{1,\ldots,1\}} f_{\mathbf{n}}(x_1,x_2,\ldots,x_n)P(x_1\mathbf{n}_1,x_2\mathbf{n}_2,\ldots,x_n\mathbf{n}_n), \tag{6.7}$$

where $\mathbf{n}_i$ denotes the i-th element of $\mathbf{n}$. Then, under appropriate cut-offs of the unknown functions on the right-hand side, we might use several n-dimensional roots of the kernel and one (or several) normalization equation(s) to find an approximation for $P(x_1,x_2,\ldots,x_n)$. We demonstrate this in Subsections 6.3.4 and 6.5.2.

Further, we might consider another generalization. Looking closely at Equation (6.1), we assume that the only special conditions for the transition rates can be found along the boundaries where $Q_i = 0$ for $i = 1,2$. We can generalize our scheme at the expense of additional unknown functions that we have to approximate to also allow for e.g. special transition rates when $Q_i = 1$ for $i = 1,2$. We are thus not restricted to queueing models where the only special conditions occur along the boundary. In essence, k-limited polling models could also be modeled in this way.

6.2.2 Finding suitable roots of the kernel

A practical challenge that comes with our approximation scheme is the fact that we require roots of the kernel. In some cases, like 1-limited polling models with 2 queues, explicit expressions for the roots can be found [29]. However, roots of the kernel are typically not easy to find explicitly. We propose a generic scheme to find roots of the kernel (in two dimensions, a generalization to n dimensions is possible), which we describe in more detail in pseudocode in Algorithm 6.2. We first give a short explanation.

We start with choosing a grid structure for the y-variable, i.e. we choose a step size, δ, and obtain the points $y = k/\delta + l/\delta \cdot i$, for $k = -\delta,\ldots,-1,1,\ldots,\delta$ and $l = -\delta,\ldots,-1$, where $i^2 = -1$. We exclude y-variables with a real or imaginary part equal to zero to prevent numerical problems. Subsequently, we check whether the obtained y is within the unit circle and if so, we find an accompanying x by means of a numerical procedure such that the kernel is zero for the combination (x,y). We check whether the obtained x is within the unit circle and if so, we add the pair (x,y) to a list of suitable root pairs. Moreover, we employ the fact that if (x,y) is a root for the kernel, then the complex conjugate $(\bar{x},\bar{y})$ is a root

as well. This reduces the computational cost to obtain sufficiently many root pairs. Another benefit is that this stabilizes the numerical procedure later on: a polynomial with real coefficients, such as a PGF, which has complex root r, also has $\bar{r}$ as a root. Moreover, we construct a while loop among increasing values of δ to ensure that we obtain sufficiently many root pairs. Summarizing, we obtain the procedure as in Algorithm 6.2.

Algorithm 6.2 Obtaining roots of the kernel.

1: Set $\delta = 1$ and initialize the set of root pairs $\mathcal{Z} = \emptyset$.
2: **while** $|\mathcal{Z}| < M_1 + M_2$ **do**
3: Put $\mathcal{Z} = \emptyset$ and put $\delta = \delta + 1$.
4: Compute a list of y's with $y = k/\delta + l/\delta \cdot i$, with $k = -\delta, \ldots, -1, 1, \ldots, \delta$ and $l = -\delta, \ldots, -1$. Pick only those y that satisfy $|y| < 1$.
5: **for** each y **do**
6: Find, numerically, an x such that $K(x, y) = 0$.
7: If $|x| < 1$, store the combination (x, y) and $(\bar{x}, \bar{y})$ in $\mathcal{Z}$.
8: **end for**
9: **end while**
10: Return δ and $\mathcal{Z}$.

Whether or not we are able to find sufficiently many root pairs (x_i, y_i) such that both x_i and y_i are located within the unit circle depends on the kernel $K(x, y)$. A priori it is not clear if there are any root pairs of $K(x, y)$ within the unit circle and if so how many there are. We thus need to either assume or prove that there are sufficiently many zero pairs within the unit circle in order to employ our algorithm. We will prove that there are infinitely many of such root pairs for each of the models that we consider in the remainder of this chapter.

Example 6.1 (Two $M/M/1$ queues in series: roots of the kernel) *We derived that the kernel equation for the model with two $M/M/1$ queues in series is:*

$$K(x, y) = xy\big(\mu(1 - x) + \nu_1(1 - y/x) + \nu_2(1 - 1/y)\big).$$

We first state Rouché's theorem (see e.g. [6]), which we will use in the proof of Lemma 6.2 which is formulated below.

Theorem 6.1 (Rouché's theorem) *Let the bounded region D have a simple closed contour, C, as its boundary. Let $f(z)$ and $g(z)$ be analytic both in D and on C.*

Assume that $|f(z)| < |g(z)|$ on C. Then $g(z) - f(z)$ has in D the same number of zeros as $g(z)$, all zeros counted according to their multiplicity.

Then, we have the following lemma.

Lemma 6.2 *The kernel equation for the model with two $M/M/1$ queues in series, $K(x, y)$, has infinitely many root pairs (x_i, y_i) with $|x_i| = 1$ and $|y_i| < 1$.*

Proof. *We adapt the proof of Lemma 8.2 in [209] to our setting. We employ Rouché's theorem. To this end, we define*

$$f(y) = -\left(\frac{\nu_1}{\mu}\bar{x}y^2 + \frac{\nu_2}{\mu}\right),$$
$$g(y) = \left(1 + \frac{\nu_1}{\mu} + \frac{\nu_2}{\mu} - x\right)y.$$

Then we have, if $|x| = 1$, that

$$\mu x(f(y) + g(y)) = K(x, y).$$

Moreover, $f(y)$ and $g(y)$ are analytic inside and on the unit disk.

We have that, if $|x| = 1$ and $x \neq 1$,

$$|f(y)| \leq \frac{\nu_1}{\mu}|\bar{x}||y|^2 + \frac{\nu_2}{\mu} = \frac{\nu_1}{\mu}|y|^2 + \frac{\nu_2}{\mu}$$

and

$$|g(y)| = \left|1 + \frac{\nu_1}{\mu} + \frac{\nu_2}{\mu} - x\right||y| > \left(\frac{\nu_1}{\mu} + \frac{\nu_2}{\mu}\right)|y|.$$

Then we have for all $|y| = 1$ that

$$|f(y)| \leq \frac{\nu_1}{\mu} + \frac{\nu_2}{\mu},$$
$$|g(y)| > \frac{\nu_1}{\mu} + \frac{\nu_2}{\mu}.$$

Rouché's theorem then tells us that $f(y) + g(y)$ has the same number of roots inside $|y| = 1$ as $g(y)$. $g(y)$ has one root with $|y| < 1$ ($g(0) = 0$). So $K(x, y)$ has one solution satisfying $|y| < 1$ for all x with $|x| = 1$ and $x \neq 1$. $K(x, y)$ thus has infinitely many root pairs satisfying $|x| = 1$ and $|y| < 1$. □

6.2.3 Choice of the roots

The choice for the root pairs of the kernel may have a significant influence on the accuracy of our approximation scheme. This also seems to play a role in other approximation schemes, such as in [68], where approximations for a two-class queue with an alternating service discipline are provided and some examples show numerical instabilities. The choice of the root pairs seems to cause these problems (see also Section 6.4). Although we have been unable to detect a clear relation between the choice of the roots and the outcome of the approximation, we see some general patterns and we have gathered some intuition, which we share here.

In order to better understand the relation between the choices for the root pairs and the approximation, we start with a simple example. Consider an unknown function $p(x)$ and its Taylor series around $x = 0$: $p(x) = \sum_{i=0}^{\infty} p_i x^i$ with p_i unknown. Let us assume that we might sample $p(x)$ for various values of x. Then, if we want to approximate the p_i for $i = 0, 1 \ldots, n$ (as we essentially do in our approximation scheme), which x should we choose to sample? Theory on numerical analysis tells us that it is generally best to sample $p(x)$ for values of x close to 0 as $p(x)$ is a Taylor series around $x = 0$, see e.g. [35]. We probably also want to sample $p(x)$ for some positive and some negative values of x.

Based on this intuition for a one-dimensional function, we might expect that root pairs close to $(0,0)$ work well when approximating $P(x,0)$ and $P(0,y)$ in Equation (6.1), as $\tilde{P}(x,0)$ and $\tilde{P}(0,y)$ are essentially Taylor series around $x = 0$ and $y = 0$ respectively. This is indeed something that we generally observe: if all root pairs are close to $(0,0)$, we tend to get better approximations than in the case where we choose all root pairs far from $(0,0)$.

Also the observation that we would like to know $p(x)$ for both positive and negative values of x in the one-dimensional setting, finds its analogy in the setting with approximating $P(x,0)$ and $P(0,y)$ in Equation (6.1). E.g. if every root pair (x_i, y_i) of $K(x,y)$ has the property that $\mathrm{Re}(x_i) < 0$, the quality of the approximation is generally worse than in the case where there are x_i with positive and negative real part. The same holds for the y_i.

The above considerations have, partly, led to the general approach to obtain root pairs of $K(x,y)$ as described in Algorithm 6.2. Using Algorithm 6.2, we at least get some root pairs (relatively) close to $(0,0)$ and we span the entire unit circle, at least for y. Occasionally, it might happen that using Algorithm 6.2, all root pairs satisfy a property like $\mathrm{Re}(x) < 0$, which tends to lead to relatively bad approximations. In such cases, we usually opt for a combination of Algorithm 6.2 as above and Algorithm 6.2 with x and y interchanged to make sure

that a property like "all x satisfy $\mathrm{Re}(x) < 0$" cannot occur. In general, this approach to find root pairs works well and gives rise to useful approximations, which is the main reason to work with Algorithm 6.2 in this chapter.

Example 6.1 (Two $M/M/1$ queues in series: influence of the roots) *We compare two different choices for the set of root pairs used in our method to approximate the PGF of the joint steady-state queue-length distribution of the model with two $M/M/1$ queues in series. One set has root pairs close to zero, while the other set is obtained with Algorithm 6.2 and we study the difference in the quality of the approximation (which is easy, because we know an explicit formula for $P(x,y)$, see Equation (6.4)).*

We choose $\mu = 1$, $\nu_1 = 10$ and $\nu_2 = 2.5$. We further choose $M_1 = M_2 = 10$. Using Algorithm 6.2, we continue until $\delta = 4$ and we use the first 20 root pairs obtained by Algorithm 6.2 in Approximation 1. For Approximation 2 we also use Algorithm 6.2, but we continue until $\delta = 30$ and then use the 20 root pairs with the smallest total absolute value, i.e. the (x_i, y_i) such that $|x_i| + |y_i|$ is minimal. We approximate p_{n_1,n_2} in both cases for $n_1, n_2 = 0,1,2,3$, and obtain Tables 6.1 and 6.2. We give the approximation and the absolute relative error defined as $|a-e|/e \cdot 100\%$ where a is the approximation and e the exact value.

Table 6.1: The approximation obtained using Approximation 1 with various values of n_1 displayed in the columns and n_2 displayed in the rows. Between brackets we display the absolute relative error in %.

p_{n_1,n_2}	$n_1 = 0$	$n_1 = 1$	$n_1 = 2$	$n_1 = 3$
$n_2 = 0$	0.54 (0.0082)	0.22 (0.0079)	0.086 (0.019)	0.035 (0.21)
$n_2 = 1$	0.054 (0.0033)	0.022 (0.11)	0.088 (1.3)	0.0031 (11)
$n_2 = 2$	0.0053 (1.1)	0.0024 (12)	0.00014 (84)	0.0017 (389)
$n_2 = 3$	0.00106 (97)	−0.0011 (612)	0.0023 (2580)	−0.0023 (6710)

We see in Table 6.1 that the approximation for higher values of n_1 and n_2 quickly drops in quality, whereas in Table 6.2 we see that the absolute relative errors remain quite small. In general, Approximation 2, with the root pairs close to zero, is more accurate than Approximation 1 where we use root pairs further away from zero. The quality of the approximation thus depends on the used root pairs and root pairs close to zero seem to work better (at least in this example).

We note that increasing M_1 and M_2 further, e.g. to 100, leads to more accurate approximations. The difference in the quality of the approximation between the two sets of root pairs seems to decrease when the values of M_1 and M_2 are increased.

Table 6.2: The approximation obtained using Approximation 2 with various values of n_1 displayed in the columns and n_2 displayed in the rows. Between brackets we display the absolute relative error in %.

p_{n_1,n_2}	$n_1 = 0$	$n_1 = 1$	$n_1 = 2$	$n_1 = 3$
$n_2 = 0$	0.54 (0.015)	0.22 (0.015)	0.086 (0.015)	0.035 (0.014)
$n_2 = 1$	0.054 (0.015)	0.022 (0.015)	0.0086 (0.016)	0.0035 (0.021)
$n_2 = 2$	0.0054 (0.015)	0.0022 (0.017)	0.00086 (0.035)	0.00035 (0.16)
$n_2 = 3$	0.00054 (0.020)	0.00022 (0.067)	0.000087 (0.43)	0.000036 (2.7)

In a few rare cases, specific additional information about the kernel is known that can be used to find root pairs. E.g. in the case of a 1-limited polling model with two queues, a curve, named F in [29], describing root pairs of $K(x, y)$ can be identified. This function is used to formulate and solve a boundary value problem to obtain $P(x, y)$ in [29]. Points on F can also be used as root pairs in our approximation scheme and it turns out that the quality of the approximation increases when compared with the case where we have root pairs as chosen in Algorithm 6.2. Apparently, root pairs on a curve like F have a positive impact, although we do not have a clear understanding why. For general models, however, it is hard to find such curves and as such we do not advocate the use of curves like the curve F in [29] due to limited applicability. Even though root pairs based on Algorithm 6.2 yield (slightly) worse results for a 1-limited polling model with two queues, our approach to find root pairs can be used more generally.

A further investigation of the relation between the root pairs and the quality of the approximation is beyond the scope of this chapter.

6.3 k-limited polling models

k-limited polling models form a set of queueing models that have sparked a lot of interest in the literature, that are so far considered to be analytically intractable except for some specific cases, and that fall in the class of systems that our novel approximation scheme can be applied to. In general, a k-limited polling model consists of N queues and one single server. The server alternates between the various queues and when the server starts serving queue i, it serves at most k_i customers during that visit, after which the server switches to the next queue. We will focus on examples where the server visits the queues

sequentially in a cyclic and predetermined order. These k-limited polling models might be employed to model certain types of vehicle-actuated traffic-light control strategies, which is another reason for us to study them.

There are some approximation schemes that are specifically tailored towards k-limited polling models, besides the ones that we already discussed in Subsection 6.1.1. An example is [211], using a decomposition of the entire system into one-dimensional k-limited queues with state-dependent vacations. These queues are then iteratively approximated. Another example is [21], which employs a light- and heavy-traffic approximation. Van Houdt relies on the power series algorithm, Kronecker matrix representations, and the shuffle algorithm in [202]. As a last possible approximation we name the pseudo-conservation law, first formulated in [28]. The results can in some cases be leveraged to obtain exact results, for example in case of a symmetric 1-limited polling model with an arbitrary number of queues. We will use this to determine the quality of our approximation in the case of symmetric 1-limited polling models. Typically we are able to outperform and/or obtain more general results than existing approximation methods.

In the remainder of this section, we first give a brief and general description of k-limited polling models and provide the various functions that we need to execute our approximation algorithm. We also prove that there are sufficiently many zeros of the kernel equation. After that we study various examples, starting in Subsection 6.3.2 with a symmetric 1-limited polling model with 2 queues. For this example, we are able to calculate various performance measures analytically, which allows us to validate our approximation scheme. In Subsection 6.3.3 we study a large test bed of two-queue k-limited polling models in order to assess the general quality of our approximation scheme. Finally, in Subsection 6.3.4, we show that our approximation scheme can also be used to approximate k-limited polling models with more than two queues.

6.3.1 Joint steady-state queue-length analysis

We provide a high-level derivation of the relevant functions for the k-limited polling model that we need for our method. We also slightly generalize the method described in Section 6.2. We do so in order to be able to account for being in various states, as we want to take into account in which part of the cycle we are.

We have a k-limited polling model with N queues. We denote with k_i the maximum number of customers served at queue i per cycle. We divide the cycle into $N + \sum_{i=1}^{N} k_i$ states. We denote the states with the pair (i, j), where

$i = 1,\ldots,k_j+1$ and $j = 1,\ldots,N$. The index i corresponds for $i = 1,\ldots,k_j$ to the start of the k_j service epochs at queue $j = 1,\ldots,N$ and the index $i = k_j+1$ corresponds to the start of the switchover period from queue j to queue $j+1$ (where queue $N+1$ is to be understood as queue 1). We define $\left(Q_1^{(i,j)},\ldots,Q_N^{(i,j)}\right)$ to be the random vector with the joint steady-state queue-length distribution at the start of state (i,j) and accordingly we define

$$P_{i,j}(\mathbf{x}) = \mathbb{E}[\prod_{l=1}^{N} x_l^{Q_l^{(i,j)}}]$$

with $\mathbf{x} = (x_1,\ldots,x_N)$. We further denote with Q_j the random variable of the steady-state queue-length distribution at an arbitrary moment for queue j. We also introduce μ_i, the Poisson arrival rate at queue i; B_i, the generally distributed service time at queue i, $i = 1,\ldots,N$, with mean b_i and Laplace-Stieltjes Transform (LST) $\beta_i(.)$; and S_i, the generally distributed switchover time from queue i to queue $i+1 \mod N$, with mean s_i and LST $\sigma_i(.)$. We assume that all arrival processes, service times, and switchover times are independent. Further, we define $z(\mathbf{x}) = \sum_{i=1}^{N} \mu_i(1-x_i)$. Then, we have that

$$\begin{aligned} P_{1,1}(\mathbf{x})K(\mathbf{x}) = \sum_{j=1}^{N}\sum_{i=1}^{k_j}\Bigg(& P_{i,j}(x_1,\ldots,x_{j-1},0,x_{j+1},\ldots,x_n)x_j^{i-1}(x_j-\beta_j(z(\mathbf{x}))) \\ & \beta_j(z(\mathbf{x}))^{k_j-i}\sigma_j(z(\mathbf{x}))\prod_{l=j+1}^{N}\left(\beta_l(z(\mathbf{x}))^{k_l}\sigma_l(z(\mathbf{x}))\right)\prod_{l=1}^{j-1}x_l^{k_l}\Bigg), \end{aligned} \tag{6.8}$$

where

$$K(\mathbf{x}) := \prod_{j=1}^{N} x_j^{k_j} - \prod_{j=1}^{N}\beta_j(z(\mathbf{x}))^{k_j}\sigma_j(z(\mathbf{x})). \tag{6.9}$$

A derivation of these functions can be found in Appendix 6.A where also the PGF for the joint steady-state queue-length distribution at an arbitrary moment is given. Expressions for the other $P_{i,j}(\mathbf{x})$ can also be derived. We note that compared to the general form of the functional equation presented in Equation (6.7), we only have functions where only *one* of the x_j is replaced with 0 as can be seen in Equation (6.8). As such, we have $\sum_{j=1}^{N} k_j$ unknown functions, rather than the two in Equation (6.1) or the 2^N-1 in Equation (6.7). This is not a problem as we show in the coming numerical examples.

Finally, we require that the load,

$$\rho = \sum_{j=1}^{N} \mu_j b_j,$$

is less than 1 and that the utilization for queue i, which reads

$$u_i = \rho + \mu_i \frac{\sum_{j=1}^{N} s_j}{k_i},$$

is less than 1 to ensure stability for queue i, see e.g. [78].

As a last remark before turning to the number of zeros of $K(\mathbf{x})$ within the unit circle, we note that we have to generalize the notation slightly. For the remainder of this section, we denote the cut-off of the function $P_{i,j}(\mathbf{x})$ for $i = 1,\ldots,k_j$ and $j = 1,\ldots,N$ with $M_{i,j}$.

Number of zeros of the kernel

In this subsection, we prove that there are infinitely many zeros within the unit circle for Equation (6.9). This implies that we are able to find sufficiently many zeros which we need for our approximation scheme. We prove that the number of zeros within the unit circle for the kernel is infinite by means of applying Rouché's theorem, as detailed in Lemma 6.3.

Lemma 6.3 *The equation*

$$K(\mathbf{x}) = \prod_{j=1}^{N} x_j^{k_j} - \prod_{j=1}^{N} \beta_j(z(\mathbf{x}))^{k_j} \sigma_j(z(\mathbf{x})) = 0 \tag{6.10}$$

has infinitely many solutions $\mathbf{x} = (x_1,\ldots,x_N)$ *with* $|x_j| < 1$ *for* $j = 1,\ldots,N$.

Proof. We want to employ Rouché's theorem. With the notation as in Theorem 6.1, we choose D to be the unit disk and C to be the unit circle. We prove that

$$|f(z(\mathbf{x}^*))| = \left| \prod_{j=1}^{N} \beta_j(z(\mathbf{x}^*))^{k_j} \sigma_j(z(\mathbf{x}^*)) \right| < 1,$$

where $\mathbf{x}^* = (hse^{2\pi i 1/N}, hse^{2\pi i 2/N},\ldots, hse^{2\pi i N/N})$, with $|h| = 1$ and a fixed s with $|s| = 1$ and $s \neq 1$.

We start with noting that

$$\left|\prod_{j=1}^{N}\beta_j(z(\mathbf{x}))^{k_j}\sigma_j(z(\mathbf{x}))\right| = \prod_{j=1}^{N}\left|\beta_j(z(\mathbf{x}))\right|^{k_j}\left|\sigma_j(z(\mathbf{x}))\right|. \tag{6.11}$$

We prove that $|\beta_1(z(\mathbf{x}^*))| < 1$ and similarly the other terms on the right-hand side of Equation (6.11) can be shown to be less than 1.

As $\beta_1(x)$ is an LST, we have

$$\begin{aligned}
|\beta_1(z(\mathbf{x}^*)| &= \left|\int_0^\infty \exp\left(-\sum_{j=1}^{N}\mu_j\left\{1-hse^{2\pi ij/N}\right\}t\right)\mathrm{d}\mathbb{P}(B_1<t)\right| \\
&\leq \int_0^\infty \left|\exp\left(-\sum_{j=1}^{N}\mu_j\left\{1-hse^{2\pi ij/N}\right\}t\right)\right|\mathrm{d}\mathbb{P}(B_1<t) \\
&= \int_0^\infty \exp\left(-\sum_{j=1}^{N}\mu_j\left\{1-\mathrm{Re}\left(hse^{2\pi ij/N}\right)\right\}t\right)\mathrm{d}\mathbb{P}(B_1<t) \\
&< \int_0^\infty 1\mathrm{d}\mathbb{P}(B_1<t) = 1,
\end{aligned}$$

where in the first inequality we use the triangle inequality for integrals and in the last inequality we use that $\sum_{j=1}^{N}\mu_j\{1-\mathrm{Re}(hse^{2\pi ij/N})\}t > 0$. The latter is the case because $\left|e^{2\pi ij/N}\right| = 1$, $|h| = 1$, and $|s| = 1$, so

$$\mathrm{Re}\left(hse^{2\pi ij/N}\right) \leq 1$$

for all j. We moreover have that at least one $hse^{2\pi ij/N} \neq 1$ since $s \neq 1$, implying that

$$\sum_{j=1}^{N}\mu_j\left\{1-\mathrm{Re}\left(hse^{2\pi ij/N}\right)\right\}t > 0.$$

Using Equation (6.11), we thus have that

$$\left|\prod_{j=1}^{N}\beta_j(z(\mathbf{x}^*))^{k_j}\sigma_j(z(\mathbf{x}^*))\right| < \prod_{j=1}^{N}1^{k_j}\cdot 1 = 1.$$

Moreover, we have that

$$|g(\mathbf{x}^*)| = \left|\prod_{j=1}^{N}\left(hse^{2\pi ij/N}\right)^{k_j}\right| = |(hs)|^{\sum_{j=1}^{N}k_j}\left|e^{\sum_{j=1}^{N}2\pi ijk_j/N}\right| = 1$$

as $|h| = 1$ and $|s| = 1$.

As $f(z(\mathbf{x}^*))$ and $g(\mathbf{x}^*)$ are analytic on and within the unit circle, we can employ Rouché's theorem to conclude that $K(\mathbf{x}^*)$ has $\sum_{j=1}^{N} k_j$ roots for $|h| < 1$ for a fixed s with $|s| = 1$, $s \neq 1$, and $\mathbf{x}^*$ as defined before. This implies that there are infinitely many roots within the unit circle of Equation (6.10). □

We are now set to continue with the numerical examples.

6.3.2 1-limited symmetric polling example

We start with an example for which several exact results are known so that we can validate our approximation scheme. We focus on a 1-limited polling model which is symmetric in the arrival rate and the service and switchover time distributions. We choose to vary the arrival rate and keep the other parameters of the model fixed. We choose the service-time distribution to be deterministic with value 1/3 and the switchover time distribution to be exponential with rate 7. Then, using the pseudo-conservation law [28], we can compute the mean waiting time exactly. We can compare this exact value with an approximation of the mean waiting time based on our approximation for $P_{1,1}(x_1, x_2)$, see also Appendix 6.A where we show how to derive the joint steady-state queue-length distribution at an arbitrary time from the $P_{i,j}(x_1, x_2)$. Then, we can compute the mean queue length at an arbitrary time and employ Little's law to obtain the mean waiting time.

In Table 6.3 below, we show for varying arrival rates the load on the queues, the approximation of the mean waiting time, the exact value of the mean waiting time, and two parameters of our algorithm, namely δ, the final step size in Algorithm 6.2, and the value of $M_{1,1}$ (which is equal to the value of $M_{1,2}$). As a last remark before turning to Table 6.3, we note that we employ the symmetry of the queueing system at hand in our approximation scheme, as we know that $p_{i,j} = p_{j,i}$, so we also require symmetry in our approximation scheme: we enforce $\tilde{p}_{i,0} = \tilde{p}_{0,i}$. This reduces the computation time and seems to yield (slightly) more accurate approximations.

As can be seen from Table 6.3, our approximation scheme performs well for all arrival rates. Among others, we display the absolute relative error, defined as $|a - e|/e \cdot 100\%$ where a is the approximation and e the exact value. The maximum absolute relative error is 0.24%. The quality of the approximation can be further improved if the values of $M_{i,j}$ are increased further (and if δ is adjusted accordingly), see also Figure 6.1 below.

The maximal relative error is attained for an instance with a very high load

Table 6.3: Mean waiting time for a symmetric 1-limited polling model with 2 queues, deterministic service times with value 1/3 and exponential switchover time distributions with rate 7. We have chosen $M_{1,1} = M_{1,2}$ and various arrival rates, μ, leading to various utilizations, u. We further display the step size δ, the approximation, the exact result, and the absolute relative error in %.

μ	u	$M_{1,1}$	δ	Approximation	Exact result	Abs. relative error
0.05	0.05	10	7	0.23083	0.23083	$< 0.001\%$
0.15	0.14	20	7	0.26944	0.26944	$< 0.001\%$
0.25	0.24	40	7	0.31771	0.31771	$< 0.001\%$
0.35	0.33	60	7	0.37976	0.37976	$< 0.001\%$
0.45	0.43	80	8	0.46250	0.46250	$< 0.001\%$
0.55	0.52	100	8	0.57833	0.57833	$< 0.001\%$
0.65	0.62	100	8	0.75208	0.75208	$< 0.001\%$
0.75	0.71	150	9	1.04167	1.04167	$< 0.001\%$
0.85	0.81	150	9	1.62083	1.62083	$< 0.001\%$
0.95	0.91	200	9	3.35833	3.35833	$< 0.001\%$
1.00	0.95	300	11	6.83336	6.83333	$< 0.001\%$
1.01	0.96	375	12	8.57084	8.57083	$< 0.001\%$
1.02	0.97	450	13	11.4668	11.4667	0.001%
1.03	0.98	600	15	17.2620	17.2583	0.021%
1.04	0.99	900	18	34.7157	34.6333	0.238%

(above 0.99). This points to a challenge in our algorithm: high loads require high values of $M_{i,j}$, as we essentially need to estimate more probabilities to obtain a good approximation (and the caveat is that higher $M_{i,j}$ imply a larger computation time). To investigate the influence of the parameter $M_{i,j}$, we take a closer look at the case where the arrival rate is 1.04 and where we vary the $M_{i,j}$. We obtain Figure 6.1, where we have the value of $M_{i,j}$ on the horizontal axis and the approximation on the vertical axis. We see that there is a dependence between the values of the $M_{i,j}$ and the quality of the approximation. Figure 6.1 indicates that $M_{i,j}$ should be sufficiently high in order to obtain good approximations and depending on the error that one is willing to accept, $M_{i,j} \approx 700$ probably suffices in this particular example. A general rule-of-thumb for the height of $M_{i,j}$ is difficult to establish. However, we generally see that a higher load requires a higher $M_{i,j}$.

Our approximation scheme is thus capable of providing satisfactory approximations for the mean waiting time. Moreover, we readily obtain approximations

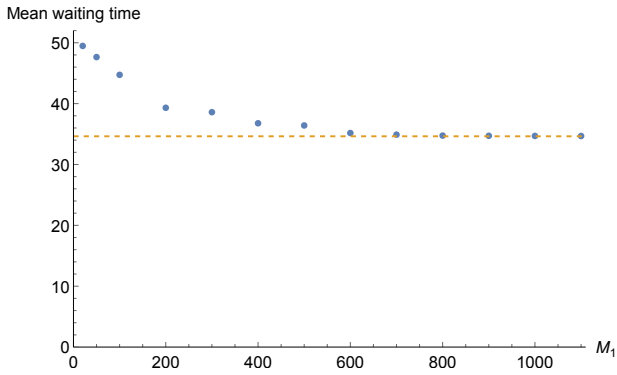


Figure 6.1: Mean waiting time estimation for various values of $M_{1,1}$ for a 1-limited polling model with 2 queues, a Poisson arrival rate of 1.04, deterministic service times with value 1/3, and exponential switchover times with rate 7. The dashed line represents the exact value.

for the PGF of the joint steady-state queue-length distribution, which also allows us to obtain higher moments and variances of e.g. the queue-length distribution.

6.3.3 Test bed for two-queue *k*-limited polling models

In this subsection, we create a test bed in order to assess the general quality of our approximation scheme when applied to k-limited polling models. To this end, we set up a test bed with a large variety of examples and we compare the results of our approximation scheme with extensive simulation results. An overview of the various parameter settings can be found in Table 6.4. We have varied the squared coefficient of variation (SCV) between 0, 1, and 2 for the service time distribution and between 0 and 1 for the switchover time distribution. In case the SCV is equal to 2, we fit a so-called hyperexponential distribution with balanced means as described in Example 2 of [22]. If the imbalance in the arrival rate is equal to a, then the total arrival rate at the two queues is equal to μ_{tot} such that $\mu_1 + \mu_2 = \mu_{tot}$, $a\mu_1 = \mu_2$, and such that the maximal load on either of the queues is equal to the load listed in Table 6.4.

In total, the number of cases in our test bed is equal to 2880. There are some symmetric cases for which we can take all $M_{i,j}$ the same as we did in

Table 6.4: Test bed used to compare the approximation to simulation results.

Parameters	Values
(k_1, k_2)	(1,1), (1,3), (3,3), (5,5)
Load	0.1, 0.3, 0.5, 0.7, 0.9
B_1	0.1, 1
B_2	0.1, 1
S_1	0.1, 1
S_2	1
SCV service times	0, 1, 2
SCV switchover times	0, 1
Imbalance arrival rates	1/3, 2/3, 1

Subsection 6.3.2, but there are also many asymmetric cases for which the approximation with all $M_{i,j}$ the same does not yield good approximations. We devise a general rule-of-thumb to obtain good values for the $M_{i,j}$. If we denote the total number of root pairs with t and the load on queue i with u_i, we choose the $M_{i,1}$ as $\lfloor t/k_1 \cdot (u_2 + 1/2)/(u_1 + u_2 + 1)\rfloor - 1$ and for the $M_{i,2}$ as $\lfloor t/k_2 \cdot (u_1 + 1/2)/(u_1 + u_2 + 1)\rfloor - 1$, if $k_1 = k_2$. If $k_1 \neq k_2$, we need to adjust for this and we choose the $M_{i,1}$ to be $\lfloor t/k_1 \cdot (u_2 + 1/2 + k_1)/(u_1 + u_2 + 1 + k_1 + k_2)\rfloor - 1$ and $\lfloor t/k_2 \cdot (u_1 + 1/2 + k_2)/(u_1 + u_2 + 1 + k_1 + k_2)\rfloor - 1$ for the $M_{i,2}$. In this way we can account for differences in the load and the k_i in both queues. The idea behind this general rule is that we need to account for differences in the loads and service limits in each queue. The higher the load on queue 1 (2), the longer the tails of the $P_{i,2}(x_1, 0)$, $i = 1, \ldots, k_2$ ($P_{i,1}(0, x_2)$, $i = 1, \ldots, k_1$) are. This in turn implies that we need (relatively) high values for the corresponding $M_{i,j}$ and indeed, in our rule-of-thumb, $M_{i,j}$ depends proportionally on the loads u_1 and u_2. Similarly, we need to take differences in k_1 and k_2 into account when $k_1 \neq k_2$ and also here we choose for a proportional dependence between the $M_{i,j}$ and the k_j. The devised rule-of-thumb generally seems to yield reasonably good approximations. Further, we provide a minimal *average* number of the various $M_{i,j}$, which we set to $(30, 75, 100, 200, 300)$ for maximal loads of respectively $(0.1, 0.3, 0.5, 0.7, 0.9)$. For each separate example we find an accompanying δ to make sure that we have sufficiently many root pairs.

In Tables 6.5 and 6.6 we focus on mean waiting times. In Table 6.5, we display the absolute relative error made by our approximation scheme when compared with extensive simulation results. When a is the approximation for the mean waiting time and s the mean waiting time obtained from extensive

simulations, we display the absolute relative error, $|a-s|/s \cdot 100\%$, categorized in bins of various sizes.

Table 6.5: Absolute relative error for the mean waiting time at queue i, denoted with $\mathbb{E}[W_i]$, expressed in % and categorized in bins for all cases in the test bed.

	$0-0.1\%$	$0.1-1\%$	$1-5\%$	$5-10\%$	$10-15\%$	$15-20\%$	$>20\%$
$\mathbb{E}[W_1]$	90.9	4.76	2.12	0.76	0.31	0.10	1.08
$\mathbb{E}[W_2]$	90.8	4.27	1.91	0.59	0.28	0.17	1.97

Table 6.6: Average absolute relative error for the mean waiting times at queues 1 and 2, expressed in % and categorized in bins. Under (a) we distinguish between the various combinations of the k_j and under (b) between various loads.

(a)

(k_1,k_2)	$0-0.1\%$	$0.1-1\%$	$1-5\%$	$5-10\%$	$10-15\%$	$15-20\%$	$>20\%$
$(1,1)$	99.7	0.28	0	0	0	0	0
$(1,3)$	86.5	7.50	4.58	0.69	0.42	0	0.28
$(3,3)$	87.5	6.53	2.64	0.83	0.56	0.28	1.67
$(5,5)$	79.7	8.89	4.03	1.25	0.69	0.14	5.28

(b)

Load	$0-0.1\%$	$0.1-1\%$	$1-5\%$	$5-10\%$	$10-15\%$	$15-20\%$	$>20\%$
0.1	98.8	1.04	0.17	0	0	0	0
0.3	97.7	1.91	0	0.17	0.17	0	0
0.5	93.8	4.34	1.04	0.17	0.17	0	0.52
0.7	88.9	4.69	3.12	0.69	0.35	0.17	2.08
0.9	62.7	17.0	9.72	2.43	1.39	0.35	6.42

We see that we obtain accurate approximations for the mean waiting times in many instances. In Table 6.5, we see that for more than 95% of all examples in the test bed we have an absolute relative error below 1% (and in many cases even below 0.1%) for both the mean waiting time in queue 1 and in queue 2 when we compare our approximations with extensive simulation results. Especially for the "easy" cases with a low load and/or low values of the k_j, we have accurate approximations as can be observed in Table 6.6. This is explained by the fact that in those cases we can choose the values of the $M_{i,j}$ to be relatively low, which still yields a qualitatively good approximation. Also when either the load or the values of the k_j are increased, we often obtain good approximations

for the mean waiting time, but those are at the expense of (slightly) larger approximation errors and/or longer computation times. Nevertheless, we would like to argue that, also in those cases, our approximation scheme is performing quite well.

In a few cases, our approximation scheme deviates more than 20% from the simulation results for at least one of the estimated mean waiting times. This amounts to 64 cases (out of 2880 cases). We investigated some of those cases separately and we were able to find a better approximation when increasing the values of the $M_{i,j}$ and/or slightly changing the ratio between the values of the $M_{i,j}$. This points towards a difficulty of our algorithm: how should the $M_{i,j}$ be chosen? Although we have a general rule-of-thumb, it does not work well in every single case and (a bit of) experimentation is sometimes required to find the right values. Especially when the load increases and/or the values of the k_j increase, the choice of the $M_{i,j}$ becomes more critical. However, in every separately investigated case we have been able to find better approximations than the ones displayed in Table 6.6 by adapting the various $M_{i,j}$.

We conclude with a few remarks. The total computation time for all approximations in the test bed is considerable (several days) on a high performance computing cluster. In many cases however, we could have chosen lower values for the $M_{i,j}$ leading to qualitatively similar approximations which would have decreased the computation time. We choose relatively large values for the $M_{i,j}$ due to the wide range of cases that we study in the test bed. Often, we could thus have performed our approximation with a lower computation time while maintaining the quality of the approximation. Moreover, our implementation of the approximation scheme is (probably) not the most efficient one. This could decrease the computation time further.

6.3.4 1-limited polling with three queues

As mentioned before, our method is also capable of approximating queueing models with more than two queues. We illustrate this by looking at an asymmetric k-limited polling model with three queues. We choose $k_j = 1$ for $j = 1,2,3$. We assume Poisson arrivals as before and we choose $\mu_j = 0.25$. The service-time distribution at queue j is chosen to be deterministic with value $1/j$ and all switchover time distributions are exponential with parameter 5. This implies that the utilization for each queue is $73/120 \approx 0.608$.

A difference with two-queue models is that we need a root triplet rather than a root pair. One has to be careful when selecting the root triplets that are used, as linear dependencies within the set of linear equations that we need

to solve might lead to numerical problems/instabilities. One could work with a structure like in Algorithm 6.2 for the two-queue scenario, but we opt for a different approach here (that also works in two dimensions). We generate a random list of x_2 and x_3 that are located within the unit circle and then find an accompanying x_1 such that $K(x_1, x_2, x_3) = 0$. We use a numerical root-finding procedure to obtain such an x_1. We cannot guarantee that such an x_1 exists, but in our algorithm we are almost always able to find such an x_1. If the obtained x_1 is within the unit circle, we add the triplet to a list of root triplets for $K(x_1, x_2, x_3)$ which are subsequently used to build the system of linear equations. Together with one or more normalization equations, we are then able to find the finite number of unknowns which we are looking for. We choose all cut-offs to be 15 and we work with the normalization equation

$$\lim_{x_1 \to 1} P_1(x_1, 1, 1) = 1.$$

In Table 6.7 we see that we obtain approximations with a quite small absolute error when we compare our results with simulation results: the absolute relative error is maximally 0.51%. We are not always within the 95% confidence interval obtained from extensive simulation runs, but we are always close to the simulated value.

Table 6.7: Various approximation and simulation results for performance measures of the three-queue k-limited polling model. Sim. stands for simulation and the lower and upper bound correspond to a 95% confidence interval. $\mathbb{E}[W_i]$ stands for the mean waiting time at queue i.

	Approximation	Sim. lower bound	Sim. upper bound
$\mathbb{E}[Q_1]$	0.6450	0.6448	0.6451
$\mathbb{E}[Q_2]$	0.4714	0.4713	0.4716
$\mathbb{E}[Q_3]$	0.4146	0.4148	0.4151
$\mathbb{E}[W_1]$	1.5801	1.5793	1.5803
$\mathbb{E}[W_2]$	1.3857	1.3856	1.3864
$\mathbb{E}[W_3]$	1.3251	1.3260	1.3267
$\mathrm{Var}(Q_1)$	0.8333	0.8330	0.8341
$\mathrm{Var}(Q_2)$	0.6023	0.6023	0.6030
$\mathrm{Var}(Q_3)$	0.5270	0.5294	0.5300

This example clearly illustrates that our method can be applied for queueing models with more than two queues. Even though the number of unknowns that

we need to find, quickly grows in this example (we already need to estimate 768 unknowns in this example), we are still capable of finding accurate approximations in a reasonable amount of time (about half an hour for the entire table). We note that our approximation scheme is quite sensitive to the values of the cut-offs in this example. For example, a small change in the $M_{i,j}$ might lead to a relatively big change in the approximation. A further investigation of this is beyond the scope of this chapter.

6.4 A two-class queue with alternating service discipline

The model that we discuss in this section is the same model as presented in [68] and its description is as follows. We have a time-slotted model with two queues and one server. There are two types of customers, where the type of the customer corresponds to a specific queue. During each time slot and for each type of customer, there are independent arrivals. Each customer has a service time of a single slot. In each time slot, the server flips a coin and with probability α it serves a customer at queue 1 and with probability $1-\alpha$ it serves a customer at queue 2. If a queue happens to be empty at the moment that the server wants to perform a service in that queue, the server idles until the next time slot.

In [68] the functional equation for the joint PGF of the queue-length distribution is derived but not solved. The authors in [68] rather study the dominant poles of this PGF and use the obtained information in approximation schemes for the joint steady-state queue-length distribution. The dominant poles are used to estimate the tail probabilities and for the remaining probabilities a set of linear equations is formed based on certain roots of the kernel. This strongly reminds of our method, yet our approach is more general; our method is based on PGFs and does not need the information coming from a dominant pole; and we use different roots, which seems to play a major role in the accuracy of the approximation, see Subsection 6.4.1 below. We continue with the functions that we need as input for our approximation scheme.

We define $p_{i,j}$ to be the joint probability that there are i customers in queue 1 and j customers in queue 2. The functional equation for the joint PGF of the queue lengths at the start of a slot, $P(x,y) = \sum_{i,j} p_{i,j} x^i y^j$, is given in Equation (14) in [68]. If we define $A_i(z)$ to be the PGF of the number of arrivals in a single slot at queue i, we get:

$$K(x,y)P(x,y) = A_1(x)A_2(y)\big((1-\alpha)(y-1)xP(x,0) + \alpha(x-1)yP(0,y)\big),$$

where

$$K(x,y) = xy - \big((1-\alpha)x + \alpha y\big) A_1(x) A_2(y).$$

We have the following lemma:

Lemma 6.4 *The equation*

$$K(x,y) = xy - \big((1-\alpha)x + \alpha y\big) A_1(x) A_2(y) = 0 \tag{6.12}$$

has infinitely many solutions with $|x| < 1$ *and* $|y| < 1$.

Proof. See Theorem 1 in [68]. □

The authors in [68] derive the dominant singularities of $P(x,y)$, $P(x,0)$, and $P(0,y)$. The residues of those functions at the dominant singularities are also derived. As is shown in e.g. [188, Subsection 2.3.3], [190], and [191], these might be used to approximate the tail probabilities of the (in our case) joint steady-state queue-length distribution. The idea to approximate tail probabilities with a geometric distribution is, thus, not new. We are in particular interested in the dominant singularities of $P(x,0)$ and $P(0,y)$, which are derived in Lemmas 3 and 4 of [68]. The corresponding residues are given in Theorems 2 and 3 in [68], which we provide here for further reference. We denote with τ_i the dominant pole for queue i, $i = 1,2$, and with B_i the residue at that pole. The τ_i are given implicitly (which can be found using a numerical solver) and are the unique solution to the equations below satisfying $1 < \tau_i < \sigma_i$ where σ_i denotes the radius of convergence of $A_i(z)$; for more details see [68]. We have that

$$\begin{aligned}
\tau_1 &= \big((1-\alpha)\tau_1 + \alpha\big) A_1(\tau_1), \\
\tau_2 &= (1-\alpha+\alpha\tau_2) A_2(\tau_2), \\
B_1 &= \frac{(1-\alpha)A_1(\tau_1) - A_2'(1)}{(1-\alpha)A_1(\tau_1) + \big((1-\alpha)\tau_1 + \alpha\big) A_1'(\tau_1) - 1} \cdot \frac{(\alpha - A_1'(1))(\tau_1 - 1)}{1-\alpha}, \\
B_2 &= \frac{\alpha A_2(\tau_2) - A_1'(1)}{\alpha A_2(\tau_2) + (\alpha\tau_2 + 1 - \alpha) A_2'(\tau_2) - 1} \cdot \frac{(1-\alpha-A_2'(1))(\tau_2 - 1)}{\alpha}.
\end{aligned}$$

The knowledge of those poles and residues enables one to provide approximations for the tail probabilities present in $P(x,0)$ and $P(0,y)$, as we have that

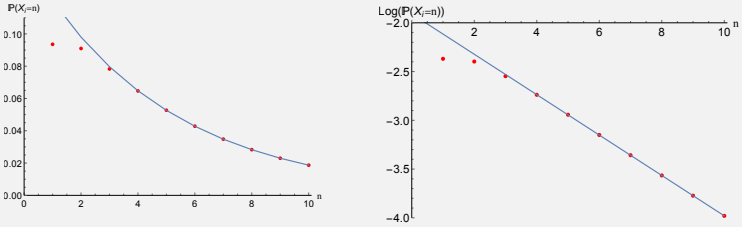
$$\begin{aligned}
\tilde{p}_{i,0} &\sim \frac{B_1}{\tau_1^{i+1}}, \\
\tilde{p}_{0,j} &\sim \frac{B_2}{\tau_2^{j+1}}.
\end{aligned} \tag{6.13}$$

Dominant pole approximation

The dominant pole approximation provides tail-probability approximations for the steady-state queue-length distribution (most often used for single-server queueing models; an example of the dominant pole approximation applied to a two-dimensional queue can be found in [68]). Under appropriate conditions, we might approximate the tail probabilities, p_n as follows:

$$p_n \approx \frac{s}{t^{n+1}}, \tag{6.14}$$

for some constants s and t, implying that the tail probabilities decay geometrically. The constants s and t can often be derived from the PGF of the queue-length distribution. For more information, we refer the interested reader to [210] and [188, Subsection 2.3.3].



(a) (b)

Figure 6.2: The dominant pole approximation for the overflow queue for the FCTL queue with $g = r = 5$ and Poisson arrivals in each slot with rate 0.45. In (a) we plot the probabilities (dots) for a specific queue length and the approximation (line). In the graph, the probabilities and the approximation are almost indistinguishable when the queue length is at least 5. In (b) we plot the same probabilities on a log-scale.

We demonstrate the dominant pole approximation for the overflow queue in the FCTL queue. The PGF of the overflow queue, $X_g(z)$, is (cf. Equation (1.5))

$$X_g(z) = \frac{z^g \sum_{i=0}^{g-1} X_i(0) \left(1 - \frac{Y(z)}{z}\right) \left(\frac{Y(z)}{z}\right)^{g-i-1}}{z^g - Y(z)^c}.$$

We might obtain t and s for the FCTL queue as follows. t is the root of $z^g - Y(z)^c$ outside the unit circle with smallest absolute value (such a root exists because of Pringsheim's theorem, see e.g. [167, page 235]) and s is the residue of $X_g(z)$ at t, i.e. $s = \lim_{z \to t} (z-t) X_g(z)$. Then, we approximate the probability that X_g is equal to n as in Equation (6.14).

The quality of the approximation is usually very acceptable, even for probabilities that are not in the tail as can be seen in Figure 6.2. Figure 6.2(b) confirms that the probabilities indeed decay geometrically, so that the geometric form as in Equation (6.14) is indeed right.

6.4.1 Example with high and asymmetric load

The model described in Devos et al. [68] is a hard model to analyze and in general good approximations for the joint steady-state queue-length distribution are obtained. For most examples studied by Devos et al. the devised approximation method indeed works well, but for some it yields relatively poor results. We studied all examples discussed in [68] and our approximations have a similar or a higher quality than the ones obtained by Devos et al. For Examples 1 and 2, the quality of the approximation in [68] is similar, whereas we obtain a (slightly) better approximation for Examples 3, 4, and 5. We study Example 5 in more detail as this example seems to cause the biggest problem for the approximation scheme in Devos et al.

In Example 5 in [68], it is assumed that the arrival distribution at queue 1 is geometric with parameter 0.164818 and the arrival distribution at queue 2 is Poisson with parameter 0.762360 (these input values are randomly generated by Devos et al.). Moreover, we have $\alpha = 0.214682$. We compare four different approximations: the one obtained by Devos et al. (the case with $M = 15$ in Figure 6 in [68]); our approximation method with $M_1 = M_2 = 16$, with M_1 the cut-off for $P(x,0)$ and M_2 the cut-off for $P(0,y)$, and root pairs obtained with Algorithm 6.2 and with Algorithm 6.2 with x and y interchanged; our approximation method with $M_1 = 286$ and $M_2 = 247$ (in accordance with the rule-of-thumb introduced in Subsection 6.3.3 for k-limited polling models) and root pairs obtained with Algorithm 6.2 and with Algorithm 6.2 with x and y interchanged; and our approximation method with $M_1 = M_2 = 16$, and root pairs obtained with Algorithm 6.2 and with Algorithm 6.2 with x and y interchanged, and where we approximate the remaining tail probabilities with the dominant pole approximation as in Equation (6.13). The various approximations can be

found in Table 6.8, where we also display simulation results.

Table 6.8: Various approximation methods for Example 5 in [68]. Each row corresponds to a probability that is estimated and each column represents an approximation method. From left to right, we display our approximation scheme with $M_i = 16$; our approximation with $M_1 = 286$ and $M_2 = 247$; our approximation with $M_i = 16$ together with a dominant pole approximation for the tail probabilities; the approximation from [68] as in Figure 6 in [68] with $M = 15$; and simulation results.

	$M_i = 16$	$M_1 = 286$	Dominant pole	[68]	Sim.
$p_{0,0}$	0.00953	0.00396	0.00393	0.00417	0.00394
$p_{1,0}$	0.00988	0.00410	0.00407	0.00435	0.00408
$p_{2,0}$	0.00912	0.00376	0.00372	0.00403	0.00375
$p_{3,0}$	0.00802	0.00326	0.00322	0.00358	0.00326
$p_{4,0}$	0.00692	0.00275	0.00271	0.00311	0.00273
$p_{5,0}$	0.00517	0.00227	0.00223	0.00268	0.00226
$p_{6,0}$	0.00391	0.00186	0.00181	0.00230	0.00185
$p_{7,0}$	0.00334	0.00150	0.00144	0.00196	0.00150
$p_{8,0}$	0.00335	0.00121	0.00114	0.00166	0.00121
$p_{9,0}$	0.00101	0.00097	0.00091	0.00139	0.00096
$p_{10,0}$	−0.00060	0.00077	0.00070	0.00114	0.00077
$p_{0,1}$	0.0154	0.00642	0.00638	0.00417	0.00641
$p_{0,2}$	0.0177	0.00740	0.00735	0.00677	0.00738
$p_{0,3}$	0.0186	0.00780	0.00775	0.00777	0.00778
$p_{0,4}$	0.0190	0.00795	0.00791	0.00815	0.00795
$p_{0,5}$	0.0185	0.00797	0.00794	0.00825	0.00798
$p_{0,6}$	0.0182	0.00789	0.00787	0.00814	0.00791
$p_{0,7}$	0.0177	0.00775	0.00774	0.00822	0.00777
$p_{0,8}$	0.0172	0.00756	0.00756	0.00604	0.00756
$p_{0,9}$	0.0149	0.00734	0.00738	0.01495	0.00734
$p_{0,10}$	0.0142	0.00711	0.00715	−0.02173	0.00710

Studying Table 6.8, we observe several interesting features. Firstly, we are able to obtain quite accurate approximations. Further, if we do not use the dominant pole approximation for tail probabilities, we need to take a relatively high value for M_i to obtain a satisfactory approximation. As can be seen, $M_i = 16$ is by no means sufficient to obtain accurate approximations, but when the M_i are increased, we obtain more accurate approximations. This is at the expense of a longer computation time. However, if we use the dominant pole approximation,

choosing $M_i = 16$ seems to be sufficient to obtain accurate approximations. This reduces the required computation time compared to the case where $M_1 = 286$ and $M_2 = 247$, while the quality of the approximation is similar in both cases.

It seems that the choice of the roots causes the difference in quality between our approach and the approximation in [68]. Currently, we do not see a clear relation between the choice of the roots and the quality of the approximation (see also the discussion in Subsection 6.2.3). We note that the roots that we obtain more or less span the entire unit circle, whereas this does not seem to be the case for the roots used in [68].

This example shows that our method might benefit from knowledge of the tail behavior of the steady-state probabilities. It potentially leads to a reduction in the M_i, which then leads to a significant reduction in the computation time. If information about the tail behavior is available, we recommend to use it as it improves the quality of the approximation and reduces the computational complexity. Our method can then more easily be used for queueing models with high loads and/or queueing models with more than two queues.

6.5 Traffic lights with double-lane access control

In this section we consider an extension of the k-limited polling model/vehicle-actuated control strategy considered in Section 6.3. In this section, we allow the server to serve two customers from two different queues simultaneously. In a traffic setting, this would correspond to a vehicle-actuated strategy where two opposing and non-conflicting streams of vehicles receive a green time simultaneously. This model has rarely been studied (as far as we are aware), probably because of its complicated nature. An exception is the study in Chapter 4, which investigates, by means of simulation, a heavy-traffic scaling of this model (see Subsection 4.4.2). For a graphical representation of the intersection and the control strategy we refer the reader to Figure 4.1(b).

We continue with a detailed description of the model in Subsection 6.5.1. We focus on a queueing model with 4 queues where 2 queues might be served simultaneously. In Subsection 6.5.2 we continue with an example with deterministic service and switchover times that mimics the setting in Subsection 4.4.2.

6.5.1 Joint steady-state queue-length analysis

As mentioned before, we focus on a model with 4 queues, where some queues are served simultaneously. For the ease of exposition, we assume that queues 1

and 2 are served simultaneously and we will refer to queues 1 and 2 as queues in group 1. This means that queues 3 and 4 are served simultaneously as well and we will refer to those queues as queues in group 2. We assume that a maximum of k_j customers are served at each queue in group j, $j = 1,2$, during a visit period to group j. The arrival process of customers at queue i is a Poisson process with rate μ_i, $i = 1,2,3,4$, and the service times at queue i are assumed to be random variables with distribution B_i with mean b_i and LST $\beta_i(.)$. After a service period at group j, a random switchover time to group $j+1$ is initiated (where group 3 is to be understood as group 1). This random time is denoted with S_j, and the first moment and the LST of the length of the switchover times are denoted with s_j and $\sigma_j(.)$. The various arrival processes, service times, and switchover times are assumed to be independent.

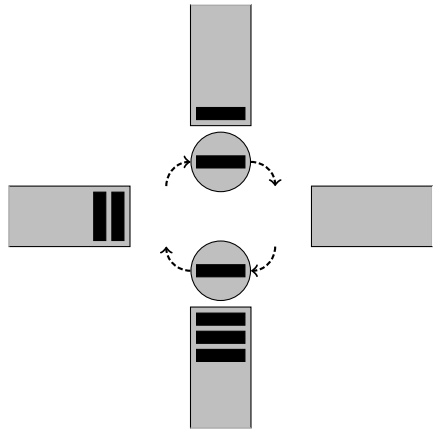


Figure 6.3: Queueing process for the vehicle-actuated traffic-light control strategy with double-lane access. Note that the servers switch at the same time and only switch if either both queues they are serving are empty or the maximum number of customers to be served during the current visit period is reached.

The service process at all groups of queues is as follows. If we take customers into service, we wait with taking the next set of customers into service until *all* customers of the previous set have completed their service. Then, if both queues in a group are non-empty, one customer of each queue is getting service; if one queue is empty, only a single customer from the other queue is taken into service; if both queues in group j are empty, we immediately initiate a switchover to group $j+1$. We also initiate a switchover if the maximum number of services

for a group of queues, denoted with k_j for group j, is reached. Moreover, we assume that once a queue empties, it stays empty for the remainder of the service visit of the server to group j (as in the FCTL queue, see Assumption 1.1). A visualization of the considered polling model can be found in Figure 6.3.

We divide the cycle into $k_1 + k_2 + 2$ states. We denote the states with the pair (i, j) where $i = 1, \ldots, k_j + 1$ and $j = 1,2$. The index i corresponds for $i = 1, \ldots, k_j$ to the start of the k_j service epochs for group j and the index $i = k_j + 1$ corresponds to the start of a switchover from group j to group $j + 1$. Then, we define $\left(Q_1^{(i,j)}, \ldots, Q_4^{(i,j)}\right)$ to be the random vector with the joint steady-state queue-length distribution at the start of state (i, j) and accordingly we define

$$P_{i,j}(\mathbf{x}) := \mathbb{E}[\prod_{l=1}^{4} x_l^{Q_l^{(i,j)}}],$$

with $\mathbf{x} = (x_1, \ldots, x_4)$. Further, we define $z(\mathbf{x}) = \sum_{i=1}^{4} \mu_i(1 - x_i)$. Then, we get that

$$\begin{aligned}
K(\mathbf{x})P_{1,1}(\mathbf{x}) = &\sum_{i=1}^{k_1} \beta_{3,4}(z(\mathbf{x}))^{k_2} \sigma_1(z(\mathbf{x}))\sigma_2(z(\mathbf{x}))\beta_{1,2}(z(\mathbf{x}))^{k_1-i}(x_1x_2)^i \cdot \\
&\left\{ P_{i,1}(0, x_2, x_3, x_4)\left(\frac{\beta_2(z(1, x_2, x_3, x_4))}{x_2} - \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right) + \right. \\
&P_{i,1}(x_1, 0, x_3, x_4)\left(\frac{\beta_1(z(x_1, 1, x_3, x_4))}{x_1} - \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right) + \\
&P_{i,1}(0, 0, x_3, x_4) \cdot \\
&\left.\left(1 - \frac{\beta_1(z(x_1, 1, x_3, x_4))}{x_1} - \frac{\beta_2(z(1, x_2, x_3, x_4))}{x_2} + \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right)\right\} + \\
&(x_1x_2)^{k_1} \sum_{i=1}^{k_2} \beta_{3,4}(z(\mathbf{x}))^{k_2-i}(x_3x_4)^i \sigma_2(z(\mathbf{x})) \cdot \\
&\left\{ P_{i,2}(x_1, x_2, 0, x_4)\left(\frac{\beta_4(z(x_1, x_2, 1, x_4))}{x_4} - \frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right) + \right. \\
&P_{i,2}(x_1, x_2, x_3, 0)\left(\frac{\beta_3(z(x_1, x_2, x_3, 1))}{x_3} - \frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right) + \\
&P_{i,2}(x_1, x_2, 0, 0) \cdot \\
&\left.\left(1 - \frac{\beta_3(z(x_1, x_2, x_3, 1))}{x_3} - \frac{\beta_4(z(x_1, x_2, 1, x_4))}{x_4} + \frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right)\right\},
\end{aligned} \tag{6.15}$$

where $\beta_{i,j}(.)$ is the LST of the maximum of the random variables B_i and B_j and

with

$$K(\mathbf{x}) = (x_1 x_2)^{k_1}(x_3 x_4)^{k_2} - \beta_{1,2}(z(\mathbf{x}))^{k_1}\beta_{3,4}(z(\mathbf{x}))^{k_2}\sigma_1(z(\mathbf{x}))\sigma_2(z(\mathbf{x})). \tag{6.16}$$

For a derivation of these expressions, we refer the interested reader to Appendix 6.B. We have the following lemma:

Lemma 6.5 *The equation*

$$K(\mathbf{x}) = (x_1 x_2)^{k_1}(x_3 x_4)^{k_2} - \beta_{1,2}(z(\mathbf{x}))^{k_1}\beta_{3,4}(z(\mathbf{x}))^{k_2}\sigma_1(z(\mathbf{x}))\sigma_2(z(\mathbf{x})) = 0 \tag{6.17}$$

has infinitely many solutions $\mathbf{x} = (x_1, \ldots, x_4)$ *with* $|x_j| < 1$ *for* $j = 1, \ldots, 4$.

Proof. An application of Rouché's theorem yields the result in a similar way as in the proof of Lemma 6.3. The proof is therefore omitted. □

6.5.2 Four-lane example

In this subsection we study an intersection with four lanes, where the two opposing lanes receive a green light simultaneously and with a maximum green time for each group of two lanes as we did in Subsection 4.4.2. We switch to the next group of lanes when either both lanes in group j are empty or when the maximum green time for group j, denoted with k_j, has been reached. Further, we have deterministic service and switchover times to create a slotted structure (as in the FCTL queue). The total maximum service time per lane is a multiple of a single slot. We choose to analyze a symmetric model, meaning that the arrival rates are the same for all four lanes; that the maximum green time for both groups is the same; that the length of each slot is the same; and that the switchover time between both groups is the same. This enables us to exploit symmetry between the various queues, which reduces the computational complexity of our approximation scheme. We choose the arrival rate at all lanes to be Poisson with parameter 0.25; we choose the maximum green time for each group to be 5; we choose deterministic service times with value 1; and we choose the switchover times to be deterministically equal to 1.

As for the polling model with three queues, we cannot directly rely on Algorithm 6.2 for obtaining roots of the kernel, as we now need a root quadruple instead of a root pair. Similarly as in the three-dimensional polling model, we choose x_2, x_3, and x_4 randomly within the unit circle and then find an accompanying x_1 so that $K(\mathbf{x}) = 0$, with $K(\mathbf{x})$ as in Equation (6.16). As in the three-dimensional polling model, we cannot guarantee that such an x_1 exists, but in

our algorithm we are almost always able to find such an x_1. When choosing the same cut-off for the various $P_{i,j}(\mathbf{x})$ in Equation (6.15), denoted with $M_{i,j}$, we get the results as in Table 6.9.

Table 6.9: Approximations for various performance measures for the double-lane access control of traffic lights example with various values for the $M_{i,j}$ and simulation results for the performance measures (where sim. res. is an abbreviation of simulation results). We display results for the mean and variance of the marginal queue length at queue 1 at the start ($Q_1^{(1,1)}$) and end of the green period ($Q_1^{(6,1)}$).

$M_{i,j}$	$\mathbb{E}[Q_1^{(1,1)}]$	$\mathrm{Var}(Q_1^{(1,1)})$	$\mathbb{E}[Q_1^{(6,1)}]$	$\mathrm{Var}(Q_1^{(6,1)})$
7	1.0843	1.2202	0.048028	0.06897
8	1.0617	1.3903	0.065087	0.15453
9	1.1081	1.5463	0.071901	0.23989
10	1.0563	1.4687	0.067741	0.19623
11	1.0676	1.2753	0.051058	0.09694
12	1.0689	1.3239	0.055104	0.12265
sim. res.	1.0773	1.3293	0.055730	0.11970

In Table 6.9 we observe that the various values of the $M_{i,j}$ do not always lead to accurate results and there is also no very clear pattern in the size of the approximation error, or at least not as clear as in Figure 6.1. In general, we see a decrease in the size of the approximation error when the $M_{i,j}$ increase and for the case $M_{i,j} = 12$, we think that we have a good approximation.

Even though the differences in the $M_{i,j}$ are small, they have a rather big impact on the computation time. E.g., for the case where $M_{i,j} = 7$, we have a linear system of equations with 1440 unknowns, whereas for the case with $M_{i,j} = 12$, we have a linear system of equations with 5915 unknowns. The computation time (with our implementation) in the former case is about 1 hour, whereas for the latter case this is about 90 hours. This points towards a limitation in our algorithm: when the number of queues increases, there is a quick and sharp increase in the number of unknowns that needs to be determined to get a good approximation. Especially in view of the need to solve a non-sparse system of linear equations which is linear in size of the number of unknowns, this is a complicating factor. Moreover, we require the solution to have a high precision (we use a 50 digit precision in this section), because we subsequently use the solution in further calculations (e.g. to obtain the mean queue length at the start of a visit period). Future work could be devoted to overcoming these issues around the computation time by designing a more efficient implementa-

tion. Nevertheless, this model fits our framework and, as far as we know, no approximations or exact results for this model have been derived. Moreover, we study this model because of its complexity and because of its application to road-traffic models. We have shown that our approximation scheme yields satisfactory approximations for the studied performance measures.

6.6 Conclusion

We have formulated a novel approximation scheme for multidimensional queueing models and demonstrated some of its numerical properties. Based on a functional equation for the joint steady-state queue-length distribution, we developed a methodology which uses roots of the kernel and subsequently uses the solution of a set of linear equations to provide an approximation for the PGF of the joint steady-state queue-length distribution. As we have shown, our approximation method yields good results in a plethora of examples, including the notoriously hard to analyze k-limited polling models.

A point of concern is the computation time that is needed to approximate queueing models with more than two queues and two-queue systems with a very high load. In such examples, we need to estimate a relatively large number of unknowns, causing an increased computation time for our approximation scheme as the size of the set of linear equations that needs to be solved increases. Although this is to some extent an artifact of our approximation scheme, there are various ways to decrease the computational complexity. One way is to make use of a dominant-pole type of approximation, as we have shown for the two-class queue with an alternating service discipline in Section 6.4. Such an extension essentially mitigates the negative effects of estimating certain tail probabilities to be zero. This causes a decrease in the number of unknowns that needs to be estimated by our approximation scheme, which reduces the computation time. This is a motivation to find dominant poles for two-dimensional (or even n-dimensional) queueing models.

We encourage further experimentation with approximating different models using our scheme, especially models with more than two queues. As we demonstrated, our approximation scheme is capable of providing accurate approximations for queueing models with three queues or more as well, but we did not do an in-depth study on queueing models with more than two queues.

The approximation scheme that we developed is amenable for several improvements. One is already elaborated upon in Section 6.4.1: if one has information about the tail behavior of the queue-length probabilities, then they can

be incorporated into the method at the benefit of reducing the computational complexity. We advocate a further investigation of which roots are to be used to find the best possible approximation given a certain number of roots that one can use. Although the developed method for finding roots generally seems to work well, we see that there is a potentially significant influence of the used roots and as such, we advocate a further study to try to understand what the relation is between the quality of the approximation and the used roots.

The implementation of the algorithm might also be improved upon in order to decrease the computation time that is needed to come to the approximations. There are probably several ways to improve upon our algorithm. For example, we have implemented our approximation scheme in Mathematica version 12.2 [221] and we expect that an implementation in `C++` would decrease the computation time considerably.

Also, we did not give any error bounds for our approximation scheme. Dwelling upon numerous experiments, we expect it to be difficult to come up with error bounds. Nevertheless, such an investigation on whether error bounds can be found is of interest, both from the perspective of the error bounds themselves and from the perspective of whether it is possible to establish such bounds at all. The reason for this is twofold: it would help in the understanding of our algorithm and it would potentially add to the understanding of the underlying queueing models.

Appendix

6.A PGFs for *k*-limited polling models

In this appendix, we derive the PGFs that we use in Section 6.3. For the model description and the notation, we refer to Subsection 6.3.1.

We relate the PGFs of the joint steady-state queue-length distribution at the start of the various states to one another. We get the following equations:

$$P_{i,j}(\mathbf{x}) = \left(P_{i,j}(\mathbf{x}) - P_{i-1,j}(x_1,\ldots,x_{j-1},0,x_{j+1},\ldots,x_N)\right)\frac{\beta_j(z(\mathbf{x}))}{x_j} + P_{i-1,j}(x_1,\ldots,x_{j-1},0,x_{j+1},\ldots,x_N), \tag{6.18}$$

for $i = 1,\ldots,k_j$ and $j = 1,\ldots,N$. We also have

$$P_{1,j+1}(\mathbf{x}) = P_{k_j+1,j}(\mathbf{x})\sigma_j(z(\mathbf{x})), \tag{6.19}$$

for $j = 1,\ldots,N$, where $P_{1,N+1}(\mathbf{x})$ is to be understood as $P_{1,1}(\mathbf{x})$. The cases corresponding to Equation (6.18) can be explained in the following way: first, we condition on queue j being empty or not. If queue j is non-empty, there is a customer taken into service in state (i,j) which is in service for a random time B_j. After such a service time, we make a transition to state $(i+1,j)$. The number of arrivals in between the start of state (i,j) and the start of state $(i+1,j)$ then has PGF $\beta_j(z(\mathbf{x}))$ and there is one service completion at queue j, which explains the factor $1/x_j$. If queue j is empty, we immediately make a transition from state (i,j) to $(i+1,j)$ and, as this takes no time, there are no arrivals. The cases corresponding to Equation (6.19) are explained as follows: in between the start

of state (k_j+1, j) and the start of state $(1, j+1)$, there are arrivals during a period S_j, which has PGF $\sigma_j(z(\mathbf{x}))$.

Using Equations (6.18) and (6.19), we are able to derive a functional equation for the $P_{i,j}(\mathbf{x})$. Repeated substitution yields the following:

$$\begin{aligned}
P_{1,1}(\mathbf{x}) =& P_{k_N+1,N}(\mathbf{x})\sigma_N(z(\mathbf{x})) \\
=& \Bigg(\left(P_{k_N,N}(\mathbf{x}) - P_{k_N,N}(x_1, x_2, \ldots, x_{N-1}, 0)\right) \frac{\beta_N(z(\mathbf{x}))}{x_N} + \\
& P_{k_N,N}(x_1, x_2, \ldots, x_{N-1}, 0) \Bigg) \sigma_N(z(\mathbf{x})) \\
=& \left(P_{k_N,N}(\mathbf{x}) \frac{\beta_N(z(\mathbf{x}))}{x_N} + P_{k_N,N}(x_1, x_2, \ldots, x_{N-1}, 0) \left(1 - \frac{\beta_N(z(\mathbf{x}))}{x_N}\right)\right) \sigma_N(z(\mathbf{x})) \\
=& \cdots = P_{1,1}(\mathbf{x}) \prod_{j=1}^{N} \frac{\beta_j(z(\mathbf{x}))^{k_j}}{x_j^{k_j}} \sigma_j(z(\mathbf{x})) + \\
& \sum_{j=1}^{N} \sum_{i=1}^{k_j} \left(P_{i,j}(x_1, \ldots, x_{j-1}, 0, x_{j+1}, \ldots, x_N) \left(1 - \frac{\beta_j(z(\mathbf{x}))}{x_j}\right) \cdot \right. \\
& \left. \left(\frac{\beta_j(z(\mathbf{x}))}{x_j}\right)^{k_j - i} \sigma_j(z(\mathbf{x})) \prod_{l=j+1}^{N} \left(\frac{\beta_l(z(\mathbf{x}))}{x_l}\right)^{k_l} \sigma_l(z(\mathbf{x})) \right).
\end{aligned}$$

This yields:

$$\begin{aligned}
P_{1,1}(\mathbf{x})K(\mathbf{x}) = \sum_{j=1}^{N} \sum_{i=1}^{k_j} \Bigg(& P_{i,j}(x_1, \ldots, x_{j-1}, 0, x_{j+1}, \ldots, x_N) x_j^{i-1} (x_j - \beta_j(z(\mathbf{x}))) \\
& \beta_j(z(\mathbf{x}))^{k_j - i} \sigma_j(z(\mathbf{x})) \prod_{l=j+1}^{N} \left(\beta_l(z(\mathbf{x}))^{k_l} \sigma_l(z(\mathbf{x}))\right) \prod_{l=1}^{j-1} x_l^{k_l} \Bigg),
\end{aligned}$$

where

$$K(\mathbf{x}) := \prod_{j=1}^{N} x_j^{k_j} - \prod_{j=1}^{N} \beta_j(z(\mathbf{x}))^{k_j} \sigma_j(z(\mathbf{x})).$$

Lastly, we derive the PGF of the joint steady-state queue-length distribution at arbitrary moments using the theory developed in [30]. We define

$$P(\mathbf{x}) = \mathbb{E}[\prod_{l=1}^{N} x_l^{Q_l}].$$

In order to derive an expression for $P(\mathbf{x})$, we need to introduce one more PGF for each queue: the PGF of the joint steady-state queue-length distribution at service completions at queue j, denoted with $\mathscr{P}_j(\mathbf{x})$. Combining Equations (4) and (5) from [30], we get that

$$\mathscr{P}_j(\mathbf{x}) = \frac{\gamma_j \beta_j(z(\mathbf{x}))}{x_j - \beta_j(z(\mathbf{x}))} \Big(P_{1,j}(\mathbf{x}) - P_{k_j+1,j}(\mathbf{x}) \Big),$$

with

$$\gamma_j = \frac{1-\rho}{\mu_j \sum_{j=1}^N s_j}.$$

Theorem 1 in [30] then states that

$$P(\mathbf{x}) = \frac{\sum_{j=1}^N \mu_j (1-x_j) \mathscr{P}_j(\mathbf{x})}{\sum_{j=1}^N \mu_j (1-x_j)}.$$

This enables us to find the marginal queue-length distribution at arbitrary times and (together with Little's law) the mean waiting time of customers at queue i.

6.B PGFs for traffic lights with double-lane access control

In this appendix, we derive the PGFs that we use in Section 6.5. For the model description and the notation, we refer to Subsection 6.5.1.

We relate the PGFs of the joint steady-state queue-length distribution at the start of various states to one another. We leave the derivation of the following equations to the reader.

$$\begin{aligned}
&P_{i,1}(\mathbf{x}) \\
&= \big(P_{i-1,1}(\mathbf{x}) - P_{i-1,1}(0,x_2,x_3,x_4) - P_{i-1,1}(x_1,0,x_3,x_4) + P_{i-1,1}(0,0,x_3,x_4) \big) \\
&\frac{\beta_{1,2}(z(\mathbf{x}))}{x_1 x_2} + \big(P_{i-1,1}(0,x_2,x_3,x_4) - P_{i-1,1}(0,0,x_3,x_4) \big) \frac{\beta_2(z(1,x_2,x_3,x_4))}{x_2} + \\
&\big(P_{i-1,1}(x_1,0,x_3,x_4) - P_{i-1,1}(0,0,x_3,x_4) \big) \frac{\beta_1(z(x_1,1,x_3,x_4))}{x_1} + P_{i-1,1}(0,0,x_3,x_4)
\end{aligned}$$

$$= P_{i-1,1}(\mathbf{x})\frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2} + P_{i-1,1}(0,x_2,x_3,x_4)\left(\frac{\beta_2(z(1,x_2,x_3,x_4))}{x_2} - \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right) +$$
$$P_{i-1,1}(x_1,0,x_3,x_4)\left(\frac{\beta_1(z(x_1,1,x_3,x_4))}{x_1} - \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right) +$$
$$P_{i-1,1}(0,0,x_3,x_4)\left(1 - \frac{\beta_1(z(x_1,1,x_3,x_4))}{x_1} - \frac{\beta_2(z(1,x_2,x_3,x_4))}{x_2} + \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right),$$
$$\text{for } i = 2,\dots,k_1+1,$$

$$P_{i,2}(\mathbf{x}) = P_{i-1,2}(\mathbf{x})\frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4} +$$
$$P_{i-1,2}(x_1,x_2,0,x_4)\left(\frac{\beta_4(z(x_1,x_2,1,x_4))}{x_4} - \frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right) +$$
$$P_{i-1,2}(x_1,x_2,x_3,0)\left(\frac{\beta_3(z(x_1,x_2,x_3,1))}{x_3} - \frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right) +$$
$$P_{i-1,2}(x_1,x_2,0,0)\left(1 - \frac{\beta_3(z(x_1,x_2,x_3,1))}{x_3} - \frac{\beta_4(z(x_1,x_2,1,x_4))}{x_4} + \frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right),$$
$$\text{for } i = 2,\dots,k_2+1,$$

$$P_{1,j+1}(\mathbf{x}) = P_{k_j+1,j}(\mathbf{x})\sigma_j z(\mathbf{x})), \qquad \text{for } j = 1,2,$$

where $P_{1,3}(\mathbf{x})$ is to be understood as $P_{1,1}(\mathbf{x})$.

Using these functions, we get the following expression for $P_{1,1}(\mathbf{x})$ by repeated substitution:

$$K(\mathbf{x})P_{1,1}(\mathbf{x}) = \sum_{i=1}^{k_1}\beta_{3,4}(z(\mathbf{x}))^{k_2}\sigma_1(z(\mathbf{x}))\sigma_2(z(\mathbf{x}))\beta_{1,2}(z(\mathbf{x}))^{k_1-i}(x_1x_2)^i\cdot$$
$$\left\{P_{i,1}(0,x_2,x_3,x_4)\left(\frac{\beta_2(z(1,x_2,x_3,x_4))}{x_2} - \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right) +\right.$$
$$P_{i,1}(x_1,0,x_3,x_4)\left(\frac{\beta_1(z(x_1,1,x_3,x_4))}{x_1} - \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right) +$$
$$P_{i,1}(0,0,x_3,x_4)\cdot$$
$$\left.\left(1 - \frac{\beta_1(z(x_1,1,x_3,x_4))}{x_1} - \frac{\beta_2(z(1,x_2,x_3,x_4))}{x_2} + \frac{\beta_{1,2}(z(\mathbf{x}))}{x_1x_2}\right)\right\} +$$
$$(x_1x_2)^{k_1}\sum_{i=1}^{k_2}\beta_{3,4}(z(\mathbf{x}))^{k_2-i}(x_3x_4)^i\sigma_2(z(\mathbf{x}))\cdot$$
$$\left\{P_{i,2}(x_1,x_2,0,x_4)\left(\frac{\beta_4(z(x_1,x_2,1,x_4))}{x_4} - \frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right) +\right.$$

$$P_{i,2}(x_1,x_2,x_3,0)\left(\frac{\beta_3(z(x_1,x_2,x_3,1))}{x_3}-\frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right)+$$
$$P_{i,2}(x_1,x_2,0,0)\cdot$$
$$\left(1-\frac{\beta_3(z(x_1,x_2,x_3,1))}{x_3}-\frac{\beta_4(z(x_1,x_2,1,x_4))}{x_4}+\frac{\beta_{3,4}(z(\mathbf{x}))}{x_3x_4}\right)\Bigg\},$$

with

$$K(\mathbf{x})=(x_1x_2)^{k_1}(x_3x_4)^{k_2}-\beta_{1,2}(z(\mathbf{x}))^{k_1}\beta_{3,4}(z(\mathbf{x}))^{k_2}\sigma_1(z(\mathbf{x}))\sigma_2(z(\mathbf{x})).$$

The equations for $P_{i,j}(\mathbf{x})$ with $i=1,\ldots,k_j+1$ and $j=1,2$ can be derived in a similar way.

Chapter 7

Platoon forming algorithms for intelligent street intersections

7.1 Introduction

In this chapter, we turn our focus to a futuristic setting. In the near future self-driving or autonomous vehicles might become the standard type of vehicle occupying the roads. In the US, self-driving vehicles have been driving around for quite some time already [119]. So, there is a need to study novel control algorithms for such self-driving vehicles as they allow for different and new strategies which improve the general traffic performance.

As we will show in this chapter, large time savings can be gained when self-driving vehicles are present on the roads. We show this by directly comparing a model with self-driving vehicles and nowadays traffic, for which we obtain simulation results using SUMO [129]. We provide a comparison by assuming that the arrival processes of vehicles in SUMO and our self-driving vehicles model are identical.

However, it is unlikely that the traffic load on intersections will remain the same if self-driving vehicles are introduced. Self-driving vehicles might cause induced demand: many more people will have access to "driving" around, think of elderly people and children. A case-study based on the city of Oslo, Norway, shows that the amount of vehicle kilometers is reduced with 13 percent in the most favorable scenario, but almost doubles in the worst scenario [61]. Of course there are a lot of ifs and buts, but it clearly demonstrates that not all

congestion will simply vanish when self-driving vehicles are present. So, an efficient way to accommodate the crossing of self-driving vehicles at intersections is needed, which is the topic of this chapter.

The traditional way of regulating the crossings of vehicles at a busy intersection is by installing traffic lights, e.g. with static signaling using timers such as is done in the FCTL queue or by means of vehicle-actuated control, see e.g. [159]. Anticipating the emergence of self-driving vehicles, efficient and fair algorithms for intersection access should be designed. Platoon Forming Algorithms (PFAs) provide such alternatives for self-driving vehicles, no longer letting the traffic lights dictate the switching process and hence batch forming, but letting the vehicles organize themselves in batches, well in advance of arriving at the intersection as in [133, 134, 184]. In this way, platoons of vehicles are formed that can pass the intersection collectively.

There is a natural tension between capacity and fairness. One of the fairest switching rules is to let vehicles pass the intersection in order of arrival (on an intersection wide basis). This rapidly becomes unsustainable, because each switch requires an additional clearance time, which decreases the capacity of the intersection. In near-saturation conditions, when the flows together impose a high volume-to-capacity ratio, the loss of capacity due to switching will have a dramatic effect on delays. Our PFAs aim to balance capacity and fairness.

In PFAs, vehicles arriving at the intersection arrange themselves in platoons, not adapting their relative position to other vehicles on the same lane but adapting their speed. The key feature is that cars, while approaching the intersection, adjust their speeds and upon arrival at the intersection are at high speed, occupying the conflict area of the intersection as briefly as possible. In this way, time bans to give way to other traffic flows still exist, but the platoons are processed in the quickest possible way, because the size and speed of the platoons, of all directions, are organized by the PFA. This is, to some extent, also the purpose in e.g. [66], where a small-scale experiment in Helmond is described.

PFAs are one particular example of the "slower is faster" effect, which is also observed in e.g. [92] and [93], where, perhaps counter-intuitively, slowing down early results in less delay on average in the future. Moreover, this phenomenon results in environmental advantages as less braking-and-pulling-up-again is needed and cars reach their destination more quickly.

The importance of intersection access algorithms has been recognized for several years. Examples of PFAs can be found in [184], which introduces a batch formation algorithm based on arrival times of vehicles and a maximum batch size, and in [133, 134], which use an approach based on polling models. Polling models have a long tradition in communication networks, but the au-

thors in [133] have shown that they can be leveraged to organize autonomous vehicles at intersections as well. One of the key questions in polling models is how to decide which queue should be served (and how many customers should be served before advancing to the next queue). This is exactly one of the main topics of this chapter, where we develop algorithms that determine how to construct platoons of autonomous vehicles and when to give each platoon access to the intersection. A Speed Profile Algorithm (SPA) provides the key link between the PFAs and polling models, as we will show in more detail later.

The area of application of PFAs is not restricted to intersections. There are numerous practical examples where PFAs could be used to achieve a good performance. An example in traffic would be the merging of different streams of vehicles (discussed in e.g. [170]). Another possible application can be found in automated guided vehicles (AGVs) systems, where AGVs may have conflicting routes or have to merge, see e.g. [115] where similar ideas are used.

The main contributions in this chapter can be formulated as follows:

(i) We introduce several new Platoon Forming Algorithms (PFAs), based on enhanced polling policies, that perform well regarding mean delay.

(ii) We also introduce a new class of Speed Profile Algorithms (SPAs). SPAs ensure an efficient use of the intersection, by optimizing the trajectory of (platoons of) vehicles driving towards the intersection, ensuring the arrival at their designated times.

(iii) Employing those SPAs, a link between polling models and PFAs is established, making it possible to conduct a performance analysis. Using interpolation techniques from [22] we develop accurate approximations for the mean delay for the studied PFAs.

(iv) A notion of *fairness* of a PFA is introduced in this chapter. Fairness in queueing models (and therefore PFAs) is important in the perception of customers (or drivers), see e.g. [165]. We use the definition of fairness as given in [176] to assess the fairness of the various PFAs.

(v) Furthermore, we provide a comparison between the performance of traditional traffic technologies and PFAs through simulations in SUMO.

Chapter outline

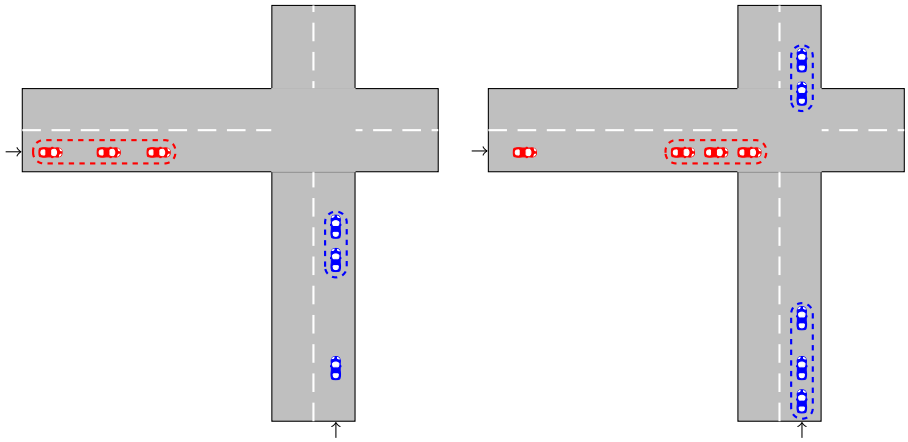
This chapter is organized in the following way. We start with a description of the various ingredients of the model and provide an extensive description of the

new PFAs that we introduce in Sections 7.2 and 7.3 respectively. Section 7.4 is devoted to SPAs. Afterwards, in Section 7.5, we revisit polling models and show a link between PFAs and polling models that enables us to give a performance analysis for PFAs based on results for polling models. Subsequently, Section 7.6 provides a comparison between the traditional traffic light (represented by simulations in SUMO) and our PFAs, focusing on mean delay, and we wrap up with some conclusions in Section 7.7.

7.2 Model formulation

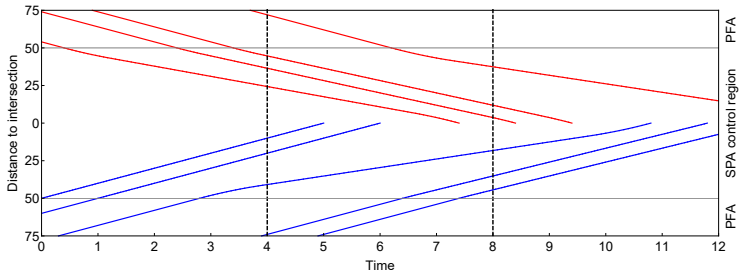
We will consider models in which autonomous vehicles are crossing an intersection. We assume the existence of a control region around the intersection with at the center a centralized controller communicating with all vehicles within the control region. In fact, this control region can be divided into two sub-regions: the inner part is called the "SPA control region". As soon as a vehicle enters this part of the control region, its trajectory is determined by the speed profiling algorithm. In the outer part, which we call the "PFA control region", the access time of each of the arriving vehicles to the intersection is determined. The reason why we need separate control regions for the PFA and the SPA is that we need the trajectory to be fixed once a vehicle enters the SPA control region. Inside the PFA control region, vehicle access times may be adjusted due to the arrival of other vehicles. Indeed, in the PFA control region, the central controller creates platoons of vehicles by scheduling the crossing times of the vehicles according to some policy (the PFA) in such a way that every vehicle is able to cross the intersection at its designated time. We assume that we can control the speed of a vehicle and do so in such a way that the intersection is used efficiently. We make sure that vehicles drive at maximum speed at the moment that they start crossing the intersection, using ideas introduced in [133]. Instead of stopping at the stop line and still having to accelerate when crossing the intersection, a vehicle is already slowed down before it reaches the intersection and starts accelerating again, such that it is driving at full speed when reaching the conflict area of the intersection. This, among others, implies that the time to cross the intersection is the same for each vehicle. The last assumption discussed here, is that we assume that the central controller can look "ahead" for the same amount of time for each of the lanes, to ease the notation and algorithms.

We clarify how this works in a simple example, depicted in Figure 7.1. For simplicity, we show vehicles arriving from only two different approaches (marked red and blue). The central controller uses a PFA to compute the ac-



(a) Intersection at time $t = 4$.

(b) Intersection at time $t = 8$.



(c) Trajectories for the red and blue traffic flows.

Figure 7.1: A schematic representation of the model discussed in this chapter. The platoon forming algorithms in this chapter determine how the platoons are constructed. In the next step, a speed profiling algorithm determines how each individual vehicle approaches the intersection. Figures (a) and (b) correspond, respectively, to the situation in (c) at times $t = 4$ and $t = 8$ seconds.

cess times to (the conflict area of) the intersection for each vehicle entering the control region. The intersection drawn in Figures 7.1(a) and (b) only depicts the inner (SPA) part of the control region. Figure 7.1(c) shows the corresponding trajectories. Note that all vehicles drive at full speed in the PFA control area (from 75 – 50 meters distance) and start their trajectories controlled by the SPA at 50 meters distance. The two parts of the control region are separated by a gray line. The blue vehicle entering the SPA control region at time $t = 0$

encounters no hinder from other vehicles and proceeds at full speed, without delay. The first red vehicle was originally scheduled to arrive at the intersection directly after the first blue vehicle. When, however, the second blue vehicle entered the PFA control region at $t = 1$ (probably arriving in a platoon from an upstream intersection), this blue vehicle is allowed to join the platoon started by the previous blue vehicle. This means that the first red vehicle is rescheduled, being delayed, hence it gets access to the intersection *after* the second blue vehicle at a safe distance. Due to this delay, the next two red vehicles are able (and allowed) to join the red platoon. The actual trajectories towards the intersection are determined by the SPA, which ensures an efficient usage of the intersection. Note that all vehicles cross the intersection at full speed.

An advantage of the control region, besides the ability to control the speed of arriving vehicles, is that we can adjust the scheduling of the vehicles based on the arriving vehicles that are not yet at the intersection. This specific anticipation is key to the forming of platoons and is up to the central controller at the intersection and results in a specific PFA. There are many PFAs, yet we will specifically focus on PFAs that find their origin in polling models, because they are efficient, well understood, and have proven their value in other application areas, such as communication systems and production lines.

7.3 Platoon forming algorithms

We present our new PFAs as standalone algorithms, based on service disciplines for polling models, which are described in a way fit for PFAs. We also briefly discuss the Batch Algorithm, originating from [184], which serves as a benchmark for our PFAs. The PFAs we discuss, are all derived from so-called branching-type disciplines, which find their origin in the polling literature, see e.g. [168]. Branching-type service disciplines include the exhaustive and the gated discipline, which all allow for many analytical results.

Before we start with the description of the PFAs we introduce some concepts and notation. The PFA determines the crossing time of each of the vehicles in the control region that have not yet crossed the intersection. We represent this schedule by entities that we call "vehicles". A vehicle V has three properties: a lane d_V, an earliest crossing time a_V, and the currently scheduled crossing time c_V. We assume that at every point in time we have such a list of vehicles, ordered on basis of the c_V's. The PFA updates (some of) the crossing times of the vehicles upon arrival and departure epochs of vehicles in the PFA control region. The latter is dealt with in an easy way: if the current time is $c_V + B$,

where B denotes the difference in crossing times between two vehicles on the same lane, then vehicle V is removed from the ordering because vehicle V just crossed the intersection. We further assume that there are vehicles arriving at the intersection from n lanes.

Turning towards arrivals of vehicles within the PFA control region, we need to consider the crossing times of all vehicles already scheduled in order to schedule V. There are several ways to schedule those vehicles and the first we discuss is the *exhaustive discipline*, as described in Algorithm 7.1. An intuitive explanation of the exhaustive discipline is the following: if a vehicle that arrives in the control region is able to get within B seconds of the vehicle in front of it on the same lane (which might occur if the vehicle is delayed by its predecessor), it is allowed to join the same platoon as its predecessor. This would imply that all vehicles on different lanes have to wait an additional B seconds, the difference in crossing time between two vehicles. If a vehicle cannot join the platoon in front of it, it will form a new platoon. If no vehicle (on the current lane) is able to join the platoon currently crossing the intersection, a platoon of vehicles at the next lane may cross the intersection. As a result we have a cyclic structure of departures of platoons, because we, in a fixed order, sequentially check each lane for vehicles that want to depart. This exhaustive discipline is known for its low mean delay, which is the main reason for us to consider this discipline. We further introduce one more constant, S, that represents the time between the start of crossing of two vehicles on different lanes (similar to clearance times at intersections nowadays).

Although the exhaustive PFA will have very good delay characteristics, we will consider the *gated PFA* (discussed below) as well. The intuitive explanation of the gated algorithm is quite close to that of the exhaustive discipline, with one exception. It is not always allowed to join a platoon, even if a vehicle is able to get within B seconds from its predecessor on the same lane. As described in more detail below, platoons are finalized at an earlier moment than with exhaustive service. This moment of finalizing a platoon is, in the polling literature, compared to putting a gate behind the last customer (corresponding to the last vehicle in the platoon). Newly arriving customers will have to wait (behind the virtual gate) for the next server visit, which corresponds to the formation of a new platoon in our setting. An advantage of the gated discipline is that there is less variation in the size of platoons and, hence, cycle lengths are less variable as well. It may result in longer delays though, as we will see in the numerical examples in Section 7.5.

For the implementation of the gated PFA, we need to keep track of a couple of additional variables for each lane. In this gated discipline we are namely

Algorithm 7.1 exhaustive algorithm.

Input: current ordering of vehicles, denoted $(V_1, V_2, \ldots, V_K)$, ordered on basis of c_V; V_{last}, defined as V_K or the last vehicle that crossed the intersection if the ordering is empty; and a to be scheduled vehicle V_0 with earliest arrival time at the intersection a_{V_0} in lane d_{V_0}.
if $c_{V_{last}} + B < a_{V_0}$ **then** ▷ V_0 is scheduled last
  if $d_{V_0} = d_{V_{last}}$ **then**
    Put $c_{V_0} \leftarrow a_{V_0}$. ▷ V_0 proceeds without delay
  else
    Put $c_{V_0} \leftarrow \max\{a_{V_0}, c_{V_{last}} + S\}$. ▷ Check if additional clearance time is needed
  end if
else
  Put $t_i \leftarrow \begin{cases} c_{L_i} & \text{where } L_i \text{ is last scheduled vehicle in lane } i, \\ -\infty & \text{if lane } i \text{ is empty and no such vehicle exists.} \end{cases}$
  if $t_{d_{V_0}} + B > a_{V_0}$ **then** ▷ V_0 is able to join a platoon
    Put $c_{V_0} \leftarrow t_{d_{V_0}} + B$.
    for each vehicle V in the ordering with $c_V > t_{d_{V_0}}$ **do**
      Put $c_V \leftarrow c_V + B$. ▷ Delay other vehicles
    end for
  else
    for l in $(d_{V_0} - 1, d_{V_0} - 2, \ldots, 1, n, n-1, \ldots, d_{V_0} + 1)$ **do**
      if $t_l + S > a_{V_0}$ **then** ▷ V_0 starts new platoon after last platoon in lane l
        Put $c_{V_0} \leftarrow t_l + S$.
        for each vehicle V in the ordering with $c_V > t_l$ **do**
          Put $c_V \leftarrow c_V + S$. ▷ Delay other vehicles
        end for
        break
      end if
    end for
  end if
end if
Add vehicle V_0 to the ordering.
Output: the new ordering $(V_1, V_2, \ldots, V_0, \ldots, V_K)$

"putting gates" which can be seen as "fixing the vehicles of a platoon", meaning that future arrivals in the same lane cannot join the currently formed platoon (i.e. they are "behind the gate"). We define two additional, ordered sets for each lane, f_i and t_i, representing the set of start times of platoons on lane i and the end times of platoons at lane i (so the start of service of the last vehicle).

Joining a platoon is only allowed if the lane is *not* the lane from which vehicles are currently departing (the platoon is not yet fixed). If a car in lane i is able to reach the intersection (without any other interfering traffic) before one of the times in f_i, then that car is allowed to join that platoon (so the platoon is enlarged). If such a car is not able to reach the intersection before one of the times in f_i, then it creates a new platoon. In general, departures of vehicles are dealt with in the same way as in the exhaustive discipline. We again have the cyclic structure as in the exhaustive discipline. The gated algorithm can then be described as in Algorithm 7.2.

PFAs in terms of polling models

Algorithms 7.1 and 7.2 are rather complicated and lengthy. However, the underlying intuition/description in terms of polling models is rather straightforward as we explain here.

For the exhaustive discipline, we have the following description in standard polling terminology: as long as a queue is not empty, the server stays at that queue and keeps serving customers. In terms of PFAs and self-driving vehicles, this translates to the following: as many vehicles as possible are added to a platoon. The only reason why the next vehicle would not be able to join the platoon in front of it is because it is not able to arrive at the intersection at a time B after its predecessor, even when that vehicle would drive at full speed. The platoon is thus finalized as soon as no vehicles can join it anymore, in which case "the queue is empty" and we switch to the next lane.

We might explain the gated discipline in the following way: if the server starts working on customers in a queue, the server will stay at that queue until the moment that all customers that were present at the start of service of the first customer, have left. I.e., it is as if a gate is put behind the last customer present in the queue when the server arrives and only customers in front of this gate are served before the server switches to the next queue. In PFA terms, this means that a platoon is finalized at the moment that the first vehicle of the platoon starts crossing the intersection. Every vehicle that has been able to join the platoon at that moment is allowed to cross the intersection in the same platoon. Every vehicle that could not join the platoon at that moment, needs to wait for a full cycle.

Algorithm 7.2 gated algorithm.

Input: current ordering of vehicles, denoted $(V_1, V_2, \ldots, V_K)$, ordered on basis of c_V; V_{last}, defined as V_K or the last vehicle that crossed the intersection if the ordering is empty; the sets f_i and t_i for $i = 1, \ldots, n$ representing the start of platoons and end of platoons at lane i; and a to be scheduled vehicle V_0 with earliest arrival time at the intersection a_{V_0} in lane d_{V_0}.
if $c_{V_{last}} + B < a_{V_0}$ **then** ▷ V_0 is scheduled last
  if $d_{V_0} = d_{V_{last}}$ **then**
    Put $c_{V_0} \leftarrow a_{V_0}$. ▷ V_0 proceeds without delay
  else
    Put $c_{V_0} \leftarrow \max\{a_{V_0}, c_{V_{last}} + S\}$. ▷ Check if additional clearance time is needed
  end if
  Add time c_{V_0} to $f_{d_{V_0}}$ and time c_{V_0} to $t_{d_{V_0}}$.▷ Register c_{V_0} as start of a new platoon
else
  if there is a time in $f_{d_{V_0}} > a_{V_0}$ **then** ▷ V_0 is able to join a platoon
    Put $f \leftarrow$ the lowest time in $f_{d_{V_0}}$ such that $f > a_{V_0}$.
    Put $t \leftarrow$ the corresponding end of platoon in $t_{d_{V_0}}$.
    Put $c_{V_0} \leftarrow t + B$.
    for each value t^* in $t_1, \ldots, t_n$ with $t^* > t$ **do** ▷ Update t and f
      Put $t^* \leftarrow t^* + B$
      Put the corresponding start of platoon $f^* \leftarrow f^* + B$.
    end for
    for each vehicle V in the ordering with $c_V > c_{V_0}$ **do**
      Put $c_V \leftarrow c_V + B$. ▷ Delay other vehicles
    end for
  else
    for l in $(d_{V_0} - 1, d_{V_0} - 2, \ldots, 1, n, n-1, \ldots, d_{V_0} + 1)$ **do**
      if there is a time in $t_l + S > a_{V_0}$ **then** ▷ V forms a new platoon
        Find the lowest time t in t_l such that $t + S > a_{V_0}$.
        Put $c_{V_0} \leftarrow t + S$.
        if there is a time in f_l such that $t = f_l$ **then**
          for each value t^* in $t_1, \ldots, t_n$ with $t^* > t + S$ **do** ▷ Update t and f
            Put $t^* \leftarrow t^* + 2S$
            Put the corresponding start of platoon $f^* \leftarrow f^* + 2S$.
          end for
          for each vehicle V in the ordering with $c_V > t + S$ **do**
            Put $c_V \leftarrow c_V + 2S$. ▷ Delay other vehicles
          end for
        else ▷ V_0 is able to join a platoon
          for each value t^* in $t_1, \ldots, t_n$ with $t^* > t + S$ **do** ▷ Update t and f
            Put $t^* \leftarrow t^* + S$
            Put the corresponding start of platoon $f^* \leftarrow f^* + S$.
          end for

```
                    for each vehicle V in the ordering with c_V > t + S do
                        Put c_V ← c_V + S.                    ▷ Delay other vehicles
                    end for
                end if
                Add time c_{V_0} to f_{d_{V_0}} and c_{V_0} to t_{d_{V_0}}.
                break
            end if
        end for
    end if
end if
if c_{V_0} is undefined then
    Put c_{V_0} ← c_{V_K} + B.
    Add time c_{V_0} to f_{d_{V_0}} and time c_{V_0} to t_{d_{V_0}}.
end if
Add vehicle V_0 to the ordering.
Output: the new ordering (V_1, V_2, ..., V_0, ..., V_K)
```

As a reference to algorithms so far established in the literature, we also consider the Batch Algorithm from [184]. For the full description we refer to [184, Supplementary Information, Section 1.5]. The Batch Algorithm consists of a combination of a gated PFA (also in the Batch Algorithm "gates" are put) and a maximum number of vehicles that is dealt with in each cycle.

7.4 Speed profile algorithms

Now that we know how to schedule the crossing times of vehicles at the intersection, we turn to the other key ingredient of our model, which is the speed control of arriving vehicles. We start with some requirements that the PFAs have to satisfy before we can control the speed of the arriving vehicles in a proper and safe way. The main condition a PFA has to satisfy is *regularity*.

Definition 7.1 (Regularity [133, 134]) *A polling policy is* regular *if an arrival in a queue does not change the* order *of service of all currently present vehicles. I.e. the new arrival is inserted somewhere in the order of service of all waiting vehicles.*

A regular PFA ensures that if a vehicle is rescheduled, its crossing time is *increased*. A decrease would potentially lead to a scheduled crossing time at which the vehicle cannot be at the intersection (e.g. due to the fact that the vehicle has decelerated and cannot accelerate quickly enough to reach the intersection

in time). The exhaustive and gated algorithms discussed in Algorithms 7.1 and 7.2 are examples of regular polling policies, because a new arrival does not change the order of service of the vehicles that are already scheduled and, as a consequence, the crossing times of vehicles can only increase.

A regular polling policy, together with assuming a sufficiently big control region, ensures that the intersection coordination algorithm in [133, 134] and the speed profile algorithms that we will introduce are solvable. As mentioned before, these assumptions are necessary with respect to the (possibility of) vehicles being rescheduled. As can be seen in Algorithms 7.1 and 7.2, the access time of (some of the) vehicles to the intersection might be increased. The above assumptions ensure that we can find feasible and safe trajectories for every vehicle, also in case of rescheduling, cf. [133, 134].

Besides these two assumptions on regularity and the size of the control region, we also need to make sure that there are not too many vehicles in the control region at the same time: if there are too many vehicles present in the control region, it might be the case that a newly arriving vehicle cannot decelerate to a complete stop in time. In this case, the distance between entering the control region and the stopping position of its predecessor is too short. This phenomenon is called *overcrowding*, see [134]. A way to deal with this issue is proposed as well: we assume that a vehicle that cannot enter the control region safely, does not enter the control region at all cf. [134].

7.4.1 Optimization based speed profile algorithms

In this subsection, we discuss two algorithms that, satisfying the above conditions, result in an efficient use of the intersection, which is our main purpose. To this end, we require that vehicles drive at maximum speed while crossing the intersection, so we need to control the speed of arriving vehicles while they are in the control region. An optimization algorithm can be formulated to achieve this as is shown in [134, the MotionSynthesize procedure]. In order to solve this minimization problem, time is discretized. The MotionSynthesize procedure is then reduced to a linear optimization problem for which efficient solvers exist.

The optimization procedure has several nice properties, among which is that the algorithm is provably safe. A formal definition of “safe” and the required conditions (such as “no overcrowding”) are given in [134], but intuitively it simply means that no collisions will occur in the control region.

Another property of the MotionSynthesize procedure is that the distance between vehicle and intersection is minimized across the whole time period that a vehicle is in the control region. This is equivalent with the minimization of

the area under the distance-time diagram, where the distance is defined as the distance between vehicle and intersection. The physical length of the queue of vehicles is thus also minimized. This is favorable in a network setting, minimizing the amount of spillback to other intersections. Yet, this specific property of minimizing the distance between vehicle and intersection has a high energy consumption and may not be very pleasant for passengers.

Below, in Algorithm 7.3, we discuss a slightly different formulation of the problem where we minimize the total amount of the absolute value of the *acceleration* instead of the *distance* between vehicle and intersection. We do this, because this would result in less energy consumption by vehicles driving towards the intersection and because the ride towards the intersection is more comfortable in comparison with minimizing the distance between the vehicle and the intersection. However, assuming regularity of the PFA and a sufficiently big control region is not sufficient to ensure a feasible optimization problem as it is for the MotionSynthesize procedure. We formulate a mild additional constraint to guarantee feasibility of the optimization problem, which is that one needs to be sure that when the preceding vehicle is done decelerating, the next vehicle is able to decelerate to that same speed before the preceding vehicle is decelerating further (due to rescheduling for example). As will turn out, a vehicle starts decelerating immediately after entering the control region (see e.g. Figure 7.3). As a consequence, if a vehicle is entering the control region, it needs to be sure that it is able to decelerate to the speed of its predecessor while maintaining a certain distance to its predecessor at the same time, showing that we need this additional assumption.

Before we turn to the algorithm, we introduce some notation. Each vehicle has a trajectory that is computed along the lines of the algorithm, given the current time, t_0, and the scheduled crossing time t_f (in this section, for consistency with [134], we use the notation t_f to denote the scheduled crossing time, instead of c_V). The algorithm will compute $x(t)$, the place of the vehicle at time t, for $t_0 \leq t \leq t_f$, the speed $v(t)$ at time t, and the acceleration $a(t)$ at time t. Furthermore, $y(t)$ denotes the trajectory of the predecessor (if any) for $t_0 \leq t \leq t_{f,y}$; $t_{f,y}$ denotes the final crossing time of the predecessor of the vehicle we are currently planning; l denotes the minimal distance between the front part of two successive vehicles; a_m denotes the maximum acceleration; $-a_m$ denotes the maximum deceleration; and v_m denotes the maximum speed. The initial conditions, i.e. the location and speed at the start of the trajectory of the vehicle, are given by $x(t_0) = x_0$ and $v(t_0) = v_0$. To put the location $x(t)$ into perspective, we measure $x(t)$ as the (negative) distance between the vehicle and the start of the conflict area of the intersection, i.e. $x(t_0) = x_0 = -X$ and $x(t_f) = 0$,

when the vehicle enters the control region at a distance X from the intersection. Then, we are able to formulate Algorithm 7.3.

Algorithm 7.3 MotionSynthesize procedure with a minimal acceleration

1: Input: x_0, v_0, t_0, t_f,$t_{f,y}$, y.
2: Compute

$$\text{MotionSynthesizeAcc}(x_0, v_0, t_0, t_f, t_{f,y}, y) := \underset{x:[t_0,t_f]\to\mathbb{R}}{\arg\min} \int_{t_0}^{t_f} |a(t)|\,\mathrm{d}t$$

subject to
$x''(t) = a(t)$, for all $t \in [t_0, t_f]$;
$0 \le x'(t) \le v_m$, for all $t \in [t_0, t_f]$;
$|a(t)| \le a_m$, for all $t \in [t_0, t_f]$;
$|x(t) - y(t)| \ge l$, for all $t \in [t_0, t_{f,y}]$;
$x(t_0) = x_0$; $x'(t_0) = v_0$;
$x(t_f) = 0$; $x'(t_f) = v_m$.

3: Output: $x(t)$.

Algorithm 7.3 can be discretized in order to obtain a linear optimization problem, just as the MotionSynthesize procedure and has a valid solution under the set of conditions formulated above, i.e. regularity of the PFA, a sufficiently big control region, and the assumption on decelerating of the predecessor of a vehicle. The main difference between Algorithm 7.3 and the MotionSynthesize procedure from [134] is that instead of minimizing the distance from vehicle to intersection, we minimize the (absolute value of the) acceleration applied by the vehicle while being in the control region. This obviously has consequences for the amount of energy consumption. Disadvantages include that the physical length of the queue grows and that vehicles cannot enter the control region as close to each other (as vehicles slow down immediately when entering the control region).

In the next subsection we present closed-form alternatives to the MotionSynthesize procedure and Algorithm 7.3, similar in spirit as the results in e.g. [69, 118]. So instead of the need to solve a linear optimization problem each time, we have a set of calculations that we can perform to find the trajectory of a vehicle, which is optimal with respect to minimizing the distance or acceleration.

These closed-form expressions immediately imply that Algorithm 7.3 yields a valid and safe trajectory. In Remark 7.3 we return to this topic.

7.4.2 Closed-form speed profile algorithms

In this subsection, we derive closed-form alternatives to the MotionSynthesize procedure in [134] and to Algorithm 7.3. We start with the MotionSynthesize procedure and make two important observations that form the basis for our closed-form SPA:

(i) The optimization problems formulated in the MotionSynthesize procedure and Algorithm 7.3 always lead to piece-wise constant acceleration;

(ii) If all vehicles decelerate (and possibly stop) at most *once*, at most four changes in the acceleration occur.

These observations imply that if we can find the four points at which the acceleration changes, we are able to determine the trajectory in closed form. We note that the exhaustive and gated algorithms indeed have the desirable property that vehicles need to decelerate at most once. From the polling literature we know that the exhaustive service discipline ensures that customers will always be served before the end of the cycle in which they arrive. With gated service, customers will always be served in the next cycle. Translated to our traffic model, this means that no vehicle will ever need to stop more than once. As a consequence, the acceleration changes at most four times. We shortly describe the corresponding five parts of the arriving trajectory.

- No acceleration or deceleration from t_0 until t_{dec};
- Deceleration at maximum rate from t_{dec} until t_{stop};
- A stop from t_{stop} until t_{acc};
- Acceleration at maximum rate from t_{acc} until t_{full};
- No acceleration or deceleration from t_{full} until t_f.

We note that some of those time points might coincide with each other. All that remains is that we have to find t_{dec}, t_{stop}, t_{acc}, and t_{full} in such a way that we minimize the average distance between the vehicle and the intersection. This leads to Algorithm 7.4, where we assume that $t_0 = 0$ to ease the notation and that $v_0 = v_m$. We can allow for general v_0, but we show later that this would

Algorithm 7.4 closed-form alternative to the MotionSynthesize procedure.

Input: x_0, t_f, $t_{f,y}$, and y.
if $t_f - t_{f,y} = B$ **then**
    Consider trajectory y and determine the time at which the vehicle continues at full speed. Call this time t_{full}.
else
    Put $t_{full} \leftarrow t_f$.
end if
Put

$$L \leftarrow v_m \left(t_f - \frac{v_m}{a_m} \right).$$

▷ L represents the distance covered if a vehicle stops for 0 seconds
if $L \geq |x_0|$ **then** ▷ The vehicle has to stop
    Put $t_{acc} \leftarrow t_{full} - v_m / a_m$.
    Put $t_{stop} \leftarrow t_{acc} - (t_f - v_m / a_m - |x_0| / v_m)$.
    Put $t_{dec} \leftarrow t_{stop} - v_m / a_m$.
else ▷ The vehicle does not have to stop
    Define

$$\tilde{t} \leftarrow \sqrt{\frac{t_f v_m - |x_0|}{a_m}}. \tag{7.1}$$

▷ $\tilde{t}$ is the deceleration time
    Put $t_{acc} \leftarrow t_{full} - \tilde{t}$.
    Put $t_{stop} \leftarrow t_{acc}$.
    Put $t_{dec} \leftarrow t_{acc} - \tilde{t}$.
end if
Then

$$a(t) = x''(t) \leftarrow \begin{cases} 0 & \text{if } 0 \leq t < t_{dec}, \\ -a_m & \text{if } t_{dec} \leq t < t_{stop}, \\ 0 & \text{if } t_{stop} \leq t < t_{acc}, \\ a_m & \text{if } t_{acc} \leq t < t_{full}, \\ 0 & \text{if } t_{full} \leq t < t_f. \end{cases} \tag{7.2}$$

Knowing $a(t)$, we can compute $x(t)$ by integrating twice and using the conditions $x(0) = x_0$ and the velocity at time 0 being v_m.
Output: $x(t)$.

always result in a sub-optimal trajectory. The input consists of the (negative) distance between vehicle and intersection at $t = 0$, again denoted by x_0, the scheduled crossing time of the vehicle, t_f, and the trajectory of the predecessor of the vehicle for which we are currently planning the trajectory, y, and its crossing time $t_{f,y}$. We prove that the MotionSynthesize procedure and Algorithm 7.4 are equivalent, which is the subject of the next lemma.

Lemma 7.1 *The MotionSynthesize procedure and Algorithm 7.4 are equivalent in the sense that both minimize the distance between vehicle and intersection across the time period t_0 to t_f.*

Proof. We split the proof in two parts. First we prove that the times t_{dec}, t_{stop}, t_{acc}, and t_{full} in Algorithm 7.4 indeed result in the trajectory having the minimal area under the distance-time graph, *assuming that* the optimal trajectory contains at most one period of deceleration. Then we prove that the obtained *form* of the trajectory, with at most one period of deceleration, is indeed optimal.

Part 1. As indicated before, for now, we only consider trajectories that contain at most *one* period of deceleration. We allow that $v_0 < v_m$ (but we will show now that that is suboptimal), but we do require that $v(t_{full}) = v(t_f) = v_m$. We distinguish between the case where a vehicle comes to a full stop and the case where it does not.

Full stop. First we consider the case where the vehicle (denoted by V) comes to a full stop, from $t = t_{stop}$ to $t = t_{acc}$. This class of trajectories is visualized as the black line in Figure 7.2. It turns out that this curve is completely characterized by two parameters, which we choose to be the initial speed v_0 and the moment when we start driving at full speed again, t_{full}.

The optimization criterion in the MotionSynthesize algorithm is to minimize the area below the graph $|x(t)|$ for $0 \le t \le t_f$. This is equivalent to minimizing the average distance to the intersection. First we give an intuitive explanation as to why it makes sense to continue at full speed as long as possible. In Figure 7.2 we have plotted two alternative trajectories to show that they result in a larger average distance to the intersection. The red dashed trajectory is equivalent to the optimal trajectory, but with a lower starting speed ($v_0 < v_m$). By starting at a lower speed, while fixing t_{full}, we have to continue longer at this lower speed before we come to a complete stop. This means that t_{dec} and t_{stop} increase, which immediately increases the area below the graph. Another alternative is the dashed green trajectory, which starts at full speed, but has a lower value

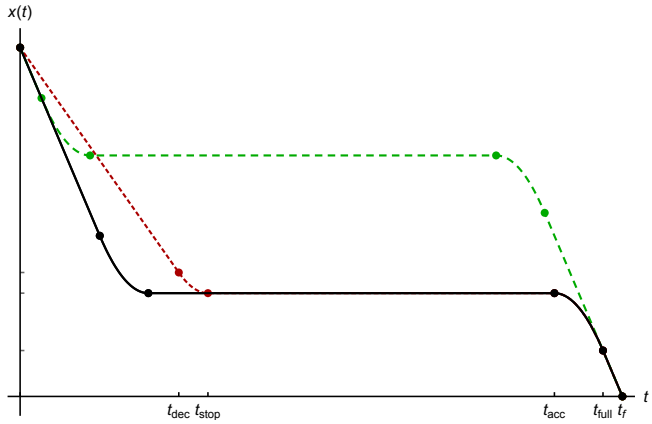


Figure 7.2: Three sample trajectories with one full stop. The optimal trajectory is plotted in black. The dashed green trajectory has a smaller value of t_{full} compared to the optimal trajectory, whereas the dashed red trajectory has a smaller value of v_0.

for t_{full}. Note that t_{full} is restricted by V's predecessor. Without predecessor, it is optimal to take $t_{full} = t_f$, but if there is a predecessor (which apparently is the case for the black trajectory in Figure 7.2), it is optimal to let both vehicles have the same t_{full}. This is the only way to ensure that both vehicles cross the intersection at full speed, with minimum distance between them. Taking a smaller value of t_{full}, as in the green trajectory, means that V comes to a stop further from the intersection, which significantly increases the average distance.

These arguments provide an intuitive explanation, but we will formalize this now by explicitly computing the area below $|x(t)|$ for our closed-form trajectories. First we give the closed-form expression for $x(t)$, by considering the five sub-areas separately, and using the fact that $x(t)$ is linear when the speed is constant and quadratic while decelerating/accelerating. Equation (7.3) is easiest to understand when starting at $t = t_f$ and constructing the trajectory backwards to $t = 0$, and using these auxiliary results:

$$t_{stop} - t_{dec} = \frac{v_0}{a_m},$$
$$t_{full} - t_{acc} = \frac{v_m}{a_m},$$

$$x(t_{stop}) - x(t_{dec}) = \frac{v_0^2}{2a_m},$$
$$x(t_{full}) - x(t_{acc}) = \frac{v_m^2}{2a_m}.$$

We obtain:

$$x(t) = \begin{cases} (t - t_f)\, v_m & \text{for } t_{full} \le t \le t_f, \\ (t_{full} - t_f)\, v_m - \frac{v_m^2}{2a_m} + \frac{a_m}{2}(t - t_{acc})^2 & \text{for } t_{acc} \le t \le t_{full}, \\ (t_{full} - t_f)\, v_m - \frac{v_m^2}{2a_m} & \text{for } t_{stop} \le t \le t_{acc}, \\ (t_{full} - t_f)\, v_m - \frac{v_m^2}{2a_m} - \frac{a_m}{2}(t - t_{stop})^2 & \text{for } t_{dec} \le t \le t_{stop}, \\ x_0 + v_0\, t & \text{for } 0 \le t \le t_{dec}. \end{cases} \tag{7.3}$$

Note that t_{dec} follows from continuity of $x(t)$:

$$t_{dec} = \frac{1}{v_0}\left(|x_0| - (t_f - t_{full})\, v_m - \frac{v_0^2 + v_m^2}{2a_m}\right).$$

The area below the trajectory, $\mathscr{A}_v := \int_0^{t_f} |x(t)|\, \mathrm{d}t$, is equal to:

$$\begin{aligned} \mathscr{A}_v &= \frac{t_{dec}}{2}\left(x(t_{dec}) - x_0 + \frac{v_0^2}{a_m}\right) + \frac{v_0^3}{6a_m^2} + \\ &\quad t_{full}\left((t_f - t_{full})\, v_m + \frac{v_m^2}{2a_m}\right) - \frac{v_m^3}{6a_m^2} + \frac{v_m}{2}(t_f - t_{full})^2 \\ &= \frac{v_0^4 + 3\left(v_m^2 + 2a_m((t_f - t_{full})\, v_m + x_0)\right)^2}{24a_m^2 v_0} + \frac{v_m}{2}\left(t_f^2 - t_{full}^2 + t_{full}\frac{v_m}{a_m}\right) - \frac{v_m^3}{6a_m^2}. \end{aligned}$$

We now exploit that only the first part of the expression for $\mathscr{A}_v$ depends on the initial speed v_0, as observed before. By taking the derivative with respect to v_0 and using $v_0 \le v_m$ it follows that $\mathscr{A}_v$ is decreasing in v_0, under the following condition:

$$(t_f - t_{full})\, v_m + 2\frac{v_m^2}{2a_m} \le |x_0|.$$

This is exactly the "no overcrowding" assumption discussed earlier, which now gets quantified: a vehicle entering the control region at full speed should have

enough space to come to a full stop and accelerate again in order to reach full speed at time t_{full}. The above proves that the initial speed should be taken as large as possible, i.e. $v_0 = v_m$.

Now that we have established that we should choose $v_0 = v_m$, we assume this equality from now on and denote the area as $\mathscr{A}$ (to distinguish it from $\mathscr{A}_v$). This significantly simplifies the expression, which now becomes

$$\mathscr{A} = (v_m t_f + x_0)\left(t_f - t_{full} + \frac{v_m}{2a_m}\right) + \frac{x_0^2}{2v_m}.$$

It is readily seen that the area $\mathscr{A}$ is now *linearly decreasing* in t_{full}, which immediately proves that we should take t_{full} as large as possible to minimize $\mathscr{A}$. Exactly how large t_{full} is allowed to be, depends on the predecessor.

No full stop. We now briefly consider the case where V does not come to a full stop. The analysis is quite similar, so we will mainly focus on the differences. The first difference is that t_{stop} is removed from the trajectory. Instead, we now have that the speed at $t = t_{acc}$ is greater than zero. Note that this speed, which we denote by v_1, is less than or equal to v_0, because V decelerates between t_{dec} and t_{acc}. The trajectory $x(t)$ now consists of at most four parts, given by:

$$x(t) = \begin{cases} (t - t_f)v_m & \text{for } t_{full} \le t \le t_f, \\ (t - t_f)v_m + \frac{a_m}{2}(t - t_{full})^2 & \text{for } t_{acc} \le t \le t_{full}, \\ x_0 + v_0 t - \frac{a_m}{2}(t - t_{dec})^2 & \text{for } t_{dec} \le t \le t_{acc}, \\ x_0 + v_0 t & \text{for } 0 \le t \le t_{dec}. \end{cases} \tag{7.4}$$

We can eliminate the unknowns by using the relations

$$t_{acc} - t_{dec} = \frac{v_0 - v_1}{a_m},$$
$$t_{full} - t_{acc} = \frac{v_m - v_1}{a_m}.$$

The requirement that $x(t)$ is continuous in t_{acc} leads to the last equation that can be solved to obtain t_{acc}. The area below $|x(t)|$ can now be computed:

$$\mathscr{A}_v = \frac{v_m}{2}(t_f - t_{acc})^2 + \frac{(v_0 - v_1)^3 - (v_m - v_1)^3}{6a_m^2} - x_0 t_{acc} - \frac{v_0}{2} t_{acc}^2.$$

Eliminating t_{acc} and differentiating with respect to v_1 immediately shows that $\mathscr{A}_v$ is decreasing in v_1. Since we are trying to minimize $\mathscr{A}_v$, we should take v_1

as large as possible, i.e. $v_1 = v_0$. After this substitution, all expressions simplify and it can again be shown that the derivative of $\mathscr{A}$ with respect to v_0 is always less than or equal to zero, where equality is only reached when $t_{acc} = 0$ and there is no other option for V than to accelerate immediately. This means that we should take v_0 as large as possible, which again implies that we should take t_{full} as large as possible, what we also do.

It should be noted that the case $v_0 = v_m$ needs to be considered separately, because if the conditions allow a maximal initial speed, v_1 is completely fixed:

$$v_1 = v_m - \sqrt{a_m(t_f v_m - |x_0|)}.$$

This means that t_{full} does not follow from v_0, but it can be chosen arbitrarily (between the minimum and maximum allowed values). To minimize the distance between the vehicle and the intersection, we thus get

$$t_{acc} = t_{full} - \frac{v_m - v_1}{a_m} = t_{full} - \frac{v_m - \left(v_m - \sqrt{a_m(t_f v_m - |x_0|)}\right)}{a_m} = t_{full} - \tilde{t},$$

with $\tilde{t}$ as defined in Equation (7.1).

Implementation. Algorithm 7.4 is an implementation of the optimal trajectory for the general case. The formulation of the algorithm is slightly different, because we are using the results that v_0 and t_{full} should be as large as possible. As argued above, an upper bound to the time t_{full} is determined by the trajectory y of the predecessor of V, and is fixed. If the crossing times differ a time B, then the time at which the predecessor starts driving at full speed, $t_{f,y}$, should be equal to t_{full} (because we want to take it as large as possible), and otherwise it is simply t_f, which is the way we choose t_{full} in lines 2-6.

Then combining the defined times, we obtain Equation (7.2), which minimizes the area under the distance-time graph. This is exactly the same criterion as we optimize for in the MotionSynthesize procedure. The only thing left to show, is that all other trajectories satisfying the required constraints regarding maximum speed and acceleration, have a larger average distance to the intersection than the one we obtain.

Part 2. This part is significantly shorter, proving that the obtained trajectory is really optimal with respect to the criterion of smallest average distance to the intersection. We remind the reader that we explicitly exploit the property of the polling-based PFAs that each vehicle needs to decelerate (and possibly stop) at

most once. Intuitively, the optimality is quite apparent: in order to minimize the average distance to the intersection, a vehicle entering the control region needs to drive at full speed as long as possible. Assume that $x(t)$ is a trajectory defined by Equation (7.3) with $v_0 = v_m$ and t_{full} as large as possible. We now consider an alternative trajectory $\tilde{x}(t) \neq x(t)$. We compare $x(t)$ with $\tilde{x}(t)$ on the five parts of the trajectory.

- For $0 \leq t \leq t_{dec}$ it is completely obvious that $|\tilde{x}(t)| \geq |x(t)|$, because $\tilde{x}(0) = x(0) = x_0$ and $\tilde{x}'(t) \leq x'(t) = v_m$ for $0 \leq t \leq t_{dec}$.
- We now turn to the *last* part of the trajectory. For $t_{full} \leq t \leq t_f$, we have $\tilde{x}(t) = x(t)$ because t_{full} was defined as the largest possible value for t where V should start driving at full speed.
- Looking at the part before this one, $t_{acc} \leq t \leq t_{full}$, we see that $|\tilde{x}(t)| \geq |x(t)|$ because $\tilde{x}'(t_{full}) = x'(t_{full}) = v_m$ and $\tilde{x}''(t) \leq x''(t) = a_m$.
- The period $t_{stop} \leq t \leq t_{acc}$ is also trivial, because $\tilde{v}(t) \geq v(t) = 0$ here, meaning that $|\tilde{x}(t)| \geq |x(t)|$.
- This leaves us with the last part, which is the second period $t_{dec} \leq t \leq t_{stop}$. We have already established that $|\tilde{x}(t_{dec})| \geq |x(t_{dec})|$ and $|\tilde{x}(t_{stop})| \geq |x(t_{stop})|$. Since $\tilde{x}'(t_{dec}) \leq x'(t_{dec}) = v_m$ and $\tilde{x}''(t) \leq x''(t) = a_m$, it also follows that $|\tilde{x}(t)| \geq |x(t)|$ in this area.

The conclusion is that for all $t \in [0, t_f]$ we have $|\tilde{x}(t)| \geq |x(t)|$, which implies that

$$\int_0^{t_f} |\tilde{x}(t)| \, \mathrm{d}t \geq \int_0^{t_f} |x(t)| \, \mathrm{d}t.$$

This proves that the path $x(t)$ is optimal with respect to the criterion of the MotionSynthesize procedure. Since it has also been proven in [133] that the MotionSynthesize algorithm yields an optimal path, both algorithms must return the same path. □

Remark 7.1 *The astute reader will notice that we do not provide an explicit expression for $x(t)$ in Algorithm 7.4. Instead, we provide its second derivative, $a(t)$, and the boundary conditions. This has the advantage that we have one formulation that is valid for both cases (full stop and no full stop). One can easily verify that Equation* (7.3) *(full stop) and Equation* (7.4) *(no full stop) both reduce to Equation* (7.2) *after differentiating twice, and that t_{dec}, t_{stop}, t_{acc}, and t_{full} as computed in Algorithm 7.4 correspond to the values discussed in the first part of the proof. Note that we choose $t_{stop} = t_{acc}$ in the case of no full stop.*

Remark 7.2 *Although the exhaustive and gated PFAs ensure that there is at most* one *period of deceleration, for other disciplines, like the Batch Algorithm or the k-limited discipline, this might not be the case. The period from t_0 until t_f might have to be split in more than five different periods. A similar type of speed profile algorithm is still possible, but is more involved and therefore omitted in the interest of space and clarity of the algorithm and argumentation.*

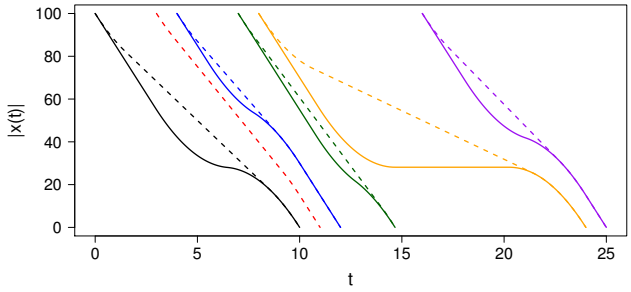


Figure 7.3: Algorithm 7.4 (solid lines) and Algorithm 7.5 (dashed lines) for several vehicles with t (sec) on the horizontal axis and $|x(t)|$ (meters) on the vertical axis for several vehicles.

So, Algorithm 7.4 has the same desirable properties as the MotionSynthesize procedure, but is computationally much less expensive and also provides intuition on the shape of the trajectories. A visualization of such trajectories can be found in Figure 7.3 (represented by the solid lines).

We can also formulate such an alternative for Algorithm 7.3, where we, again, put $t_0 = 0$ to ease the notation. We allow for general v_0 now. In fact, this is essential to this algorithm, because a vehicle might start decelerating immediately upon arrival in the SPA part of the control region. We assume that a following vehicle has decelerated accordingly, if necessary, in the PFA part of the control region. In practice, either vehicle-to-vehicle or vehicle-to-controller communication might be used to ensure this speed adjustment. The general structure of Algorithm 7.3 is similar to that of Algorithm 7.4. Also in this case, the acceleration is piece-wise constant, yet there are at most three changes in the acceleration. We shortly describe those four parts of the arriving trajectory.

- Deceleration at maximum rate from t_0 until t_{cruise};
- No acceleration or deceleration from t_{cruise} until t_{acc};

- Acceleration at maximum rate from t_{acc} until t_{full};
- No acceleration or deceleration from t_{full} until t_f.

This is also visible in Figure 7.3, where a visualization of some trajectories computed with Algorithm 7.5 is given (represented by the dashed lines). Note that we start decelerating as soon as possible, because we want to cruise at a relatively low speed. If we would not cruise at a low speed, then we would have to decelerate more (as we covered a longer distance at a high speed). So we decelerate maximally for some time, continue at a constant speed for some time and then accelerate maximally (taking advantage of the lower cruising speed as long as possible). The resulting algorithm is formulated in Algorithm 7.5 and equivalence with Algorithm 7.3 is proven.

Lemma 7.2 *Algorithm 7.3 and Algorithm 7.5 are equivalent in the sense that both minimize the absolute value of the applied acceleration across the time period* t_0 *to* t_f.

Proof. We again split the proof in two parts, but now we first prove optimality of the form of the trajectory and then we check the computation of t_{cruise}, t_{acc}, and t_{full} in Algorithm 7.5.

Part 1. The optimal trajectory consists of at most four parts. The last part, from t_{full} until t_f, is determined in the same way as shown in the proof of Lemma 7.1.

The first three parts of the trajectory are split in the following way: decelerating (until t_{cruise}), cruising at a fixed speed (until t_{acc}), and accelerating (until t_{full}), where the first and last period may have zero length. We want to minimize the area under the absolute value of the acceleration-time graph. We decelerate as early as possible and accelerate as late as possible, and both at the maximum rate. If we would not do one of these three things, it means that we would have to decelerate more as we drive at a high speed longer (and as e.g. the average speed is fixed, namely x_0/t_f). So, indeed the first three parts of a trajectory consist of decelerating at maximum rate, then cruising at a fixed (and relatively low) speed and then accelerating at maximum rate.

Part 2. As argued in the proof of Lemma 7.1, the time t_{full} is determined by the trajectory y of the predecessor of V and is fixed. So t_{full} is chosen as in lines 2-6.

Algorithm 7.5 closed-form alternative to Algorithm 7.3.

Input: x_0, v_0, t_f, $t_{f,y}$, and y.
if $t_f - t_{f,y} = B$ **then**
    Consider trajectory y and determine the time at which the vehicle continues at full speed. Call this time t_{full}.
else
    Put $t_{full} \leftarrow t_f$.
end if
Put

$$t_1 \leftarrow \frac{a_m t_f + v_0 - v_m}{2a_m} - \frac{\sqrt{4a_m|x_0| + (a_m t_f - v_0)^2 - 2(a_m t_f v_m + v_0^2) - 4a_m(t_f - t_{full})v_m + 2v_0 v_m - v_m^2}}{2a_m} \tag{7.5}$$

Put

$$t_2 \leftarrow \frac{a_m t_f + v_0 - v_m}{2a_m} + \frac{\sqrt{4a_m|x_0| + (a_m t_f - v_0)^2 - 2(a_m t_f v_m + v_0^2) - 4a_m(t_f - t_{full})v_m + 2v_0 v_m - v_m^2}}{2a_m} \tag{7.6}$$

Put $t_{cruise} = t_1$ and $t_{acc} = t_2$.
Then,

$$a(t) = x''(t) \leftarrow \begin{cases} -a_m & \text{if } 0 \le t < t_{cruise}, \\ 0 & \text{if } t_{cruise} \le t < t_{acc}, \\ a_m & \text{if } t_{acc} \le t < t_{full}, \\ 0 & \text{if } t_{full} \le t < t_f. \end{cases} \tag{7.7}$$

Knowing $a(t)$, we compute $x(t)$ using the conditions $x(0) = x_0$ and $v(0) = v_0$.
Output: $x(t)$.

Knowing this, we can compute the remainder of the trajectory. We can compute the traversed distance if we immediately decelerate for a time t and accelerate as late as possible for a time $t + v_m/a_m - v_0/a_m$ (because it might be that $v_0 \neq v_m$), which is

$$\begin{aligned} &v_0 t - \frac{1}{2} a_m t^2 + \left(v_m - a_m \left(t + \frac{v_m}{a_m} - \frac{v_0}{a_m}\right)\right)\left(t + \frac{v_m}{a_m} - \frac{v_0}{a_m}\right) + (t_f - t_{full})\, v_m + \\ &\left(v_m - a_m \left(t + \frac{v_m}{a_m} - \frac{v_0}{a_m}\right)\right)\left(t_f - 2t - \frac{v_m}{a_m} + \frac{v_0}{a_m}\right) + \frac{1}{2} a_m \left(t + \frac{v_m}{a_m} - \frac{v_0}{a_m}\right)^2. \end{aligned} \tag{7.8}$$

Equating Equation (7.8) with $|x_0|$ and solving for t, results in two positive values. The smaller one is given as t_1 in Equation (7.5) and the larger one as t_2 in Equation (7.6). So we can put $t_{cruise} = t_1$ and $t_{acc} = t_2$.

Then, when we combine the defined times, we obtain Equation (7.7). With this choice of times, we see that we minimize the area under the absolute value of the acceleration-time graph. This is exactly the same criterion as we optimize for in Algorithm 7.3, so the two algorithms yield the same trajectory. □

Remark 7.3 *Algorithms 7.3 and 7.5 are solvable, if the PFA is regular, the control region is sufficiently big, and the cars are sufficiently far apart from each other when entering the control region (as mentioned before). The regularity of the PFA ensures that the vehicles keep driving behind each other (and, e.g., do not have to overtake). Our closed-form expressions in Algorithm 7.5 provide immediate quantitative insight in the conditions required for solvability. In this case, lines* 2 *to* 6 *are sufficient to determine the influence of the predecessor of the vehicle that we are currently planning. The sufficiently big control region ensures that proper t_{full}, t_1, and t_2 can be found, in such a way that vehicles do not collide, which is also the case for the requirement on the distance between cars when they enter the control region. A full proof would be similar to the proof of Lemma (IV.4) in [134] and would follow along the same lines.*

7.5 Performance analysis

Having covered the two main ingredients of the model, we turn to the performance analysis. The two measures that we consider are the mean delay and the fairness. In order to obtain results on mean delay and fairness, we first establish a link between the model we described so far, and polling models.

7.5.1 Polling model

We have a slightly different polling model than the one that is usually considered in the literature. Therefore, we first describe the type of polling model that we consider.

We face a polling model with n queues, each with a distinct Poisson arrival process with parameter μ_i, which are assumed to be independent from each other. Each queue has its own generally distributed service time from which is sampled independently. A single server is visiting each of the n queues in a certain (possibly random) order to serve customers. After a certain period at a queue, determined by the service discipline, the server switches to the next queue. We assume that if we switch, a setup time is incurred. This setup time is nonzero if the queue to which we switch is not empty. However, if the queue to which we switch is empty, we assume the setup time to be zero. In such a case, we continue immediately to the next queue where, again, a setup time is incurred (see e.g. [179] where a similar setup policy is used). We assume, for simplicity and the ease of exposition, that setup times only depend on the queue to which the server switches. Moreover, if all queues were empty before the arrival of a vehicle, we assume that a setup was started at the most recent departure epoch. This polling with residual setups has not been studied before in the polling literature as far as we are aware, but naturally represents the behavior of our PFAs.

We will analyze the performance of our PFAs regarding the mean delay through the polling models as described above. Although we take a vertical queueing approach in those polling models (i.e. the vehicles are all stopped at the stop line at the intersection, occupying no space, see e.g. [113, Section 3.2]), the SPA provides a one-to-one relation between the vertical queueing model and the PFAs. We visualize this in Figure 7.4, where the black line represents a self-driving vehicle, and the red dotted line represents the corresponding "vehicle" in the vertical queueing model. Both "vehicles" enter the control region at the same time (so also the earliest possible arrival time at the intersection is the same for both). They also have the same service time, because as soon as the vehicles start to cross the intersection they have the same trajectory. So the delay for both vehicles is the same as visualized in Figure 7.4. Alternatively, if an observer would be able to observe vehicles only when they enter the control region at a 100 meters distance from the intersection and at the moment that they cross the intersection, the observer would not be able to distinguish between a vehicle following the red dotted trajectory and the solid black trajectory in Figure 7.4.

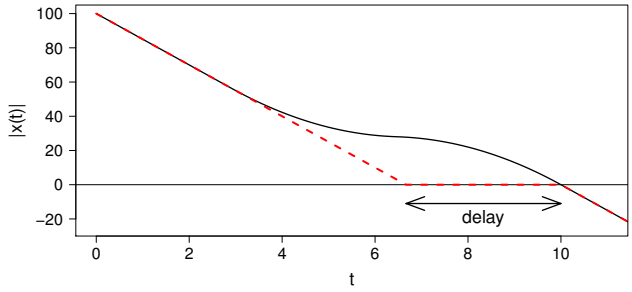


Figure 7.4: Visualization of the link between the traffic model with PFAs and polling models. The black line represents a self-driving vehicle and the red dotted line represents the corresponding "vehicle" in the vertical queueing model.

To make the connection between the traffic model and polling models more explicit, we argue how the traffic model translates to a polling model. The time B in between vehicles from the same stream accessing the intersection is the service time in the polling model, whereas the clearance time S is the setup time in the polling model. Which queue or lane is to be served is decided upon by the service discipline and the PFA respectively.

So, our intersection model precisely fits the framework of polling models. We will use the ideas and results already obtained for polling models to obtain a performance analysis of the traffic model discussed so far. From now on in this section, we will be focusing on the polling model and related results, and therefore use queueing terminology.

7.5.2 Mean delay

The specific assumptions that we made, result in a polling model that does not fall into the standard framework and a fully analytical solution is difficult (if not impossible) to derive. So, we aim to develop accurate approximations for the exhaustive and gated PFA, which are much easier to compute and which are still quite accurate, and refrain from providing an analytical solution. We focus on obtaining approximations for the mean delay that still require some analytical results, but that are easier to derive than the exact value of the mean delay.

We start with a definition of delay. The delay D_i at lane i is defined as the actual time of a car crossing the intersection minus the free-flow time in which

a car could cross the intersection (which is the delay in both the polling model and the intersection model). Further, we denote with B_i the service time at queue i, whereas S_i denotes the setup time when we switch to queue i. We have Poisson arrivals with rate μ_i and define $\rho_i = \mu_i \mathbb{E}[B_i]$ and $\rho = \sum_i \rho_i$, where ρ is similar to the vehicle-to-capacity ratio. The approximation that we propose for the mean delay is of the form,

$$\mathbb{E}[D^P_{i,app}] = \frac{K^P_{0,i} + K^P_{1,i}\rho + K^P_{2,i}\rho^2}{1-\rho}, \tag{7.9}$$

like in [22], where $K^P_{j,i}$ are constants that are yet to be determined and P denotes the PFA. The constants, that might depend on P and the arrival distribution (we only consider Poisson arrivals), are derived through requiring Equation (7.9) to be exact in various limiting cases. These three cases are the following: Equation (7.9) should match the mean delay for queue i in the light-traffic limit, the derivative of the light-traffic limit, and the heavy-traffic limit. Then we have a system of three equations with three unknowns, which we can solve to find the constants $K^P_{j,i}$. These approximations are based on the framework described in [22], which is in turn based on ideas developed in [166]. Note that Equation (7.9) is only valid for $\rho < 1$, which is the condition for the polling model (and therefore also for our PFAs) to be stable.

We start with deriving the light-traffic limit for the mean delay for general service time and setup time distributions. The light-traffic here corresponds to the case where

$$\mathbb{P}(\text{server not working and not setting up}) \uparrow 1,$$

which means that both $\mu_i \mathbb{E}[B_i]$ and $\mu_j \mathbb{E}[S_i]$, $i, j = 1, \ldots, n$, should be close to zero. We denote with X^{res} the residual or overshoot of the random variable X with mean $\mathbb{E}[X^{res}] = \mathbb{E}[X^2]/(2\mathbb{E}[X])$. Then we have the following lemma where, as mentioned before, we restrict the setup times to depend only on the queue to which we switch.

Lemma 7.3 *The light-traffic limit for the mean delay, up to and including first-order terms, for* all *discussed PFAs, is*

$$\mathbb{E}[D^{LT}_i] = \rho_i \mathbb{E}[B^{res}_i] + \sum_{j \neq i} \rho_j (\mathbb{E}[B^{res}_j] + \mathbb{E}[S_i]) + \sum_{j \neq i} \mu_j \mathbb{E}[S_i]\mathbb{E}[S^{res}_i]. \tag{7.10}$$

Proof. We first note that cases where we see more than one customer when we arrive in the system are all of order $O(\rho^2)$ or higher, so we do not consider those

terms. We continue with considering what happens in each phase of the cycle and argue what the delay is of a customer arriving at queue i.

We have n different visit periods, numbered $j = 1, \ldots, n$. If $j = i$, we only have to wait for a residual service time of the customer that is currently in service (using the PASTA property of Poisson arrivals). This happens with probability $\mu_i \mathbb{E}[B_i] = \rho_i$. The contribution to the waiting time is thus $\rho_i \mathbb{E}[B_i^{res}]$. If $i \neq j$, we have to wait for the residual service time of the customer that is in service and for the setup time to our own queue i. This all happens with probability $\mu_j \mathbb{E}[B_j] = \rho_j$, so the contribution to the waiting time is $\rho_j(\mathbb{E}[B_j^{res}] + \mathbb{E}[S_i])$.

If we are currently in a setup period, we might be at queue $j = 1, \ldots, n$. The case $i = j$ does not occur, as we do not have a setup time in that case (we take the customer immediately into service). The cases $i \neq j$ occur with rate $\mu_j \mathbb{E}[S_i]$ (which converges to zero) and if we arrive during such a period, we have to wait for a residual setup time. So the contribution is $\mu_j \mathbb{E}[S_i]\mathbb{E}[S_i^{res}]$.

Summing all separate parts discussed above, we obtain Equation (7.10). Moreover, the given arguments all hold for both the gated and exhaustive PFA, finishing the proof. □

Comparison light-traffic limit of polling models with (residual) setup and switchover times

In [22], Equation (3.11), the light-traffic limit for a regular polling model with switchover times (and no setup times) is given. As mentioned before, setup times are only incurred if the queue to which we switch is non-empty and are equal to zero otherwise (after which we perform another setup when switching to the next queue). Comparing this with our light-traffic limit as in Equation (7.10), we do not have a constant term. This is the result of the behavior of our queueing model when all queues are empty: in such a case only a residual setup is performed instead of a full setup time. I.e. in light traffic and assuming residual setups, the light-traffic limit for the mean delay for each queue is 0, which makes sense for our traffic model. Rewriting (7.10) to

$$\mathbb{E}[D_i^{LT}] = \sum_i \rho_i \mathbb{E}[B_i^{res}] + (\rho - \rho_i)\mathbb{E}[S_i] + \frac{\mathbb{E}[S_i^2]}{2} \sum_{j \neq i} \mu_j,$$

reveals that some of the terms in Equation (3.11) from [22] simply cancel, because the delays during a switchover time (which corresponds to a setup in this chapter) are not of $O(1)$, but of order $O(\rho)$.

In heavy traffic, the behavior of our PFAs and regular polling models is the same, as a setup will always be performed and can be seen as a regular switchover. Consequently, the heavy-traffic limits for the exhaustive and gated PFAs are the same as the heavy-traffic limits for the exhaustive and gated disciplines in e.g. [16], where polling models with switchover times are presented. Indeed, if the lengths of the setups and switchovers are the same, the polling model with switchovers (and without setup times) is the same as the polling model with setup times (but no switchover times), because each setup will be performed in heavy traffic (as all queues tend to be non-empty when the server visits them) and a setup time can be seen as an "ordinary" switchover time. This implies that we can use the results from [16], so

$$\mathbb{E}[D_i^{HT,P}] = \frac{\omega_i^P}{1-\rho} + o((1-\rho)^{-1}), \tag{7.11}$$

with P denoting the PFA, so $P = exh$ (for the exhaustive PFA) or $P = gat$ (for the gated PFA), where, for $i = 1,2,\ldots,n$,

$$\omega_i^{exh} = \frac{1-\hat{\rho}_i}{2}\left(\frac{\sigma^2}{\sum_{j=1}^n \hat{\rho}_j(1-\hat{\rho}_j)} + \sum_{j=1}^n \mathbb{E}[S_j]\right), \tag{7.12}$$

with, in case of Poisson arrivals,

$$\sigma^2 = \frac{\sum_{j=1}^n \mu_j \mathbb{E}[B_j^2]}{\sum_{j=1}^n \mu_j \mathbb{E}[B_j]}$$

and $\hat{\rho}_j = \rho_j/\rho$. For the gated PFA we have

$$\omega_i^{gat} = \frac{1+\hat{\rho}_i}{2}\left(\frac{\sigma^2}{\sum_{j=1}^n \hat{\rho}_j(1+\hat{\rho}_j)} + \sum_{j=1}^n \mathbb{E}[S_j]\right). \tag{7.13}$$

The general approximation in Equation (7.9) is now ready to be used. We obtain the following theorem.

Theorem 7.4 *The mean delay experienced for PFA P can be approximated with Equation* (7.9)*, where*

$$\begin{aligned} K_{0,i}^P &= 0, \\ K_{1,i}^P &= \hat{\rho}_i \mathbb{E}[B_i^{res}] + \sum_{j\neq i} \hat{\rho}_j (\mathbb{E}[B_j^{res}] + \mathbb{E}[S_i]) + \sum_{j\neq i} \hat{\mu}_j \mathbb{E}[S_i^{res}]\mathbb{E}[S_i], \end{aligned} \tag{7.14}$$

$$K_{2,i}^P = \omega_i^P - K_{1,i}^P,$$

with $\hat{\mu}_j = \hat{\rho}_j / \mathbb{E}[B_j]$.

Proof. As mentioned before, we put three conditions on the constants $K_{j,i}^P$, $j = 0, 1, 2$. These are the following

$$\begin{aligned}
\mathbb{E}[D_{i,app}^P]\Big|_{\rho=0} &= \mathbb{E}[D_i^{LT}]\Big|_{\rho=0}, \\
\frac{\mathrm{d}}{\mathrm{d}\rho}\mathbb{E}[D_{i,app}^P]\Big|_{\rho=0} &= \frac{\mathrm{d}}{\mathrm{d}\rho}\mathbb{E}[D_i^{LT}]\Big|_{\rho=0}, \\
(1-\rho)\mathbb{E}[D_{i,app}^P]\Big|_{\rho\uparrow 1} &= \mathbb{E}[D_i^{HT,P}].
\end{aligned}$$

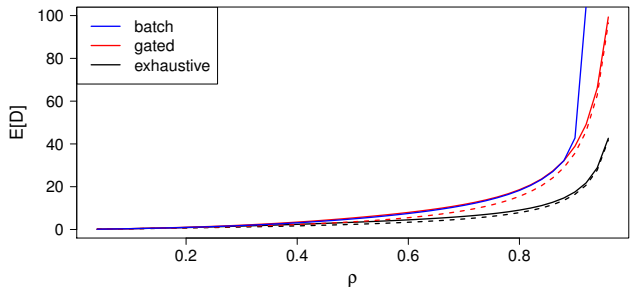
Using Lemma 7.3 and Equation (7.11), we get

$$\begin{aligned}
&K_{0,i}^P = 0, \\
&K_{0,i}^P + K_{1,i}^P = \hat{\rho}_i \mathbb{E}[B_i^{res}] + \sum_{j \neq i} \hat{\rho}_j (\mathbb{E}[B_j^{res}] + \mathbb{E}[S_i]) + \sum_{j \neq i} \hat{\mu}_j \mathbb{E}[S_i^{res}]\mathbb{E}[S_i], \\
&K_{0,i}^P + K_{1,i}^P + K_{2,i}^P = \mathbb{E}[D_i^{HT,P}] = \omega_i^P.
\end{aligned} \tag{7.15}$$

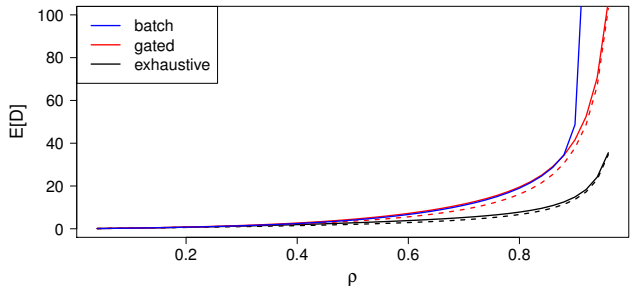
It can easily be seen that Equation (7.15) reduces to Equation (7.14). □

Remark 7.4 *The above mentioned results for the mean delay can readily be extended to results for the mean number of vehicles in the queue using Little's law. Together with the speed regulation algorithm, the physical length of the queue can be calculated (for example if we define the last vehicle that has already decelerated to be in the queue). This would give information about e.g. spillback of the intersection to other intersections.*

In general, the approximations work fine for all discussed PFAs, as can be seen in Figure 7.5, comparing the solid lines (the exact results) and the dashed lines (the approximations). We present examples where we put $v_m = 15$ m/sec, $a_m = 4$ m/sec^2, and $l = 5$ m and where two lanes cross each other. We consider two cases where the load on both lanes is split differently: one case where $\rho_1 = \rho_2$ (referred to as being symmetric) and one case where $\rho_1 = 3\rho_2$ (referred to as being asymmetric). Following [184], we put $B_i = B = 1$ second and $S_i = S = 2.375$ seconds for $i = 1, 2, \ldots, n$. The two discussed PFAs result in Figure 7.5, where also, as a benchmark, the Batch Algorithm from [184] is considered, with



(a)



(b)

Figure 7.5: Mean delay experienced by an arbitrary car for the symmetric case (a) and asymmetric case (b). The solid lines represent simulation results and the dashed lines approximations.

a maximum batch size of 100. The approximations are also good for all other settings we simulated.

We see that the exhaustive PFA performs really well if we focus on mean delay and make a comparison with the other PFAs. The difference between the gated and the exhaustive PFA can also be understood from the heavy-traffic limits in Equations (7.12) and (7.13). The performance of the Batch Algorithm is similar to that of the gated PFA, except for higher values of ρ, which is due to the maximum batch size of 100. This maximum batch size causes a lower maximum capacity for the Batch Algorithm than for the exhaustive and gated PFA and therefore, the Batch Algorithm has a sharp increase in the mean delay at a lower value of ρ than the other two PFAs. We expect the exhaustive PFA to

be close to the optimum with respect to the mean delay. This optimality was, to some extent, already observed in e.g. [124, 142, 222].

7.5.3 Fairness

In order to show that the exhaustive PFA is not the best for all performance metrics we consider fairness in this subsection. We use the definition of fairness for polling models, denoted with F, as introduced in [176],

$$F = \frac{\mathbb{E}[N_{ahead}]}{\mathbb{E}[N_{total}]},$$

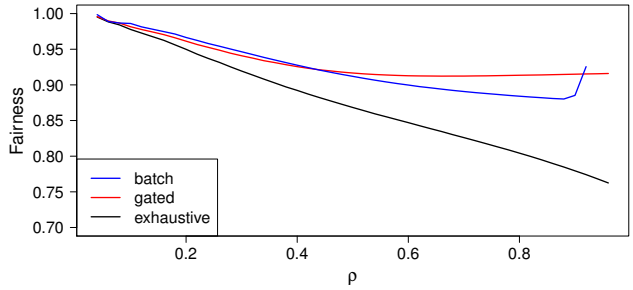
where N_{ahead} denotes the number of cars an arbitrary car sees upon arrival and that are served ahead of it; and where N_{total} denotes the total number of cars across the entire intersection an arbitrary car sees upon arrival. A fairness close to 1 is considered fair (as there are few overtakes) and a fairness close to 0 as unfair (as there are many overtakes). In words this means that we quantify the percentage of cars that did not overtake an arbitrary car (on an intersection-wide basis).

In Figure 7.6 we present simulation results for fairness for the same set of examples as for the mean delay. Considering fairness, we see once more that the gated PFA is close to the Batch Algorithm for values of ρ that are not too high. The increase of fairness for high values of ρ for the Batch Algorithm is due to the maximum batch size of 100. The exhaustive PFA performs worse on fairness, but is still above 75%. It seems that a low mean delay results in a relatively low fairness, showing a potential need to balance the two performance measures, which is also (to some extent) visible in the increase of fairness for the Batch Algorithm for high values of ρ.

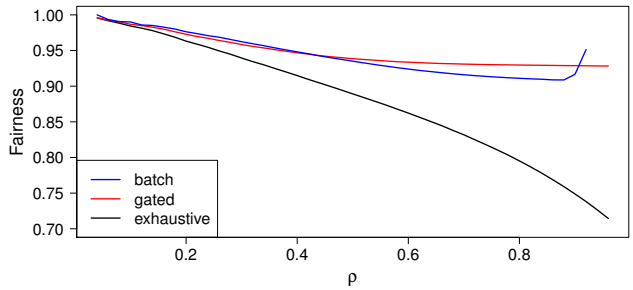
7.6 Comparison traditional traffic light and PFAs

The goal of this section is to provide a comparison between traditional traffic lights and PFAs on the basis of mean delay. As a measure for the traditional traffic light we use the traffic simulator SUMO. We will consider two scenarios in SUMO: one with fixed control and one with adaptive control (based on the so-called time loss in the SUMO User Documentation). We will compare these two scenarios with the exhaustive PFA.

We consider two examples where two lanes cross each other. In the first example, the vehicle-to-capacity ratio is the same on both lanes, whereas in the



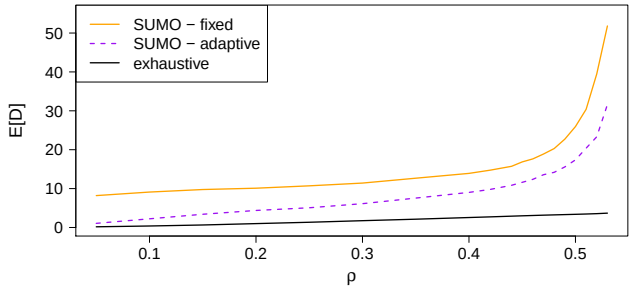
(a)



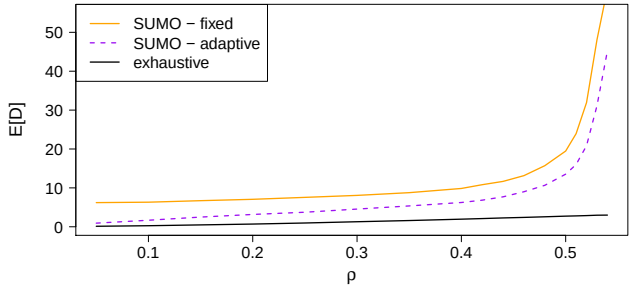
(b)

Figure 7.6: Fairness experienced by an arbitrary car for the symmetric case (a) and asymmetric case (b).

second example the ratio between the loads on the lanes is $1:3$. For the exhaustive PFA we again put $B_i = B = 1$ second and $S_i = S = 2.375$ seconds. For the fixed control simulation in SUMO and the first example we assume a green period for both lanes of 22 seconds and an amber period of 3 seconds; for the second example we pick green periods of 11 and 33 seconds and an amber period of 3 seconds. Note that some of the results for the fixed control in Figure 7.7 could be improved by adapting the length of the green period. For the adaptive control in SUMO we assume a maximum green period duration of 45 seconds and an amber period of 3 seconds for the symmetric example. For the asymmetric example we choose a maximum green period of 22 and 68 seconds and an amber period of 3 seconds. Note that we do not have to define the variable



(a)



(b)

Figure 7.7: Mean delay for an arbitrary car for traditional traffic lights (represented by SUMO) and the exhaustive PFA for the symmetric case (a) and the asymmetric case (b).

B_i in SUMO, as the vehicles themselves decide what B_i is, implying that B_i is random (and usually higher than in the PFA setting). The delay in SUMO for the fixed and adaptive control is obtained in the following way: we compute the mean time spent in the system for all vehicles and subtract the mean time vehicles spend in the system under free-flow conditions (by giving a green light for a lane all the time). We take exactly the same arrivals for all three control strategies.

In Figure 7.7 we see that there is quite a difference between the traffic light with fixed settings and the adaptive traffic light when we compare them with the exhaustive PFA. To some extent, this was also observed in [184]. The capacity of the intersection for the latter case is almost twice as high as for the traditional

traffic light, showing a huge potential in resolving congestion. This is mainly due to the speed regulation of vehicles, which increases the speed of vehicles crossing the intersection, resulting in relatively low B_i. Partly, the reduction is also due to the alternative scheduling strategy in the exhaustive PFA.

7.7 Conclusion

We have shown that significant gains can be obtained compared to nowadays traffic when speed regulation and PFAs are employed and have given ways to decrease the mean delay at intersections. This has been shown through a connection between polling models and PFAs.

It seems that the exhaustive PFA is close to the optimum when minimizing the mean delay is key. However, the exhaustive PFA exhibits relatively poor fairness characteristics. It might be worthwhile to find a balance between mean delay and (e.g.) fairness in order to obtain some kind of optimal setting for the PFA. A possibility hereto might be the k-limited discipline (as discussed in e.g. Chapters 4 and 6) and which can be seen as an alternative to the exhaustive and gated PFA that we considered in this chapter.

In principle our PFAs could be used in nowadays traffic as well. The only requirement is that it must be known on an intersection wide basis in which order the vehicles arrive. The requirement that we can control the speed of arriving vehicles is not needed to execute the PFAs. This assumption only plays a role in what the variables B_i and S_i are. Regardless of the distributions for B_i and S_i, the scheduling part of a PFA might still be used. Using some kind of speed advisory system for conventional vehicles, it might be possible to come quite close to the performance of the PFAs based on self-driving vehicles.

We advocate to investigate more realistic intersection scenarios than the two-lane scenarios considered for PFAs in Chapter 7, yet we expect similar results in examples with more than two lanes if vehicles from at most one lane are crossing the intersection. Another extension would be to allow for turning traffic (introducing different service times for vehicles on the same lane when a lane has both turning and non-turning vehicles). Accounting for e.g. pedestrians seems to be possible as well, e.g. by introducing some specific periods in each cycle during which no vehicles are allowed to cross the intersection if there are pedestrians that want to cross.

We have derived approximations for some of the relevant performance measures in this chapter. We were not able to provide an exact analysis. It would be worthwhile to make an effort to provide such an analysis for the special case

of branching-type disciplines. As long as such an exact analysis is not available, it is worthwhile to study further approximations of for example the delay distribution. It is conceivable that results for higher moments and the variance of the delay can be derived, e.g. using techniques similar to those in [71].

The framework developed in this chapter can be further extended, e.g. to model a situation with mixed traffic, meaning that there are *both* autonomous and non-autonomous vehicles. Such an extension is important to study, because there will be a period during which such a mixture of vehicles is present on the roads. As such, a framework for mixed traffic needs to be developed. Further extensions that might be considered are different ways of forming platoons and alternative ways for vehicles to approach the intersection. Especially in the case of mixed traffic, the latter is important to consider as autonomous and non-autonomous vehicles will have significantly different driving behavior.

A further investigation on the practical implementation of our PFAs might be of interest too. We made several simplifying assumptions that need to be verified. For example, we assume that all messages that need to be exchanged are received by all relevant entities and that there is no communication delay. Also a notion like string stability of a platoon of vehicles, i.e. whether oscillations in e.g. the speed of individual vehicles in a platoon of vehicles amplify or not across the different vehicles (for more information see e.g. [183]), might be investigated for our proposed models.

Chapter 8

Automated detection of unexpectedly high traffic flow in uncongested traffic states

8.1 Introduction

As we have argued before, traffic jams have become an inevitable part of road traffic. We have been focusing so far on intersections and finding e.g. good or even optimal traffic-light settings, while in this chapter we shift our focus to traffic congestion on motorways.

Reducing traffic congestion, also on motorways, is a challenging problem, be it only because traffic has a highly complex nature. One could aim to influence the amount of driving or the drivers' behavior on motorways. This can be achieved by, for example, monetary means (such as toll systems or congestion pricing, see e.g. [12, 84]), encouraging drivers to drive outside peak hours (see [72] for instance), or dynamic road signaling (see e.g. [91]). It is increasingly important to find the exact effect of these measures, but this is a complicated problem, which is partly due to the fact that the manifestation of congestion on motorways is subject to randomness, see for example [10, 194].

In this chapter, we approach the problem of reducing traffic congestion on motorways from a different perspective than the aforementioned papers, as we look at the *absence* of traffic jams. Typically, once the traffic flow, i.e.

the throughput measured in vehicles per hour, has passed a certain threshold, congestion *could* emerge. This phenomenon is referred to as a "breakdown". We are interested in days during which relatively many breakdowns were expected, but did not occur. Such days will be referred to as "high-performance days". Specifically, we develop an algorithm to automatically identify these high-performance days based on historical traffic data and test our method on a section of the A15 motorway in the Netherlands. In a future study, one could try to determine the specific characteristics of the resulting high-performance days using more detailed data. Ultimately, the goal is to find out whether the high-performance days could be caused by specific behavioral patterns of individual drivers. However, we focus on the first step, namely the automated detection of high-performance days.

Our algorithm relies on the shape of the macroscopic fundamental diagram, the well-known empirical diagram that displays the relationship between the traffic flow q (vehicles per hour) and the traffic density ρ (vehicles per kilometer) at a specific location. Many studies have shown that the fundamental diagram can be divided into two regions, a region for congestion and a region for free flow. The empirical fundamental diagram has been studied extensively and a wide variety of theoretical models has been proposed (see for example [80] for an overview). A further introduction to the fundamental diagram is given in Subsection 8.3.1. However, our aim is not a theoretical model for the fundamental diagram: we are merely interested in the critical speed, i.e. the speed which defines congestion and separates the free-flow region from the congestion region in the fundamental diagram. So, we can get around the problem of modeling the congestion region and exploit the roughly linear flow-density relationship during free flow. We show that *robust regression* can be used to obtain the free-flow speed and subsequently distinguish between free flow and congestion based on the calculated weights. Utilizing the method proposed by [10], we subsequently estimate the *breakdown probability*. This paves the way to identifying high-performance days, i.e. days with a (relatively) high flow/breakdown probability while a traffic jam remains absent. Our algorithm is thus *not* designed to explain why a traffic jam occurs. Investigating why a traffic jam occurred, is beyond the scope of this chapter. There may be many (combinations of) causes: a traffic jams might, e.g., be induced, see e.g. [34], or the reason may be found in microscopic traffic data such as the influence of downstream on-ramps or lane-changing behavior, see e.g. [58].

To the best of our knowledge, our approach to obtain the critical speed and the introduction of the notion of high-performance days are original. Many papers focus on (real-time) traffic jam estimation using GPS-data and/or trajectory

data, see e.g. [149, 162, 213]. This is partly due to the widespread availability of GPS data. However, we have chosen to use detector data, as traffic detectors are present on most Dutch motorways and provide a sufficiently high granularity. Detector data is also used in the literature; in [125] detector data is used to automatically track congestion and in [106] detector data is used to study phase transitions on German motorways. However, the work that is probably closest to our study is [67]. Therein, the authors use detector data to estimate motorway characteristics such as the free-flow speed and the critical density. These quantities are then used to calibrate a cell transmission model. We determine a related motorway characteristic (the critical speed), but in our study this is a tool to estimate the breakdown probability. Indeed, our main goal is different: we identify a surprising *absence* of traffic jams. This could be an important first step towards a better understanding of the reasons why on certain days the traffic flow is so much better than on other days, although the circumstances seem to be identical.

Our main contributions can be summarized as follows:

(i) We present a novel algorithm to automatically detect points in time which have both a high traffic flow and a high speed based on historic loop detector data. Ultimately, this leads to the identification of high-performance days.

(ii) We apply our algorithm to investigate a case study on a part of the A15 motorway in the Netherlands and we are able to identify high-performance days and several interesting patterns.

Chapter outline

The remainder of this chapter is organized as follows. In Section 8.2 we provide information about the location of the experimental region and discuss the data. We proceed with the theoretical foundation and the three main steps of the algorithm in Section 8.3. The validation of important assumptions and parameter choices is presented in Section 8.4, as well as the main insights of the case study. We close with a conclusion in Section 8.5.

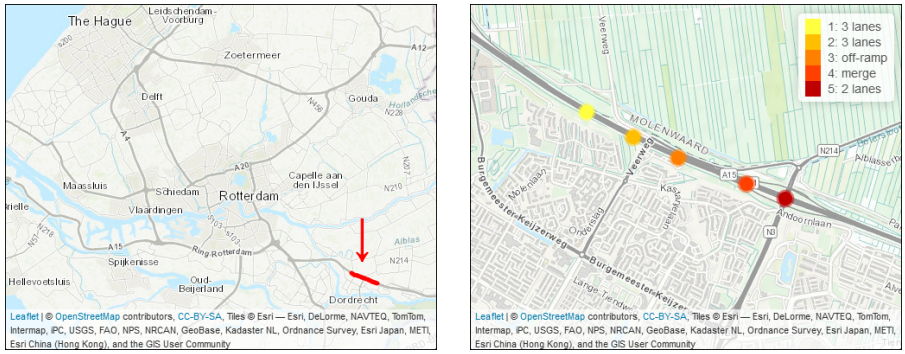
8.2 Description of the location and the data

In this section, we discuss the relevant aspects of the part of the A15 motorway from which the data is obtained. Subsequently, we elaborate on the structure of

the data set and which steps we take in the preprocessing of the data.

8.2.1 Location of the experimental region

The location under consideration is the A15 motorway near Rotterdam, at the N3 interchange with Papendrecht (see Figure 8.1). Five detectors have been placed in the eastern direction, with a distance of approximately 300 meters between consecutive detectors (see Figure 8.1(b)). Between the second and third detector, an off-ramp to Papendrecht is located. Shortly afterwards, the vehicles on the A15 merge from three to two lanes. The maximum speed along this whole trajectory of the A15 is 120 km/h (at the time of this study). The traffic jams on this trajectory belong to the most costly traffic jams in the Netherlands (see [33]) and the A15 is one of the most congested roads in the Netherlands, connecting one of the world's largest ports with the European main land, which makes this a particularly important and interesting motorway to study.



(a) (b)

Figure 8.1: (a) Overview of the trajectory, marked red and indicated by the red arrow, in relation to Rotterdam. (b) The location of the five detectors on the trajectory.

8.2.2 Description of the data set

The data is obtained from the Dutch National Data Warehouse for Traffic Information (NDW), a collaboration of 19 public authorities that cooperate on collecting, storing, and redistributing data. The data is publicly available and can be requested at the website of the NDW [139]. The data we obtained from

the NDW spans a period from January 1, 2018 until December 31, 2018. Every minute, the detectors measure, for each lane individually, the number of vehicles that have passed (i.e. the traffic flow q, in vehicles per hour) and the average speed v of the passing cars in kilometers per hour, calculated using the arithmetic mean. We can estimate the average traffic density ρ using $\rho = q/v$, although this formula is known to underestimate the density when the arithmetic mean is used to obtain the mean speed [114]. We combine the various lanes as in [193, Chapter 3]. For the sake of reducing the variability in the data, we aggregate the measurements to a period of 5 minutes, as is done in [10]. The arithmetic mean is used to obtain the average traffic flow and the average speed is calculated analogously to the average speed over multiple lanes.

The resulting data set can be described as follows. We introduce the set of locations $\mathscr{I} := \{1,2,3,4,5\}$, in accordance with Figure 8.1(b). Moreover, we focused our research on weekdays and thereby excluded all weekend days from the data, because the traffic flow is oftentimes significantly lower in the weekend. The set containing all 261 weekdays in 2018 (including e.g. holidays) is denoted by $\mathscr{J}$. After the aforementioned exclusions, we have one set of measurement dates $\mathscr{J}^{(i)} \subseteq \mathscr{J}$ for each detector $i \in \mathscr{I}$. At each location we have measurements of the average traffic flow and average vehicle speed, as well as an estimate for the density, aggregated to 5-minute intervals. Hence, for location $i \in \mathscr{I}$ and date $j \in \mathscr{J}^{(i)}$ we have a sequence of measurement times

$$\mathscr{T}^{(i,j)} := \left\{ t_1^{(i,j)}, t_2^{(i,j)}, \dots \right\} \subseteq \mathscr{T},$$

where $\mathscr{T}$ is the set containing all 5-minute intervals on a day. The corresponding set of measurements for detector i on date j is

$$\mathscr{X}^{(i,j)} := \left\{ \left(q_t^{(i,j)}, v_t^{(i,j)}, \rho_t^{(i,j)} \right) : t \in \mathscr{T}^{(i,j)} \right\}.$$

The data set containing only the flow and the density is denoted by

$$\tilde{\mathscr{X}}^{(i,j)} := \left\{ \left(\rho_t^{(i,j)}, q_t^{(i,j)} \right) : t \in \mathscr{T}^{(i,j)} \right\}.$$

In total we have $|\mathscr{I}| = 5$ locations and $|\mathscr{J}| = 261$ dates, leading to a total of $5 \cdot 261 = 1305$ instances. However, in the first step of the algorithm (i.e. estimating the critical speed), we do not include all days/critical speeds:

(i) We exclude the most extreme critical speeds of each location (see Subsection 8.3.1 for a motivation in relation to our assumptions and Subsection 8.3.2 for a further elaboration);

(ii) We exclude instances where the free-flow and congestion region are not linearly separable by a straight line through the origin, given the labeling (see Remark 8.2);

(iii) We exclude days with little or no congestion (see Subsection 8.4.3).

For the remaining steps, we do include all 1305 instances, meaning that no weekdays are beforehand excluded when identifying the high-performance days.

All the analyses were performed in the statistical software package R.

8.3 The main algorithm

We present the main algorithm in this section and elaborate on the theoretical foundation using traffic theory, robust regression, and the estimator for the breakdown probability proposed in [10]. The algorithm consists of three parts: (i) estimating the critical speed, (ii) estimating the breakdown probability, and (iii) identifying the high-performance days. In Subsection 8.3.1, we formally define the relevant notions, such as the critical speed. In Subsection 8.3.2, we explain how the critical speed is obtained using robust regression as a labeling tool. Lastly, in Subsection 8.3.3, we discuss the estimator for the breakdown probability and provide a definition for high-performance days based on "unperturbed moments".

8.3.1 The fundamental diagram and the critical speed

Studying the traffic behavior at a specific location, say location i, one can distinguish two different traffic states: free flow and congestion. As in [105], we can define free flow and congestion based on the critical speed.

Definition 8.1 (Free flow, Congestion, and Critical speed) Free (traffic) flow *is a state in which the vehicle density in traffic is small enough for interactions between vehicles to become negligible. Therefore, vehicles have an opportunity to move at their desired maximum speeds [105]. When the density increases beyond a certain threshold in free flow, vehicle interaction cannot be neglected anymore. Due to this vehicle interaction, the average vehicle speed decreases to a value lower than the* critical speed*, which is the minimum average speed that is still possible in free flow. This new state of traffic is referred to as a state of* congested traffic*.*

We denote the critical speed at location i by $v_{\text{crit}}^{(i)}$. In the fundamental diagram, this critical speed separates the free-flow region from the congestion

region. The free-flow set of location i on date j, i.e. the set containing all data points corresponding to free flow, is defined as

$$\mathscr{F}^{(i,j)} := \left\{ \left(q_t^{(i,j)}, v_t^{(i,j)}, \rho_t^{(i,j)} \right) \in \mathscr{X}^{(i,j)} : v_t^{(i,j)} \geq v_{\text{crit}}^{(i)} \right\},$$

i.e. the set of all data points of location i and date j for which the average speed is equal to or higher than the critical speed of location i. Naturally, the congestion set is defined as the complement of the free-flow set, i.e.

$$\mathscr{C}^{(i,j)} := \mathscr{X}^{(i,j)} \setminus \mathscr{F}^{(i,j)}.$$

Empirical fundamental diagram of traffic flow

The fundamental diagram of traffic flow is an important tool in traffic engineering. We specifically consider the empirical fundamental diagram where we have traffic measurements, usually from detectors. Depending on the aim of the diagram/available data any two of the following three quantities are displayed: the traffic density (in vehicles/kilometer), the average velocity (in kilometers/hour), and the traffic flow (in vehicles/hour). An example can be found in Figure 8.2, where the traffic flow is displayed horizontally and the traffic density vertically.

The fundamental diagram of traffic flow is often used to describe a macroscopic relation between the traffic flow, traffic density, and speed. It for example sheds light on macroscopic quantities like the capacity and free-flow speed and as such is an important tool in assessing the general quality of the traffic performance. Moreover, it is often used to find and/or predict the (high-level) effect of countermeasures against congestion, such as (temporary) speed limits.

An easy observation that one could make on basis of a fundamental diagram as displayed in Figure 8.2, is that there is a sharp distinction between points that are approximately on a straight line through the origin and points that clearly deviate from that line. The points on the left-hand side of Figure 8.2 indeed are all approximately on a straight line through the origin. Those points correspond to free-flow points, i.e. there is no (significant amount of) congestion. The points scattered on the right-hand side of Figure 8.2, usually belong to cases where there is congestion. This is a particular feature of the fundamental diagram that we exploit in this chapter to estimate the free-flow speed with which we are ultimately able to find high-performance days.

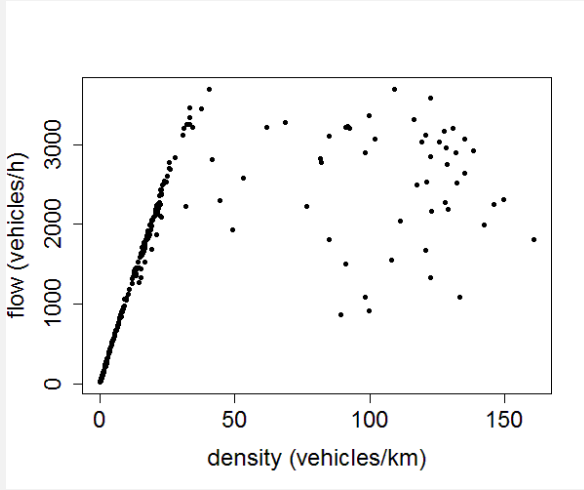


Figure 8.2: Fundamental diagram with on the horizontal axis the traffic density and on the vertical axis the traffic flow.

During free flow, the flow-density relationship can be modeled by a straight line (see the orange line in Figure 8.3(a)), which logically must pass through the origin:

$$q \approx \rho \cdot v_{\text{free}}^{(i)} \quad \forall (q, v, \rho) \in \mathscr{F}^{(i,j)}. \tag{8.1}$$

When using the data set $\mathscr{X}^{(i,j)}$, we assume the following conditions are met:

(i) The average speed during free flow $v_{\text{free}}^{(i)}$ is constant for all locations $i \in \mathscr{I}$;

(ii) The road conditions at location i are homogeneous for all dates $j \in \mathscr{J}^{(i)}$, for all locations $i \in \mathscr{I}$;

(iii) For each $i \in \mathscr{I}$ and $j \in \mathscr{J}^{(i)}$, the number of free-flow measurements significantly exceeds the number of congestion measurements.

Whenever at least one of these conditions is violated, for a certain day j at location i, day j will not be taken into account when determining $v_{\text{crit}}^{(i)}$. The first condition is rarely violated, since a constant free-flow speed follows from the definition of free flow (see e.g. [105]), given conflict free roads with a fixed

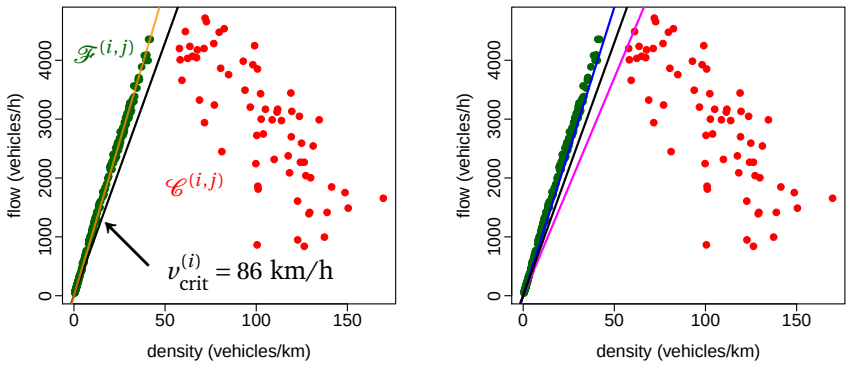


Figure 8.3: The fundamental diagram with free-flow points (green) and congestion points (red). In (a), it is shown how the free-flow region and the congestion region are linearly separable by a straight line through the origin (the black line). The slope of this line is the critical speed. Additionally, the slope of the orange line through the origin is the (constant) free-flow speed, which is 95.5 km/h. Note that the free-flow speed is significantly below the speed limit, as this is an average over both multiple vehicles, vehicle types, and multiple lanes. In (b), it is shown how the critical speed can be estimated by the line that lies exactly between the boundary line of the free-flow region (blue) and the boundary line of the congestion region (magenta).

speed limit and homogeneous conditions. Assumptions (ii) and (iii) may be violated on days where circumstances are completely different from ordinary days, for example in case of accidents, road works, or extreme weather conditions. These days could be detected using additional data and therefore be removed from the data set. However, in order to keep the algorithm as simple and self-contained as possible, we simply choose to exclude the most extreme critical speeds. We emphasize that in our experimental region the core elements of the road were fixed throughout the year, i.e. the speed limit is fixed and no traffic lanes where removed or added. Furthermore, despite the experimental region being subject to heavy congestion, congestion occurs mainly during the morning and afternoon rush hour, which means that in general the number of free-flow measurement well exceeds the number of congestion measurements. As a result, assumptions (i), (ii), and (iii) are only violated in extreme cases and removing the most extreme critical speeds will be sufficient to ensure the assumptions are met. This explains the first point regarding the removal of

several critical speeds stated in Subsection 8.2.2.

8.3.2 Using robust regression to label data points

The purpose of our algorithm is to find the free-flow set and the congestion set, for every day and location. More formally, we aim to find a label for each $(q, v, \rho) \in \mathscr{X}^{(i,j)}$ that indicates whether $(q, v, \rho) \in \mathscr{F}^{(i,j)}$ or $(q, v, \rho) \in \mathscr{C}^{(i,j)}$. A logical first step is to determine the straight line through the origin that lies exactly between the free-flow region and the congestion region, as depicted by the black line in Figure 8.3(b). The slope of this line is the estimate of the critical speed of location i for each date $j \in \mathscr{J}^{(i)}$, denoted by $v_{\text{crit}}^{(i,j)}$.

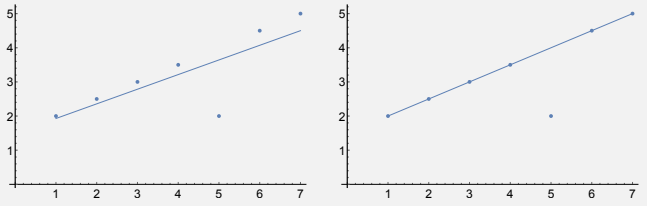
In order to obtain the critical speed and the corresponding labeling from the fundamental diagram, several methods have been studied in the literature. Examples are an iterative regression method after performing a change-point analysis [10], the use of fuzzy logic for clustering [181], and assuming a specific model for the fundamental diagram, obtaining the critical density and subsequently labeling each point [114]. However, we opt for a more intuitive and efficient method based on robust regression, that exploits the underlying structure of the fundamental diagram.

Robust regression

Robust regression essentially is a linear regression that is made (somewhat) robust against violations of the assumptions that are made when fitting a linear regression model. One of those assumptions is that there are no outliers, which is a quite strong assumption. This is one of the reasons why robust regression was developed.

In robust regression, each data point $\mathbf{x}$ is assigned a weight $w(\mathbf{x}) \in [0, 1]$ and subsequently a linear model is fitted and a reiterative weighted least squares fit is performed (where the weights are updated each step according to the new estimate). Employing such weights ensures that outliers have a smaller influence on the final estimates due to their lower weights and the model aims to fit the majority of the data, rather than the whole data set, see for example [138].

A simple example of the difference between standard linear regression and robust linear regression can be found in Figure 8.4. We clearly see that robust regression (Figure 8.4(b)) is more robust against outliers than a standard linear regression (Figure 8.4(a)).



(a) (b)

Figure 8.4: One example of linear regression in (a) and one of robust regression in (b) on the same data set. We can "clearly" see that the data point $(5,2)$ is an outlier. Robust regression decreases the weight of this data point and fits the majority of the data well. This is not the case for linear regression, where the outlier substantially influences the model, causing a mismatch between the majority of the data and the model.

We apply robust regression to the flow-density set $\bar{\mathscr{X}}^{(i,j)}$ of each location i and date j separately. Specifically, we fit the following model:

$$q_t = v_{\text{free}}^{(i,j)} \cdot \rho_t + \varepsilon_t \quad \forall (\rho_t, q_t) \in \bar{\mathscr{X}}^{(i,j)}, \tag{8.2}$$

where the ε_t are error terms with expectation zero. In our case, the "outliers" are the points corresponding to congestion. There are three reasons why this method works so well for this application:

(i) We exploit the fact that in free flow, the relation between q and ρ is linear;

(ii) We do not have to assume any specific relation between q and ρ in the congested set, because these points fulfill the role of outliers;

(iii) The method computes weights that are a measure for the contribution of each point to the final estimate, which can be used for the labeling.

Remark 8.1 *Assumption (iii) from Subsection 8.3.1, specifying that we only consider days where the number of points corresponding to congestion is smaller than the number of free-flow points, is essential. On a day where this assumption is violated, we have more points belonging to congestion, meaning that the fitted regression line would no longer pass through the free-flow set. In this case, the estimated free-flow speed* $v_{free}^{(i,j)}$ *would be significantly lower than the maximum speed, which makes these days extremely easy to detect (and remove).*

The robust regression is performed using the function rlm from the MASS-package in R, with MM-estimation and Tukey's Bisquare function for the weights with the default S-estimator as suggested in [214]. Tukey's Bisquare function behaves similarly to the squared error function except for larger errors, for which it decreases the weight (see e.g. [138]). This results in an estimate for $v_{\text{free}}^{(i,j)}$ and certain weights $w(\mathbf{x})$ for each data point $\mathbf{x} \in \bar{\mathscr{X}}^{(i,j)}$. Instead of the usual interest in the model and parameter estimation, we are interested in the *weights* associated with each data point. Using the weights, we perform the labeling: if the weight is low and if the data point corresponds to a speed lower than the free-flow speed, $v_{\text{free}}^{(i,j)}$, the data point will be labeled as congestion. All other points will be labeled as free flow. Hence, for each $\mathbf{x} = (\rho, q) \in \bar{\mathscr{X}}^{(i,j)}$ we determine $\mathbb{1}_{\mathscr{C}}(\mathbf{x}) := \mathbb{1}\{\mathbf{x} \equiv (q, v, \rho) \in \mathscr{C}^{(i,j)}; \mathbf{x} \in \bar{\mathscr{X}}^{(i,j)}\}$, i.e. the indicator function for the event that $\mathbf{x}$ corresponds to congestion or not. The critical weight has been placed at 0.01 (see Subsection 8.4.3 for a justification), hence

$$\mathbb{1}_{\mathscr{C}}(\mathbf{x}) = \begin{cases} 1 & \text{if } w(\mathbf{x}) < 0.01 \text{ and } v = q/\rho < v_{\text{free}}^{(i,j)}, \\ 0 & \text{otherwise.} \end{cases}$$

After we obtain the labels, we estimate $v_{\text{crit}}^{(i,j)}$ (see the black line in Figure 8.3(b)) by determining the slope of the straight line through the origin that lies exactly between the free-flow region and the congestion region.

Remark 8.2 *It may happen that the boundary line of the congestion region lies above the boundary line of the free-flow region (i.e. the magenta line has a larger slope than the blue line in Figure 8.3(b)), since the weights are calculated based on the Euclidean distance from the free-flow line. In this case, the free-flow region and the congestion region are not linearly separable by a straight line through the origin, given the labeling. For such instances, there will exist data points* $\mathbf{x} \equiv (q, v, \rho) \in \mathscr{C}^{(i,j)}$ *and* $\mathbf{x}' \equiv (q', v', \rho') \in \mathscr{F}^{(i,j)}$ *such that* $v > v'$*. The critical speed for such instances is indeterminate and therefore we do not include these instances in the determination of the critical speed of the corresponding location.*

In the end, the critical speed of location i is estimated as follows:

$$v_{\text{crit}}^{(i)} = \text{median}\{\mathscr{V}_{\text{crit}}^{(i)}\},$$

where

$$\mathscr{V}_{\text{crit}}^{(i)} := \left\{ v_{\text{crit}}^{(i,j)} \right\}$$

such that:

$$\left| v_{\text{crit}}^{(i,j)} - \mu\{v_{\text{crit}}^{(i,j)}\}_{j \in \mathscr{J}^{(i)}} \right| < 2\sigma\{v_{\text{crit}}^{(i,j)}\}_{j \in \mathscr{J}^{(i)}}; \tag{8.3}$$

$$v' > v \quad \forall \mathbf{x} = (q, v, \rho) \in \mathscr{C}^{(i,j)}, \quad \mathbf{x}' = (q', v', \rho') \in \mathscr{F}^{(i,j)}; \tag{8.4}$$

$$\text{MAPE}\left(\bar{\mathscr{X}}^{(i,j)}\right) \geq 0.1. \tag{8.5}$$

Here, $\mu\{\cdot\}$ and $\sigma\{\cdot\}$ denote the mean and standard deviation of the corresponding sets respectively and $\text{MAPE}\left(\bar{\mathscr{X}}^{(i,j)}\right)$ denotes the mean absolute percentage error of the regression model presented in Equation (8.2) (for more information on the MAPE, see the next paragraph).

Equation (8.3) removes the most extreme critical speeds. By excluding days with a critical speed that lies outside a range of twice the standard deviation from the average, we prevent potential violations of the assumptions from influencing the estimates (as elaborated upon in Subsection 8.3.1). Equation (8.4) excludes days where the boundary line of the congestion region lies above the boundary line of the free-flow region (see Remark 8.2). Lastly, Equation (8.5) ensures that the critical speed of a location is not based on days with little or no congestion. As one can imagine, in case of hardly any congestion, a free-flow point with a relatively slow speed might be incorrectly labeled as congestion. We therefore impose a minimal level of congestion and use the mean absolute percentage error (MAPE, see e.g. [182]) of the corresponding model (see Equation (8.2)) as a surrogate of the average congestion level. The MAPE expresses the error of the model in terms of a percentage: a low MAPE corresponds to a very accurate model, implying hardly any congestion, whereas a high MAPE indicates that various points deviate from the straight line through the origin, which corresponds to the presence of congestion during that day. The critical level of the MAPE has been placed at 0.1. In Subsection 8.4.3, this threshold will be motivated.

The set of critical speeds of location i, corresponding to the instances of location i which satisfy the three conditions presented in Equations (8.3), (8.4), and (8.5), is given by $\mathscr{V}_{\text{crit}}^{(i)}$. The critical speed of location i is subsequently determined by taking the *median* of this set. We take the median of the critical speeds among multiple days to provide a solid baseline for comparison among different days. We emphasize that in the end the critical speed of each location is estimated as the median of at least 147 critical speeds (out of 261 weekdays) and that most instances were removed based on Equation (8.4).

8.3.3 Estimating the breakdown probability and identifying the high-performance days

Congestion arises as a consequence of a breakdown, which is defined as a transition from free flow to congestion (see, e.g., [10]). Usually, this happens when the traffic flow is high and some kind of disruption occurs (e.g. a vehicle changing lanes or another sudden movement of a driver).

Definition 8.2 (Breakdown) *A breakdown, at location i and date j, is a moment $t_k^{(i,j)} \in \mathcal{T}^{(i,j)}$ such that*

$$v^{(i,j)}_{t_k^{(i,j)}} \geq v^{(i)}_{crit} > v^{(i,j)}_{t_{k+1}^{(i,j)}}.$$

Remark 8.3 *Please note that in the definition of a breakdown, we do not consider* why *the breakdown occurred. As such, we do not take the mechanisms that cause the breakdown into account. A breakdown might be caused by the high traffic flow at that specific position and time or it might, e.g., be the case that a breakdown is induced by another (downstream) breakdown. Such an induced breakdown is caused by a drop in the outflow and speed of traffic at the current location because it meets the tailback of a downstream traffic jam. Certainly in the context of (stochastic) capacity estimation such a distinction is often taken into account, see e.g. [34, 130, 136, 157]. The distinction is made because an induced breakdown is not informative when the capacity is investigated as the breakdown is not caused by the high traffic flow but by another factor. However, there are also studies that investigate breakdowns without making a distinction between induced breakdowns and breakdowns that are not induced such as [10, 70].*

For our purposes, i.e. detecting whether there is both a high flow and no breakdown, the mechanism that causes a breakdown does not seem to have a major impact. We are mainly interested in whether *there is a breakdown and not in the cause of the breakdown. Indeed, the purpose of our algorithm is merely to identify high-performance days, days with both a high traffic flow and no traffic jam.*

It is, e.g., possible to check whether a breakdown is induced or not. One could probably extend the algorithm to make a distinction between those two types of breakdowns and check whether the algorithm gives a (significantly) different output when e.g. induced breakdowns are excluded. Such a distinction might be created by a clever preprocessing of the data. It might also be possible to check for other underlying mechanisms that cause breakdowns in the obtained data and to take those into account.

We assume that breakdowns have a probabilistic nature, see e.g. [10, 194], meaning that from a macroscopic point of view the occurrence of breakdowns (given a certain traffic flow) is random. This implies the existence of a breakdown probability (as a function of the traffic flow). To estimate this probability, we use the non-parametric estimator discussed in Arnesen and Hjelkrem [10]. To calibrate this estimator, the aforementioned classification of each data point as either free flow or congestion is required. Arnesen and Hjelkrem define two functions: $Q^{(i)}(q)$, which is the number of breakdowns at location i while the traffic flow is equal to or lower than q, and $R^{(i)}(q)$, which is the number of times a breakdown did not occur at location i with a traffic flow of at least q. Subsequently, the breakdown probability $P^{(i)}(q)$, which denotes the probability of a breakdown at location i when the traffic flow is q, can be estimated by

$$P^{(i)}(q) = \frac{Q^{(i)}(q)}{Q^{(i)}(q) + R^{(i)}(q)}. \tag{8.6}$$

Remark 8.4 *To avoid including "fake breakdowns" (e.g. a single vehicle driving unnecessarily slow at night), we pose the additional constraint on a breakdown that it does not happen before 5:00 in the morning. Indeed, multiple times we observed before 5:00, at a minimal traffic flow, a sudden drop of the average speed to just below the critical speed. We assume that such events are not relevant for estimating the breakdown distribution as this could, e.g., be a truck driving at its speed limit of* 80 *km/h.*

To reduce the complexity of the estimation method, we use a surrogate for the breakdown probabilities, obtained by fitting a cumulative normal distribution function, as is done in [10].

In Section 8.1, an intuitive description of a high-performance day was given. In this section we present a criterion to determine a quantitative definition for high-performance days. To this end, we employ the estimated breakdown probability in Equation (8.6), to find *unperturbed moments*. An unperturbed moment is a moment at which the probability of a breakdown is at least 0.5, but the expected breakdown did not occur, or more mathematically:

Definition 8.3 (Unperturbed moment) *An unperturbed moment, at location i on date j, is a moment $t_k^{(i,j)} \in \mathcal{T}^{(i,j)}$ with intensity $q_{t_k^{(i,j)}}^{(i,j)} \geq q_{upt}^{(i)}$ and speed $v_{t_k^{(i,j)}}^{(i,j)} \geq v_{crit}^{(i)}$ for which it holds that*

$$P^{(i)}\left(q_{t_k^{(i,j)}}^{(i,j)}\right) \geq 1/2 \quad \wedge \quad v_{t_{k+1}^{(i,j)}}^{(i,j)} \geq v_{crit}^{(i)},$$

where $q_{upt}^{(i)}$ is the smallest value of the traffic flow q such that $P^{(i)}(q) \geq 1/2$.

A plausible definition of a high-performance day follows naturally.

Definition 8.4 (High-performance day) *A high-performance day is a day with a large number of consecutive unperturbed moments in both time and space compared to other days.*

Note that a high-performance day is thereby a relative measure, as it will depend on the location how many unperturbed moments are generally present (some locations experience more variability in terms of breakdowns in relation to the traffic flow). Indeed, a certain level of freedom in the definition of high-performance days is required. For example, quantifications such as the top 0.05 percentile, though plausible in some cases, incorrectly imply the existence of high-performance days at any location. Furthermore, concretizations of the definition in terms of the number of unperturbed moments depend on the experimental region.

8.4 Key insights and validation

In this section, we present the results of our algorithm and validate the estimation methods. In particular, we study the results of the three steps of the algorithm and present several measures of the top 10 high-performance days. In addition, we take a closer look at what exactly a high-performance day looks like and how we can use our macroscopic data to visualize the dynamics of such days for the whole trajectory. We also provide a further investigation of the top 10 high-performance days to see whether some special circumstances might have caused the good traffic performance on those days. We investigate e.g. traffic accidents in the direct surroundings of the A15 and the weather conditions. Subsequently, we elaborate on several problems one might encounter when applying the method at a different location and how these problems could be tackled. Specifically, we state how we dealt with these problems and how we obtained the critical weight and the critical level of the MAPE.

8.4.1 Results and key insights

In Table 8.1, we present the results of the first two steps of the algorithm (i.e. estimating the critical speed and the breakdown probabilities respectively). We observe that the critical speed is roughly equal for the various locations. We

see a similar pattern for the estimated free-flow speeds, which are consistently about 10 km/h above the corresponding estimated critical speeds. We also see that the smallest value of the traffic flow for which the breakdown probability is at least 0.5, decreases along the trajectory, meaning that the last two locations experience breakdowns at a lower traffic flow than the first three locations. This makes sense considering the merge from 3 to 2 lanes at the fourth location.

Table 8.1: Columns 2-5 from left-to-right: the rounded estimated critical speed of location i, the rounded estimated free-flow speed of location i (based on the median of the free-flow speeds of the instances that were used to estimate the critical speed of location i), the number of instances used for estimating the critical speed of location i (out of a total of 261 weekdays), and the smallest traffic flow for which the breakdown probability is at least 0.5. The speeds are expressed in kilometers per hour and the traffic flows are expressed in vehicles per hour.

	$v_{\text{crit}}^{(i)}$	$v_{\text{free}}^{(i)}$	$\lvert\mathcal{V}_{\text{crit}}^{(i)}\rvert$	$q_{\text{upt}}^{(i)}$
Location 1	95.5	104.5	147	4358
Location 2	93	103	162	4019
Location 3	93	102	175	3901
Location 4	94.5	104.5	180	3195
Location 5	92.5	102.5	175	3164

In Figure 8.5, we present a scatter plot displaying the average number of unperturbed moments per location for each weekday of 2018. Additionally, the color of each point corresponds to the average breakdown probability of the unperturbed moments. We observe that the days can be grouped into roughly three categories: days with hardly any unperturbed moments, days with some unperturbed moments, and days with a relatively large number of unperturbed moments. It turns out that most days in the first group correspond to days with significantly less traffic, thus implying a low traffic flow and thereby a lack of unperturbed moments. For example, the gray points in Figure 8.5 often correspond to (school) holidays. The third group, however, is of major interest to us, as these are the high-performance days.

In Table 8.2, we present several measures of the top 10 high-performance days (based on Figure 8.5), corresponding to the fourth location. We choose to only present results for the fourth location, because averaging the speeds over the various locations requires a critical speed for the whole trajectory as a baseline (whose definition is not straightforward). To study the characteristics of these days, we investigate the average speed and average fraction of

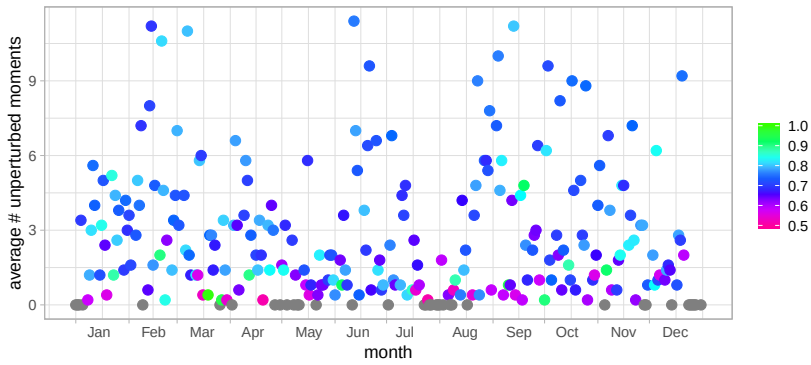


Figure 8.5: Plot of the average number of unperturbed moments for each weekday of 2018. The color of each point indicates the average breakdown probability of the unperturbed moments. In case no unperturbed moments occurred, the corresponding point is gray.

free-flow measurements. We look at three time intervals: the morning rush hour 6.30-9.30, outside peak hours 9.30-15.30 and the afternoon rush hour 15.30-19.00. We observe that, though all days show a relatively large number of unperturbed moments, the characteristics of the various days can differ greatly. For example, the top five high-performance days all have an average speed during the morning rush hour that is below the critical speed of location 4 (i.e. 94.5 km/h) and at least 10% of the measurements during the morning rush hour correspond to congestion, whereas the last three high-performance days show hardly any signs of congestion in the morning. We observe a similar pattern across all high-performance days: the mornings are significantly better (in terms of the average speed and the fraction free flow) than the afternoons. In fact, it seems that severe congestion during the afternoon was present during almost all high-performance days (only February 14 is an exception, see Subsection 8.4.2 for a potential explanation). Nevertheless, the mornings of the top 10 high-performance days are quite extraordinary, in particular when comparing the average speed and the fraction free flow with the median over all weekdays.

We now thoroughly study the traffic behavior during October 17, 2018. During this day, an average of 9 unperturbed moments was identified (see Table 8.2). This day is particularly interesting because of the seemingly large difference between the morning and afternoon rush hour. In fact, this day is the

Table 8.2: Several measures of the top 10 high-performance days, based on Figure 8.5, corresponding to the fourth location. The average speed is presented during the morning rush hour 6.30-9.30, outside peak hours 9.30-15.30, and during the afternoon rush hour 15.30-19.00, as well as the corresponding fraction free flow. The median over all weekdays of 2018 is presented as well.

	Average number unperturbed moments (per location)	Average speed morning rush hour	Average speed outside peak hours	Average speed afternoon rush hour	Fraction free flow morning rush hour	Fraction free flow outside peak hours	Fraction free flow afternoon rush hour
12-Jun	11.4	88.5	99.3	33.2	0.76	0.96	0.12
14-Feb	11.2	92.9	99.8	75.3	0.86	0.94	0.60
13-Sep	11.2	93.4	99.2	32.1	0.83	0.94	0.07
7-Mar	11	82.9	104.2	41.9	0.72	1.00	0.36
20-Feb	10.6	58.5	97.7	52.9	0.39	0.88	0.36
4-Sep	10	95.2	104.4	41.2	0.86	1.00	0.21
21-Jun	9.6	90.3	99.2	18.3	0.81	0.96	0.00
3-Oct	9.6	99.7	103.5	30.9	0.94	1.00	0.05
20-Dec	9.2	100.8	92.2	30.8	0.97	0.86	0.12
17-Oct	9	103.1	99.5	39.3	1.00	0.96	0.19
Median	2.2	71.2	99.4	44.0	0.51	0.94	0.21

Average speed legend	Fraction free flow legend
0.0	0.00
10.0	0.10
20.0	0.20
30.0	0.30
40.0	0.40
50.0	0.50
60.0	0.60
70.0	0.70
80.0	0.80
90.0	0.90
>94.5	1.00

only day in the top 10 high-performance days which does not show any congestion during the entire morning rush hour. In Figure 8.6, a joint time series of the average flow and average speed at the fourth location during this day is presented. As expected, we have a large number of unperturbed moments, mostly during the morning. The contrast between the morning and the afternoon is indeed interesting, as the breakdown, which remained absent in the morning, manifested in the afternoon at a lower traffic flow. This is in line with our probabilistic view on the occurrence of a breakdown (at least from a macroscopic point of view) and confirms that this morning was indeed extraordinary.

Additionally, one could employ visualizations to investigate the whole trajectory simultaneously, see Figure 8.7. We verified that the morning of October 17, 2018 was extraordinary at the fourth location and Figure 8.7 shows that this was the case for the whole trajectory. Indeed, we observe multiple unperturbed moments during the morning rush hour at each of the five locations. In particular, despite the high traffic flow (recall that unperturbed moments only occur at a traffic flow of at least 3164 vehicles per hour, see Table 8.1), we observe no significant speed decrease. Furthermore, as we expect based on Figure 8.6, a breakdown along the whole trajectory can clearly be seen around 15.20-15.30 (see Figure 8.7).

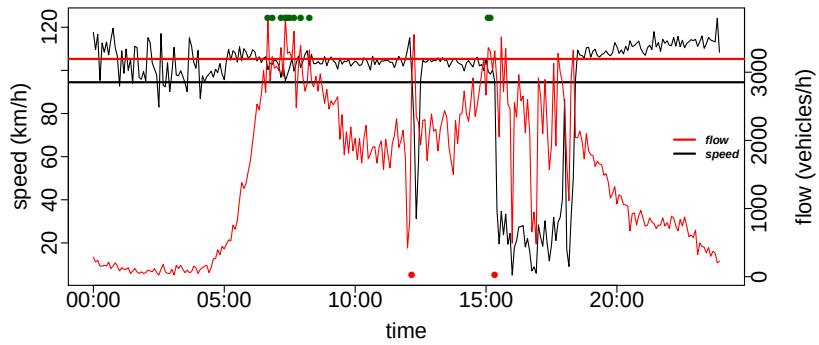


Figure 8.6: Time series of the average speed (black) and average flow (red) during October 17, 2018 at location 4. Unperturbed moments are indicated by a green dot and breakdowns are indicated by a red dot. The horizontal black line is the estimated critical speed and the horizontal red line is the smallest traffic flow for which the breakdown probability is at least 0.5.

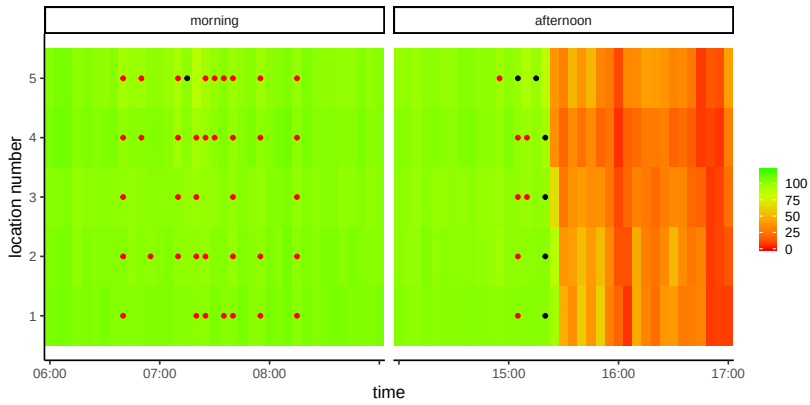


Figure 8.7: A space-time diagram of the morning rush hour and the afternoon of October 17, 2018. The average speed is displayed along the whole trajectory. Furthermore, breakdowns are marked with a black marker and unperturbed moments are marked with a red dot.

8.4.2 Further investigation of the high-performance days

A natural follow-up question would be in the direction of causality. Indeed, one could wonder *why* certain days exhibit extraordinary behavior in terms of an unexpected absence of traffic jams. We give some further details about the relevant circumstances during the 10 high-performance days in Table 8.2.

Among the 10 high-performance days are only three weekdays, namely Tuesday (3 times), Wednesday (4 times), and Thursday (3 times). Statistically, it is unlikely that there are no Mondays and Fridays, so it might be that there is a difference between different days. Oftentimes, on Fridays, the total traffic flow is relatively low during the morning rush hour and the traffic flow is not high enough to get to a 50% breakdown probability, implying the absence of unperturbed moments in the morning. Mondays do not exhibit the same low traffic flow, but, likely, there is a different explanation for the absence of Mondays in the top 10.

Table 8.3: Weather conditions at each of the top 10 days as recorded by Weerverleden.nl [195] for the city of Rotterdam, focusing on weather during day time, where cond. abbreviates condition. The sight conditions moderate (sight below 2 but above 1 kilometer) and bad (sight below 1 kilometer) in the table below occurred during the early morning and generally improved (considerably) during the day.

	Weather cond.	Sun %	Wind cond.	Sight cond.
14 Feb.	Dry	79%	Moderate breeze	Good
20 Feb.	Dry	15%	Light breeze	Good
7 Mar.	Dry	18%	Light breeze	Moderate
12 Jun.	Dry	16%	Gentle breeze	Good
21 Jun.	Dry	42%	Moderate breeze	Good
4 Sep.	Dry	24%	Light breeze	Moderate
13 Sep.	Dry	80%	Light breeze	Good
3 Oct.	Dry	73%	Gentle breeze	Good
17 Oct.	Dry	61%	Light breeze	Bad
20 Dec.	Dry, showery afternoon	3%	Moderate breeze	Good

Another, seemingly, important factor in the occurrence of high-performance days is the weather. Although we do not have access to weather measurements at the specific site, we have access to historical weather data of the city of Rotterdam (about 20 kilometers away) by means of the website Weerverleden.nl [195]. Although weather conditions may vary substantially over a 20 kilometer distance, we use the information from Rotterdam to get an indication of the weather

at the A15. We focus on weather aspects that potentially influence the traffic flow and give an overview in Table 8.3.

As can be observed in Table 8.3, the weather could generally be described as "fair" during each of the top 10 high-performance days. No strong winds, good sight, and dry conditions seem to be a common feature for each high-performance day/morning, even though during parts of some days the sight was not good (but this might also vary considerably over a 20 kilometer distance). The amount of sunshine does not seem to play an important role in the occurrence of high-performance days.

Other factors that might influence the traffic-flow performance are traffic incidents, maintenance, and holiday periods. We have investigated each of those aspects (as far as reasonably possible). It seems that there is only one traffic incident that might have influenced the traffic flow on the studied part of the A15 at any of the top 10 high-performance days. On February 14, there was a traffic incident at the nearby A16 which could have reduced the traffic flow on the A15 during the second part of the afternoon [169]. This, perhaps, clarifies the good traffic conditions during the afternoon of February 14, think e.g. of more homogeneity in the traffic flow, a relatively low inflow of traffic, or a different mixture of vehicle types. During 2018, there were several more incidents in the vicinity of the area of study, but those occurred at a relatively large distance and/or occurred well outside peak hours. It does not seem that maintenance activities had any effects (at least not during peak hours) where also road works on motorways in the vicinity have been considered. Also the effects of holiday periods seem relatively small. Only February 14 relates to a public holiday in the southern part of the Netherlands (carnival). Unfortunately, this day coincides with the aforementioned traffic incident on the A16, so it is difficult to pinpoint which of those effects is responsible for the observed traffic performance on February 14.

Summarizing, this investigation reveals some patterns, but it is difficult to state hard conclusions. It seems that particular days of the week are more prone to give rise to high-performance days and the weather also seems to have an influence. Upfront, one would also expect those features to have an influence on the quality of the traffic flow, so this is no surprise. Apart from that, it might be that traffic incidents and particular holidays have a (slight) impact on the traffic performance, but we would need more data/measurements to be sure.

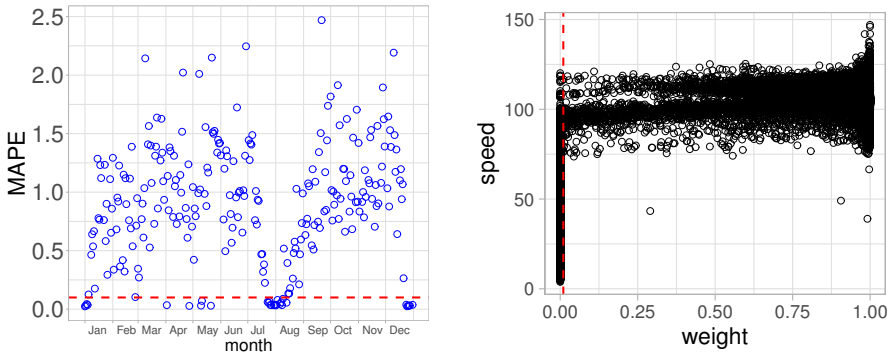
8.4.3 Validation

The critical speeds are estimated based on a labeling of the data points resulting from the robust regression method discussed in Subsection 8.3.2. As the exact shape of the fundamental diagram depends on the location, it is difficult to make general statements about the accuracy of the critical speed estimation. However, we can identify three possible issues: (i) little or no congestion occurred during a day; (ii) extreme congestion occurred during a day; and (iii) the free-flow speed was not (approximately) constant. We also present a way to determine whether or not those problems did arise (besides additional information about the experimental region). Finally, we conclude this section with a discussion on how to choose the critical weight, which is used to determine whether observations belong to the congestion set or the free-flow set.

Little or no congestion. In this case, robust regression might interpret a free-flow point with a relatively low speed as an outlier and therefore cause a free-flow point to be labeled as a congestion point. This leads to a higher estimate of the critical speed during that day. Though in our case it is not likely that the final estimate of the critical speed will be strongly influenced by several overestimates (considering that our experimental region is generally subject to heavy congestion), we still exclude days with little or no congestion. As mentioned in Subsection 8.3.2, we use the MAPE of the robust regression model presented in Equation (8.2) as a surrogate for the average congestion level. In Figure 8.8(a), a plot of the MAPE for the various days of location 1 is shown. We observe that, for example, during holiday periods, e.g. the beginning of January/end of December and the summer break, the MAPE is close to zero. Indeed, during those days the traffic flow was significantly lower and hardly any congestion occurred. Based on Figure 8.8(a) (and similar figures for the other locations), we decided to place the threshold at 0.1; instances with a MAPE of less than 0.1 are excluded when determining the critical speed, as in Equation (8.5).

Extreme congestion. Extreme congestion may lead to severe underestimations of the critical speed. One can imagine that if the number of congestion measurements becomes too large, not all congestion points will be observed as outliers by the robust regression method. In particular, what may happen is that robust regression fits a model through the congestion region, see also Remark 8.1. For MM-estimators, it is known that if more than half of the data points lie on a straight line through the origin, the final model will fit that line,

at least asymptotically (i.e. when the number of data points increases) [228]. This means that, if we assume a constant flow-density relation in free flow, the free-flow speed should be accurately estimated if more than half of the measurements correspond to free flow. However, because Equation (8.1) is only an approximate relation, the algorithm will be even more sensitive to a larger congestion set. In our case study, the fraction of free flow was generally well above 0.5. However, before employing robust regression to determine the critical speed, it is recommended one verifies that the average free-flow level is above 0.5. In case the congestion level is around 0.5 one should cautiously verify that the critical speed is correctly estimated (e.g. by studying the distribution of the estimated critical speed for the various days).



(a) (b)

Figure 8.8: A plot of the MAPE of the robust regression model for all days in (a) and a plot of the weights and corresponding speeds for location 1 in (b).

Non-constant free-flow speed. In case the free-flow speed is not constant, the structure of the fundamental diagram will change drastically. One example would be a decrease of the speed limit when the rush-hour lane is opened during peak hours. This could result in a free-flow speed curve, rather than a straight line, displaying an average speed decrease at high traffic flows. Such a scenario could be problematic for our algorithm, as the approximate flow-density relationship, presented in Equation (8.1), no longer holds. We suggest that one beforehand verifies that the free-flow speed is constant, either by using information about the experimental region or by studying the fundamental

diagram. In our case there was no dynamic speed limit and the fundamental diagrams showed no indication of a non-constant free-flow speed.

Critical weights. In Subsection 8.3.2, we introduced the critical weight, which is used to distinguish between congestion and free flow. The critical weight has been placed at 0.01, meaning that points with a weight below 0.01 are labeled as congestion. This value is determined using Figure 8.8(b), which shows a scatter plot of all speeds and corresponding weights of the first location. We observe that almost all low speeds (say speeds below 70 km/h), have a weight which is either zero or very close to zero. Speed-weight plots of the other four locations showed a similar pattern. Therefore, we conclude that a critical weight of 0.01 generally allows for a sensible labeling.

8.5 Conclusion

We have developed an algorithm to identify high-performance days based on an estimation of the critical speed and the breakdown probability. The algorithm is relatively straightforward and only requires two quantities: the average traffic flow and the average speed. The algorithm relies on the shape of the fundamental diagram. Each observation is classified as either free flow or congestion using robust regression. The critical speed is estimated as the line that separates these two sets. Using a non-parametric estimator for the breakdown probability, we are able to quantify both characteristics of a high-performance day (roughly speaking, high speed and high flow). The algorithm has been applied in a case study where we identify high-performance days on the A15 near Papendrecht.

A natural follow-up question would be in the direction of causality. Indeed, one could wonder *why* certain days exhibit extraordinary behavior, in terms of an unexpected absence of traffic jams. We have taken a look at some potential clarifications such as day of the week and the weather conditions in Subsection 8.4.2 and it seems that those two features have an influence on the high-performance days. At the same time, there are many more potential reasons why some days are high-performance days and others are not. A possible explanation could be traffic homogeneity: perhaps there were fewer trucks during the high-performance days, leading to fewer speed differences between vehicles. Alternatively, the answer may lie hidden in microscopic data: certain (desirable) behavioral characteristics of drivers might be over-represented during high-performance days. Other potentially influencing factors are the

occurrence of downstream traffic jams [34] or merging and/or lane changing actions [58]. Our algorithm perhaps provides a way towards reducing traffic jams from a different perspective and may lead to new insights as well as an easier investigation of countermeasures against traffic jams. This non-trivial extension is, however, beyond the scope of this chapter. Instead, we present this tool to facilitate further research into countermeasures against traffic jams, as the algorithm is able to identify which days need to be studied in more detail.

We must be critical of our approach as well, in particular in terms of generality. This mainly relates to the two (subjective) thresholds: the critical weight (to distinguish between congestion measurements and free-flow measurements) and the critical level of the MAPE of the regression model (to identify a lack of congestion). Both values were determined based on the five locations of the A15 Papendrecht 2018 data set. However, when testing the algorithm on other data sets, we still observed both a sensible labeling of the data points as well as a plausible recognition of days with little or no congestion. In fact, we tested the algorithm on data sets which violated the assumption of a constant free-flow speed and the algorithm still identified days with a high traffic flow and a striking absence of traffic jams. However, it is likely that at other locations, the critical weight and the critical level of the MAPE need to be adjusted.

Also, there is still room for improvement in terms of methodological aspects for the algorithm designed in the current chapter. In particular, one may want to employ more advanced estimators for the breakdown probability. The current non-parametric estimator is fully generic which, despite contributing to the generality of the method, may lack precision as certain road-specific parameters (e.g. the number of lanes or the speed limit) are not accounted for.

Chapter 9

Conclusions and future work

In this chapter we briefly reflect on the obtained results in this thesis and how they contribute to the existing literature in Section 9.1. We also discuss some topics for future research that we did not cover in the preceding chapters. Those topics for example relate to themes which link to multiple chapters and/or do not have a direct application in road-traffic engineering. We do this in Section 9.2.

9.1 Summary of contributions in this thesis

We have presented novel results for several models, but most of the contributions in this thesis relate directly or indirectly to the FCTL queue. Focusing on the FCTL queue and its extensions for now, we have extended the available methodologies to analyze the FCTL queue; we have obtained a Halfin-Whitt type of scaling for the FCTL queue; and we studied several generalizations of the FCTL queue.

We started this thesis with various chapters about the FCTL queue, a traffic-light model with fixed settings. This remains an important topic of study as we argued in Subsections 1.2.1 and 1.2.3: it is e.g. still applied in practice according to [152]. In Chapter 2, we derived a contour-integral expression for the PGF of the overflow queue for the FCTL queue which is a novel way to obtain the PGF of the overflow queue in the FCTL queue. The FCTL queue and its generalizations, for which we derived contour-integral expressions in Theorem 2.2, seem to be part of the more general framework presented in [147]. A

benefit of these contour-integral expressions is that they avoid the need to find roots, which, until recently, seemed to be unavoidable in the analysis of many queueing systems like the FCTL queue.

Another benefit of the contour-integral expression for the FCTL queue which we derived in Chapter 2, is that it allows for an asymptotic analysis. It would be much more difficult to obtain the convergence results with the root-based expression. We present the asymptotic analysis in Chapter 3. We introduce a scaling which is reminiscent of the traditional Halfin-Whitt scaling [89]. It leads to a Quality-and-Efficiency-Driven type of regime when the cycle length grows to infinity, as we demonstrate in Theorem 3.1. We e.g. have that the probability that the overflow queue is empty, is strictly between 0 and 1. This implies that, even though the load on the queue increases to 1 with increasing cycle lengths (i.e. we do not have overcapacity), we still manage to have an empty overflow queue. It is conceivable that similar asymptotic results can be derived for (part of) the more general set of models considered in e.g. [147]. In fact, they may hold for an even larger set of models, as we demonstrate by means of simulation in Chapter 4. We show that similar empty-queue results hold for several types of polling models, among which is the standard k-limited polling model, which do not seem to belong to the set of models discussed in [147].

The asymptotic results for the FCTL queue derived in Chapter 3 also give rise to new, accurate approximations for e.g. the mean overflow queue, even when the cycle length for the traffic light is as small as one minute. The developed asymptotic theory thus in turn allows us to give general insights into practically relevant scenarios and to find a general rule-of-thumb of how to choose the green times. Moreover, instead of a complicated expression in terms of an involved contour-integral expression or in terms of roots, we are able to give a relatively simple, approximating formula for the mean overflow queue. As a last bonus, we note that the approximating formulas can be used to find (approximately) optimal traffic-light settings. We have demonstrated this in Chapter 3 and we have shown that the approximations and optimization strategies seem to yield accurate and close-to-optimal results.

Another practically relevant result is the generalization of the FCTL queue which is introduced in Chapter 5. In practice, a right-turning and straight-going flow of vehicles might share a lane and receive a green light simultaneously. If, moreover, there is a crossing for pedestrians on the turning flow that receives a green light at the same moment, there might be pedestrians that block a turning vehicle, which would, in turn, block other vehicles. Especially if there are many pedestrians crossing and many turning vehicles, the effect on the queueing process could be quite substantial. The capacity of the intersection decreases and

the mean overflow queue might increase, as is also demonstrated in Chapter 5. The extension of the FCTL queue that we formulated in Chapter 5 is thus of practical relevance as it allows us to take the influence of pedestrians into account if they directly interact with vehicles. We also studied this model with a general number of lanes (which might be blocked all at once). In particular, we are also able to study the FCTL queue with multiple lanes, which is another extension of the FCTL queue.

Even though the FCTL queue is an important traffic-light control strategy as argued above, there are also benefits of vehicle-actuated traffic lights. Unfortunately, the queueing models which describe such strategies, such as a k-limited polling model, are mostly intractable. Although several approximation schemes for such queueing models have already been developed, we designed a novel approximation scheme which is presented in Chapter 6. Our method generally seems to yield accurate approximations for a large set of queueing models which is useful for applying, e.g., k-limited polling models in road-traffic models and in other application areas. Our method might for example assist in choosing the k_i. We mainly focused on models with two queues, yet we also studied two models with more than two queues. The approximation scheme slows down when more queues have to be considered simultaneously and/or if the load/vehicle-to-capacity on the queues approaches 1. The former relates to the often-encountered curse-of-dimensionality, which translates (at least in our case) to a quickly increasing computational complexity when the number of queues increases.

The difficulty of studying queueing models with a dimension of two or higher, also pops up in our investigation of future traffic-light strategies in Chapter 7. Also in this chapter, we need to resort to an approximation scheme to obtain a performance analysis which is not solely based on simulation. The approximation scheme in Chapter 7 is different in nature than the one developed in Chapter 6 and, e.g., performs well when the vehicle-to-capacity ratio is close to 1. We specifically study a model with solely autonomous vehicles. We have developed a framework in which the autonomous vehicles create platoons among themselves; drive to the intersection in a coordinated manner, such that they cross the intersection close to one another; and such that they cross at high speed. We both present algorithms for the platoon formation of the autonomous vehicles and derive closed-form expressions for the trajectories that the vehicles might drive. Moreover, we have developed a framework to assess some of the performance characteristics of the model. We demonstrate that, under several natural assumptions like the headway being smaller for autonomous vehicles than for conventional vehicles, significant performance gains can be obtained.

Chapter 8 has a more practical orientation than the other chapters. The conducted research was inspired by a question from De Verkeersonderneming and has led to insights into the behavior of traffic on highways. We have developed an algorithm to automatically identify high-performance days during which there was both a high traffic volume and no traffic jam, which is a desirable combination. Unfortunately, at this moment we do not know *why* there are large differences between various days. Seemingly, the traffic conditions are similar in the sense that there is a high traffic volume, yet at some days a breakdown occurs while at others such a breakdown does not occur. It would be very interesting to see whether there are structural differences between high-performance and regular days that we are not (yet) aware of.

More topics for future research are discussed in the next section.

9.2 Suggestions for extensions and future research

Having described our main contributions, we note that there is plenty of room for further research, both in the realm of queueing theory and transportation research as we indicated in the previous chapters. Here, we give a further list of directions in which research can be developed, especially focusing on the topics that relate to multiple chapters or to topics outside of road-traffic engineering.

- *Cyclic queueing models.* The FCTL queue belongs to a much larger class of cyclic queueing models related to vehicle dispatching with uncertain arrivals and bulk services [163, 164, 207]. A broad variety of transportation and manufacturing systems can be modeled in this way, including batch production systems, bulk movements of goods in a factory, truck shipments, and bus transportation. Within this class, many different rules can be considered that apply to customer arrivals and vehicle departures within a cycle. One could think of vehicle-cancellation policies that hold a vehicle until the queue length reaches a specified threshold. The FCTL assumption can also be viewed as a special rule that influences the dynamics within a cycle and it seems that a contour-integral type of expression for the PGF of relevant steady-state queue-length distributions can be derived for many of the models in the class of cyclic queueing models with e.g. vehicle dispatching, uncertain arrivals, and bulk services. It might also be possible to find a Halfin-Whitt type of scaling for this more general set of models and optimal allocation schemes are then probably within reach.

- *SELSPs.* Potential application areas of cyclic queueing models are traffic related, as we demonstrated, but the general set of models may have a much broader applicability. One example is in the logistic area, such as for special cases of the Stochastic Economic Lot Scheduling Problem (SELSP), see e.g. [220] for an overview of SELSPs. We would then specifically think about the case with a fixed production sequence, a cycle with a fixed length, and a global lot sizing policy, i.e. lot-sizing decisions may depend on the complete state of the system. This is an underexposed strategy according to [220].

- *Models with a Halfin-Whitt scaling.* As is demonstrated in Chapter 4, the Halfin-Whitt type of scaling rule as introduced in Chapter 3 for the FCTL queue yields favorable asymptotic properties. Using the same scaling rule, the same type of properties can probably be obtained for the set of cyclic queueing models considered in the first extension described in this subsection. All those models are essentially one-dimensional queueing models. The queueing models considered in Chapter 4 are more-dimensional instead and thus behave in a fundamentally different way, but exhibit similar Halfin-Whitt type of properties when applying a similar scaling as in the FCTL queue. It would be very interesting to find a more general, potentially overarching, set of queueing models for which Halfin-Whitt type of asymptotic results can be derived. The asymptotic properties for this more general set of models might be shown by means of simulation, but obtaining the exact limiting process (as we did for the FCTL queue) is also of interest.

- *Networks of intersections.* Extending the results that we obtained for isolated intersections to a setting with a network of intersections, is an interesting topic for future research. Networks of intersections/queueing models are usually difficult to analyze, although they are practically very relevant as is also indicated in e.g. [152]. We advocate to investigate control strategies for networks of traffic lights, using ideas stemming from both Chapters 3 and 4, especially because such an investigation might lead to structural insights besides the work that has already been done in this direction. Examples are the use of aggregation-disaggregation and decomposition techniques (see e.g. [155]) and the use of simulation-based optimization methods (see e.g. [50]).

  Another example where we need additional research to go from isolated intersections to a network of intersections is the PFA setting studied in

Chapter 7. In a network of intersections there are several complications. Firstly, the arrival processes of vehicles become dependent. Moreover, the interplay between various intersections is non-trivial (think e.g. of spillback effects). As such, a specific study on how our PFAs perform in a network scenario is of interest.

- *Multiple streams of vehicles receiving a green light.* We have mostly studied traffic-light strategies where *one* stream of vehicles receives a green light. In practice, multiple non-conflicting streams of vehicles often receive a green light simultaneously and our models would have to be adjusted to account for this.

  For example, the PFAs that we considered in Chapter 7 do not directly allow for such a scenario, but our algorithms can probably be extended to account for this. However, the delay characteristics change when multiple streams of vehicles receive a green light at the same time and we would need to adjust the approximation scheme that we devised in Chapter 7.

  The methods developed for the bFCTL queue in Chapter 5 can still be used if multiple streams of vehicles receive a green light at the same time, as long as the streams are non-conflicting. However, this is not the case if we consider a scenario where two opposing streams of vehicles receive a green traffic light simultaneously and if there is a mixture of vehicles turning left and heading straight. The left-turning traffic might be blocked by vehicles which receive a green light from the opposing stream. Depending on the exact characteristics, like both streams of vehicles having some left-turning vehicles or not, the model will probably prove to be more complex than the model in Chapter 5. Probably, a two-dimensional analysis of the queue-length distribution is unavoidable if two opposing streams both receive a green light and vehicles in both streams might be blocked by vehicles from the other stream. This would lead to a more involved analysis.

  Concluding, traffic-light models significantly change and get more complicated if there are multiple, potentially conflicting streams of vehicles receiving a green light at the same time. Further research is needed to deal with such situations appropriately.

As such, there are many relevant extensions/generalizations, both in- and outside road-traffic models, of the models that we studied in this thesis. There is thus ample room for future research.

Bibliography

[1] J. Abate, G. L. Choudhury, and W. Whitt. Calculation of the $GI/G/1$ waiting time distribution and its cumulants from Pollaczek's formulas. *Archiv für Elektronik und Übertragungstechnik*, 47(5/6):311–321, 1993.

[2] J. Abate and W. Whitt. Numerical inversion of probability generating functions. *Operations Research Letters*, 12(4):245–251, 1992.

[3] I. J. B. F. Adan. *A Compensation Approach for Queueing Problems*. PhD thesis, Eindhoven Unversity of Technology, 1991.

[4] I. J. B. F. Adan, O. J. Boxma, and J. A. C. Resing. Queueing models with multiple waiting lines. *Queueing Systems*, 37(1-3):65–98, 2001.

[5] I. J. B. F. Adan, J. S. H. van Leeuwaarden, and J. Selen. Analysis of structured Markov processes. *arXiv preprint arXiv:1709.09060*, 2017.

[6] I. J. B. F. Adan, J. S. H. van Leeuwaarden, and E. M. M. Winands. On the application of Rouché's theorem in queueing theory. *Operations Research Letters*, 34(3):355–360, 2006.

[7] I. J. B. F. Adan and G. Weiss. A skill based parallel service system under FCFS-ALIS—steady state, overloads, and abandonments. *Stochastic Systems*, 4(1):250–299, 2014.

[8] W. K. M. Alhajyaseen, M. Asano, and H. Nakamura. Left-turn gap acceptance models considering pedestrian movement characteristics. *Accident Analysis and Prevention*, 50:175–185, 2013.

[9] D. P. Allen, J. E. Hummer, N. M. Rouphail, and J. S. Milazzo. Effect of bicycles on capacity of signalized intersections. *Transportation Research Record*, 1646(1):87–95, 1998.

[10] P. Arnesen and O. A. Hjelkrem. An estimator for traffic breakdown probability based on classification of transitional breakdown events. *Transportation Science*, 52(3):593–602, 2017.

[11] X. Bai. *Performance Bounds for Random Walks in the Positive Orthant*. PhD thesis, University of Twente, 2018.

[12] P. Bergendorff, D. W. Hearn, and M. V. Ramana. Congestion toll pricing of traffic networks. In *Network Optimization*, pages 51–71. Springer, 1997.

[13] J. P. C. Blanc. Performance analysis and optimization with the power-series algorithm. In *Performance Evaluation of Computer and Communication Systems*, pages 53–80. Springer, 1993.

[14] J. P. C. Blanc, R. Iasnogorodski, and P. Nain. Analysis of the $M/GI/1 \rightarrow ./M/1$ queueing model. *Queueing Systems*, 3(2):129–156, 1988.

[15] J. H. Blanchet and P. W. Glynn. Complete corrected diffusion approximations for the maximum of a random walk. *The Annals of Applied Probability*, 16(2):951–983, 2006.

[16] M. A. A. Boon. *Polling Models: From Theory to Traffic Intersections*. PhD thesis, Eindhoven University of Technology, 2011.

[17] M. A. A. Boon, I. J. B. F. Adan, E. M. M. Winands, and D. G. Down. Delays at signalized intersections with exhaustive traffic control. *Probability in the Engineering and Informational Sciences*, 26(3):337–373, 2012.

[18] M. A. A. Boon, R. D. van der Mei, and E. M. M. Winands. Applications of polling systems. *Surveys in Operations Research and Management Science*, 16(2):67–82, 2011.

[19] M. A. A. Boon and J. S. H. van Leeuwaarden. Networks of fixed-cycle intersections. *Transportation Research Part B: Methodological*, 117:254–271, 2018.

[20] M. A. A. Boon and E. M. M. Winands. Heavy-traffic analysis of k-limited polling systems. *Probability in the Engineering and Informational Sciences*, 28(4):451–471, 2014.

[21] M. A. A. Boon and E. M. M. Winands. Critically loaded k-limited polling systems. In *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools*, pages 95–102. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.

[22] M. A. A. Boon, E. M. M. Winands, I. J. B. F. Adan, and A. C. C. van Wijk. Closed-form waiting time approximations for polling systems. *Performance Evaluation*, 68(3):290–306, 2011.

[23] M. A. M. W. Borm, B. Patch, T. Taimre, and I. J. B. F. Adan. Evaluation of a self-organized traffic light policy. In *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools*, pages 135–136, 2016.

[24] S. C. Borst and O. J. Boxma. Polling: past, present, and perspective. *TOP*, 26(3):335–369, 2018.

[25] S. C. Borst, A. Mandelbaum, and M. I. Reiman. Dimensioning large call centers. *Operations Research*, 52(1):17–34, 2004.

[26] R. J. Boucherie and N. M. van Dijk. *Queueing Networks: A Fundamental Approach*, volume 154. Springer US, 2011.

[27] O. J. Boxma and H. Daduna. The cyclic queue and the tandem queue. *Queueing Systems*, 77(3):275–295, 2014.

[28] O. J. Boxma and W. P. Groenendijk. Pseudo-conservation laws in cyclic-service systems. *Journal of Applied Probability*, 24(4):949–964, 1987.

[29] O. J. Boxma and W. P. Groenendijk. *Two queues with alternating service and switching times*, pages 261–282. Queueing theory and its applications (Liber amicorum for J.W. Cohen). North-Holland Publishing Company, 1988.

[30] O. J. Boxma, O. Kella, and K. M. Kosiński. Queue lengths and workloads in polling systems. *Operations Research Letters*, 39(6):401–405, 2011.

[31] O. J. Boxma and G. J. J. A. N. van Houtum. The compensation approach applied to a 2×2 switch. *Probability in the Engineering and Informational Sciences*, 7(4):471–493, 1993.

[32] D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12(1):258–268, 1968, in German.

[33] D. Bremmer. Dit zijn de 20 duurste files van Nederland. `https://www.ad.nl/economie/dit-zijn-de-20-duurste-files-van-nederland~a4803756/`, 2019. Date accessed: 2019-08-01, in Dutch.

[34] W. Brilon, J. Geistefeldt, and M. Regler. Reliability of freeway traffic flow: a stochastic concept of capacity. In *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, pages 125–144. Citeseer, 2005.

[35] R. L. Burden and J. D. Faires. *Numerical Analysis*. Cengage Learning, 9th edition, 2010.

[36] P. J. Burke. The output process of a stationary $M/M/s$ queueing system. *The Annals of Mathematical Statistics*, 39(4):1144–1152, 1968.

[37] G. F. Carrier, M. Krook, and C. E. Pearson. *Functions of a Complex Variable: Theory and Technique*, volume 49. SIAM, 2005.

[38] C. Chai and Y. D. Wong. Traffic performance of shared lanes at signalized intersections based on cellular automata modeling. *Journal of Advanced Transportation*, 48(8):1051–1065, 2014.

[39] J. T. Chang and Y. Peres. Ladder heights, Gaussian random walks and the Riemann zeta function. *The Annals of Probability*, 25(2):787–802, 1997.

[40] N. A. Chaudhary, V. G. Kovvali, and S. M. M. Alam. Guidelines for selecting signal timing software. Technical report, Texas Transportation Institute, Texas A&M University System, 2002.

[41] B. Chen and H. H. Cheng. A review of the applications of agent technology in traffic and transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):485–497, 2010.

[42] J. Chen, Z. Li, W. Wang, and H. Jiang. Evaluating bicycle–vehicle conflicts and delays on urban streets with bike lane and on-street parking. *Transportation Letters*, 10(1):1–11, 2018.

[43] L. Chen and C. Englund. Cooperative intersection management: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2015.

[44] P. Chen, H. Nakamura, and M. Asano. Saturation flow rate analysis for shared left-turn lane at signalized intersections in Japan. *Procedia-Social and Behavioral Sciences*, 16:548–559, 2011.

[45] P. Chen, H. Qi, and J. Sun. Investigation of saturation flow on shared right-turn lane at signalized intersections. *Transportation Research Record*, 2461(1):66–75, 2014.

[46] X. Chen, C. Shao, and Y. Hao. Influence of pedestrian traffic on capacity of right-turning movements at signalized intersections. *Transportation Research Record*, 2073(1):114–124, 2008.

[47] X. Chen, C. Shao, and H. Yue. Influence of bicycle traffic on capacity of typical signalized intersection. *Tsinghua Science and Technology*, 12(2):198–203, 2007.

[48] Y. Chen. *Random Walks in the Quarter-Plane: Invariant Measures and Performance Bounds*. PhD thesis, University of Twente, 2015.

[49] C. Cheng, Y. Du, L. Sun, and Y. Ji. Review on theoretical delay estimation model for signalized intersections. *Transport Reviews*, 36(4):479–499, 2016.

[50] L. Chong and C. Osorio. A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transportation Science*, 52(3):637–656, 2018.

[51] G. L. Choudhury and D. M. Lucantoni. Numerical computation of the moments of a probability distribution from its transform. *Operations Research*, 44(2):368–381, 1996.

[52] G. L. Choudhury and W. Whitt. Computing distributions and moments in polling models by numerical transform inversion. *Performance Evaluation*, 25:267–292, 1996.

[53] A. J. H. Clayton. Road traffic calculations. *Journal of the Institution of Civil Engineers*, 16:247–264, 1941.

[54] E. G. Coffman Jr, A. A. Puhalskii, and M. I. Reiman. Polling systems in heavy traffic: a Bessel process limit. *Mathematics of Operations Research*, 23(2):257–304, 1998.

[55] J. W. Cohen. *The Single Server Queue*. Elsevier Science Publishers B.V., 1982.

[56] J. W. Cohen. *Analysis of Random Walks*. IOS Press (Amsterdam), 1992.

[57] J. W. Cohen and O. J. Boxma. *Boundary Value Problems in Queueing System Analysis*. North-Holland Publishing Company, 1983.

[58] B. Coifman and S. Kim. Extended bottlenecks, the fundamental relationship, and capacity drop on freeways. *Procedia-Social and Behavioral Sciences*, 17:44–57, 2011.

[59] C. Comte and J. L. Dorsman. Pass-and-swap queues. *Queueing Systems*, pages 275–331, 2021.

[60] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, 1996.

[61] COWI and PTV Group. The Oslo study - how autonomous cars may change transport in cities. `https://www.ovmagazine.nl/wp-content/uploads/2019/05/NO_Report_RUTER_Frokostmoede-410-gecomprimeerd.pdf`, 2019. Date accessed: 2020-03-25.

[62] CROW. Microsoft Word - Stappenplan iVRI_bewerkt.docx. `https://www.crow.nl/downloads/pdf/verkeer-en-vervoer/verkeersmanagement/verkeersregelinstallaties/stappenplan-ivri`, 2020. Date accessed: 2020-08-20, in Dutch.

[63] C. F. Daganzo. Some properties of polling systems. *Queueing Systems*, 6(1):137–154, 1990.

[64] J. N. Darroch. On the traffic-light queue. *The Annals of Mathematical Statistics*, 35:380–388, 1964.

[65] J. N. Darroch, G. F. Newell, and R. W. J. Morris. Queues for a vehicle-actuated traffic light. *Operations Research*, 12(6):882–895, 1964.

[66] De Ingenieur. Predictive traffic lights in Helmond | De Ingenieur. `https://www.deingenieur.nl/artikel/predictive-traffic-lights-in-helmond`, 2018. Date accessed: 2020-08-20.

[67] G. Dervisoglu, G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya. Automatic calibration of the fundamental diagram and empirical observations on capacity. In *Transportation Research Board 88th Annual Meeting*, volume 15, pages 31–59, 2009.

[68] A. Devos, J. Walraevens, D. Fiems, and H. Bruneel. Approximations for the performance evaluation of a discrete-time two-class queue with an alternating service discipline. *Annals of Operations Research*, pages 1–27, 2020.

[69] W. Dib, A. Chasse, P. Moulin, A. Sciarretta, and G. Corde. Optimal energy management for an electric vehicle in eco-driving applications. *Control Engineering Practice*, 29:299–307, 2014.

[70] J. Dong and H. S. Mahmassani. Flow breakdown and travel time reliability. *Transportation Research Record*, 2124(1):203–212, 2009.

[71] J. L. Dorsman, R. D. Van der Mei, and E. M. M. Winands. A new method for deriving waiting-time approximations in polling systems with renewal arrivals. *Stochastic Models*, 27(2):318–332, 2011.

[72] D. Ettema, J. Knockaert, and E. T. Verhoef. Using incentives as traffic management tool: empirical results of the "peak avoidance" experiment. *Transportation Letters*, 2(1):39–51, 2010.

[73] G. Fayolle, R. Iasnogorodski, and V. Malyshev. *Random Walks in the Quarter Plane: Algebraic Methods, Boundary Value Problems, Applications to Queueing Systems and Analytic Combinatorics*, volume 40. Springer, 2017.

[74] L. Flatto. Two parallel queues created by arrivals with two demands II. *SIAM Journal on Applied Mathematics*, 45(5):861–878, 1985.

[75] L. Flatto and S. Hahn. Two parallel queues created by arrivals with two demands I. *SIAM Journal on Applied Mathematics*, 44(5):1041–1053, 1984.

[76] S. T. G. Fleuren. *Optimizing Pre-Timed Control at Isolated Intersections*. PhD thesis, Eindhoven University of Technology, 2017.

[77] S. T. G. Fleuren and A. A. J. Lefeber. Optimizing fixed-time control at isolated intersections: part I: a single green interval per traffic light. Technical report, Eindhoven University of Technology, 2016.

[78] C. Fricker and M. R. Jaibi. Monotonicity and stability of periodic polling models. *Queueing Systems*, 15(1-4):211–238, 1994.

[79] S. W. Fuhrmann. Performance analysis of a class of cyclic schedules. Technical report, Bell Laboratories, 1981.

[80] H. K. Gaddam and K. R. Rao. Speed-density functional relationship for heterogeneous traffic data: a statistical and theoretical investigation. *Journal of Modern Transportation*, 27(1):61–74, 2019.

[81] F. Garwood. An application of the theory of probability to the operation of vehicular-controlled traffic signals. *Supplement to the Journal of the Royal Statistical Society*, 7(1):65–77, 1940.

[82] German Aerospace Center (DLR) and others. Visualization - SUMO Documentation. `https://sumo.dlr.de/docs/Tools/Visualization.html`, 2021. Date accessed: 2021-06-05.

[83] P. W. Glynn. Diffusion approximations. *Handbooks in Operations Research and Management Science*, 2:145–198, 1990.

[84] M. Goh. Congestion management and electronic road pricing in Singapore. *Journal of Transport Geography*, 10(1):29–38, 2002.

[85] J. Goseling, R. J. Boucherie, and J. C. W. van Ommeren. A linear programming approach to error bounds for random walks in the quarter-plane. *Kybernetika*, 52(5):757–784, 2016.

[86] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2001.

[87] Y. Guo, Q. Yu, Y. Zhang, and J. Rong. Effect of bicycles on the saturation flow rate of turning vehicles at signalized intersections. *Journal of Transportation Engineering*, 138(1):21–30, 2012.

[88] R. Haijema. *Solving Large Structured Markov Decision Problems for Perishable Inventory Management and Traffic Control*. PhD thesis, University of Amsterdam, 2008.

[89] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29(3):567–588, 1981.

[90] A. Hamilton, B. Waterson, T. Cherrett, A. Robinson, and I. Snell. The evolution of urban traffic control: changing policy and technology. *Transportation Planning and Technology*, 36(1):24–43, 2013.

[91] A. Hegyi, S. P. Hoogendoorn, M. Schreuder, H. Stoelhorst, and F. Viti. SPECIALIST: a dynamic speed limit control algorithm based on shock wave theory. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 827–832. IEEE, 2008.

[92] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[93] D. Helbing and A. Mazloumian. Operation regimes and slower-is-faster effect in the control of traffic intersections. *The European Physical Journal B-Condensed Matter and Complex Systems*, 70(2):257–274, 2009.

[94] G. Hooghiemstra, M. Keane, and S. van de Ree. Power series for stationary distributions of coupled processor models. *SIAM Journal on Applied Mathematics*, 48(5):1159–1166, 1988.

[95] N. B. Hounsell and M. McDonald. Urban network traffic control. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 215(4):325–334, 2001.

[96] S. Huang, A. Toriumi, and T. Oguchi. Random nature of shared left-turn lanes at signalized intersections. In *2020 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3159–3166. IEEE, 2020.

[97] J. R. Jackson. Networks of waiting lines. *Operations Research*, 5(4):518–521, 1957.

[98] A. J. E. M. Janssen and J. S. H. van Leeuwaarden. Analytic computation schemes for the discrete-time bulk service queue. *Queueing Systems*, 50(2-3):141–163, 2005.

[99] A. J. E. M. Janssen and J. S. H. van Leeuwaarden. On Lerch's transcendent and the Gaussian random walk. *The Annals of Applied Probability*, 17(2):421–439, 2007.

[100] A. J. E. M. Janssen and J. S. H. van Leeuwaarden. 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.

[101] A. J. E. M. Janssen, J. S. H. van Leeuwaarden, and B. W. J. Mathijsen. Novel heavy-traffic regimes for large-scale service systems. *SIAM Journal on Applied Mathematics*, 75(2):787–812, 2015.

[102] S. Karlin and H. M. Taylor. *A First Course in Stochastic Processes*. Academic Press, 1975.

[103] F. P. Kelly. *Reversibility and Stochastic Networks*, volume 85. John Wiley, 1979.

[104] F. P. Kelly and J. Walrand. Networks of quasi-reversible nodes. In *Applied Probability-Computer Science: The Interface Volume 1*, pages 3–29. Birkhäuser Boston, 1982.

[105] B. S. Kerner. *Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory*. Springer Science and Business Media, 2009.

[106] B. S. Kerner and H. Rehborn. Experimental properties of phase transitions in traffic flow. *Physical Review Letters*, 79(20):4030–4033, 1997.

[107] K. B. Kesur. Optimization of mixed cycle length traffic signals. *Journal of Advanced Transportation*, 48(5):431–442, 2014.

[108] M. Khayatian, M. Mehrabian, E. Andert, R. Dedinsky, S. Choudhary, Y. Lou, and A. Shirvastava. A survey on intersection management of connected autonomous vehicles. *ACM Transactions on Cyber-Physical Systems*, 4(4):1–27, 2020.

[109] S. Kikuchi, N. Kronprasert, and M. Kii. Lengths of turn lanes on intersection approaches: three-branch fork lanes – left-turn, through, and right-turn lanes. *Transportation Research Record*, 2023(1):92–101, 2007.

[110] J. F. C. Kingman. On queues in heavy traffic. *Journal of the Royal Statistical Society: Series B (Methodological)*, 24(2):383–392, 1962.

[111] L. Kleinrock. *Message Delay in Communication Nets with Storage*. PhD thesis, Massachusetts Institute of Technology, 1963.

[112] L. Kleinrock. *Queueing Systems, Volume 2: Computer Applications*, volume 66. Wiley New York, 1976.

[113] V. L. Knoop. Traffic Flow Theory: An introduction with exercises, 2021.

[114] V. L. Knoop and W. Daamen. Automatic fitting procedure for the fundamental diagram. *Transportmetrica B: Transport Dynamics*, 5(2):129–144, 2017.

[115] L. M. C. Kockelkoren. Centralized merge control for FLEET, a material handling AGV system. Master's thesis, Eindhoven University of Technology, 2018.

[116] S. Lämmer and D. Helbing. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(04):P04019, 2008.

[117] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM, 1999.

[118] A. Lawitzky, D. Wollherr, and M. Buss. Energy optimal control to approach traffic lights. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4382–4387. IEEE, 2013.

[119] T. B. Lee. Waymo finally launches an actual public, driverless taxi service. `https://arstechnica.com/cars/2020/10/waymo-finally-launches-an-actual-public-driverless-taxi-service/`, 2020. Date accessed: 2021-01-18.

[120] T. T. Lee. $M/G/1/N$ queue with vacation time and limited service discipline. *Performance Evaluation*, 9(3):181–190, 1989.

[121] J. P. Lehoczky. Traffic intersection control and zero-switch queues under conditions of Markov chain dependence input. *Journal of Applied Probability*, 9(2):382–395, 1972.

[122] H. S. Levinson. Capacity of shared left-turn lanes – a simplified approach. *Transportation Research Record*, (1225), 1989.

[123] H. Levy and M. Sidi. Polling systems: applications, modeling, and optimization. *IEEE Transactions on Communications*, 38(10):1750–1760, 1990.

[124] H. Levy, M. Sidi, and O. J. Boxma. Dominance relations in polling systems. *Queueing Systems*, 6(1):155–171, 1990.

[125] H. Li and R. L. Bertini. Comparison of algorithms for systematic tracking of patterns of traffic congestion on freeways in Portland, Oregon. *Transportation Research Record*, 2178(1):101–110, 2010.

[126] M. Liu, M. Wang, and S. P. Hoogendoorn. Optimal platoon trajectory planning approach at arterials. *Transportation Research Record*, 2673(9):214–226, 2019.

[127] Y. Liu and G. Chang. An arterial signal optimization model for intersections experiencing queue spillback and lane blockage. *Transportation Research Part C: Emerging Technologies*, 19(1):130–144, 2011.

[128] Y. Liu, J. Yu, G. Chang, and S. Rahwanji. A lane-group based macroscopic model for signalized intersections account for shared lanes and blockages. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 639–644. IEEE, 2008.

[129] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582. IEEE, 2018.

[130] M. R. Lorenz and L. Elefteriadou. Defining freeway capacity as function of breakdown probability. *Transportation Research Record*, 1776(1):43–51, 2001.

[131] Z. Ma, J. Sun, and Y. Wang. A two-dimensional simulation model for modelling turning vehicles at mixed-flow intersections. *Transportation Research Part C: Emerging Technologies*, 75:103–119, 2017.

[132] K. J. Maes. Networks of fixed-cycle traffic-lights. Master's thesis, Eindhoven University of Technology, 2015.

[133] D. Miculescu and S. Karaman. Polling-systems-based control of high-performance provably-safe autonomous intersections. In *IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 1417–1423. IEEE, 2014.

[134] D. Miculescu and S. Karaman. Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals. *IEEE Transactions on Automatic Control*, 65(2):680–694, 2019.

[135] J. S. Milazzo, N. M. Rouphail, J. E. Hummer, and D. P. Allen. Effect of pedestrians on capacity of signalized intersections. *Transportation Research Record*, 1646(1):37–46, 1998.

[136] M. M. Minderhoud, H. Botma, and P. H. L. Bovy. Assessment of roadway capacity estimation methods. *Transportation Research Record*, 1572(1):59–67, 1997.

[137] P. B. Mirchandani and N. Zou. Queuing models for analysis of traffic adaptive signal control. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):50–59, 2007.

[138] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis*. John Wiley and Sons, 2012.

[139] NDW. Home - Nationale Databank Wegverkeersgegevens. `https://english.ndw.nu/`, 2019. Date accessed: 2021-09-30.

[140] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press, 1981.

[141] G. F. Newell. Queues for a fixed-cycle traffic light. *The Annals of Mathematical Statistics*, 31(3):589–597, 1960.

[142] G. F. Newell. Properties of vehicle-actuated signals: I. one-way streets. *Transportation Science*, 3(1):30–52, 1969.

[143] G. F. Newell and E. E. Osuna. Properties of vehicle-actuated signals: II. two-way streets. *Transportation Science*, 3(2):99–125, 1969.

[144] A. Oblakova, A. Al Hanbali, R. J. Boucherie, and J. C. W. van Ommeren. Green wave analysis in a tandem of traffic-light intersections. *Memorandum Faculty of Mathematical Sciences*, (2062), 2017.

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

[146] A. Oblakova, A. Al Hanbali, R. J. Boucherie, J. C. W. van Ommeren, and W. H. M. Zijm. 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.

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

[148] K. Ohno. Computational algorithm for a fixed cycle traffic signal and new approximate expressions for average delay. *Transportation Science*, 12(1):29–47, 1978.

[149] R. Ong, F. Pinelli, R. Trasarti, M. Nanni, C. Renso, S. Rinzivillo, and F. Giannotti. Traffic jams detection using flock mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–653. Springer, 2011.

[150] C. Osorio and M. Bierlaire. An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, 196(3):996–1007, 2009.

[151] C. Osorio and M. Bierlaire. A simulation-based optimization framework for urban transportation problems. *Operations Research*, 61(6):1333–1345, 2013.

[152] C. Osorio, X. Chen, J. Gao, M. Talas, and M. Marsico. A scalable algorithm for the control of congested urban networks with intricate traffic patterns: New York City case studies. Technical report, Massachusetts Institute of Technology, 2015. Date accessed: 2021-08-11.

[153] C. Osorio and L. Chong. A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transportation Science*, 49(3):623–636, 2015.

[154] C. Osorio and G. Flötteröd. Capturing dependency among link boundaries in a stochastic dynamic network loading model. *Transportation Science*, 49(2):420–431, 2015.

[155] C. Osorio and C. Wang. 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.

[156] C. Osorio and J. Yamani. Analytical and scalable analysis of transient tandem Markovian finite capacity queueing networks. *Transportation Science*, 51(3):823–840, 2017.

[157] K. Ozbay and E. E. Ozguven. A comparative methodology for estimating the capacity of a freeway section. In *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference*, pages 1034–1039. IEEE, 2007.

[158] A. Pacheco, M. L. Simões, and P. Milheiro-Oliveira. Queues with server vacations as a model for pretimed signalized urban traffic. *Transportation Science*, 51(3):841–851, 2017.

[159] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003.

[160] A. Pell, A. Meingast, and O. Schauer. Trends in real-time traffic simulation. *Transportation Research Procedia*, 25:1477–1484, 2017.

[161] E. Perel, N. Perel, and U. Yechiali. A polling system with 'join the shortest-serve the longest' policy. *Computers and Operations Research*, 114:1–10, 2020.

[162] N. Petrovska and A. Stevanovic. Traffic congestion analysis visualisation tool. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1489–1494. IEEE, 2015.

[163] W. B. Powell. Analysis of vehicle holding and cancellation strategies in bulk arrival, bulk service queues. *Transportation Science*, 19(4):352–377, 1985.

[164] W. B. Powell and P. Humblet. The bulk service queue with a general control strategy: theoretical analysis and a new computational procedure. *Operations Research*, 34(2):267–275, 1986.

[165] A. Rafaeli, G. Barron, and K. Haber. The effects of queue structure on attitudes. *Journal of Service Research*, 5(2):125–139, 2002.

[166] M. I. Reiman and B. Simon. An interpolation approximation for queueing systems with Poisson input. *Operations Research*, 36(3):454–469, 1988.

[167] R. Remmert. *Theory of Complex Functions*, volume 122. Springer Science and Business Media, 1991.

[168] J. A. C. Resing. Polling systems and multitype branching processes. *Queueing Systems*, 13(4):409–426, 1993.

[169] rijnmond.nl. A16 weer vrij na ongeluk met vrachtwagens – Rijnmond. `https://www.rijnmond.nl/nieuws/164881/A16-weer-vrij-na-ongeluk-met-vrachtwagens`, 2018. Date accessed: 2021-04-12, in Dutch.

[170] J. Rios-Torres and A. A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, 2017.

[171] M. Roshani and I. Bargegol. Effect of pedestrians on the saturation flow rate of right turn movements at signalized intersection – case study from Rasht city. In *IOP Conference Series: Materials Science and Engineering*, volume 245, page 042032. IOP Publishing, 2017.

[172] N. M. Rouphail and B. S. Eads. Pedestrian impedance of turning-movement saturation flow rates: comparison of simulation, analytical, and field observations. *Transportation Research Record*, 1578(1):56–63, 1997.

[173] M. Saxena, I. Dimitriou, and S. Kapodistria. Analysis of the shortest relay queue policy in a cooperative random access network with collisions. *Queueing Systems*, 94(1-2):39–75, 2020.

[174] R. Schassberger. On the waiting time in the queuing system $GI/G/1$. *The Annals of Mathematical Statistics*, 41:182–187, 1970.

[175] P. J. Schweitzer. *A Survey of Aggregation-Disaggregation in Large Markov Chains*, pages 63–89. CRC Press, 1991.

[176] G. Shapira and H. Levy. On fairness in polling systems. *Annals of Operations Research*, pages 1–33, 2016.

[177] K. Sigman and W. Whitt. Heavy-traffic limits for nearly deterministic queues. *Journal of Applied Probability*, 48(3):657–678, 2011.

[178] K. Sigman and W. Whitt. Heavy-traffic limits for nearly deterministic queues: stationary distributions. *Queueing Systems*, 69(2):145–173, 2011.

[179] M. P. Singh and M. M. Srinivasan. Exact analysis of the state-dependent polling model. *Queueing Systems*, 41(4):371–399, 2002.

[180] S. Stidham Jr. $L = \lambda W$: a discounted analogue and a new proof. *Operations Research*, 20(6):1115–1126, 1972.

[181] C. Stutz and T. A. Runkler. Classification and prediction of road traffic using application-specific fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 10(3):297–308, 2002.

[182] P. M. Swamidass. *Encyclopedia of Production and Manufacturing Management*. Springer US, Boston, MA, 2000.

[183] D. Swaroop and J. K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.

[184] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti. Revisiting street intersections using slot-based systems. *PloS ONE*, 11(3):e0149607, 2016.

[185] Y. Takahashi, K. Fujimoto, and N. Makimoto. Geometric decay of the steady-state probabilities in a quasi-birth-and-death process with a countable number of phases. *Stochastic Models*, 17(1):1–24, 2001.

[186] J. Tan, H. Feng, X. Meng, and L. Zhang. Heavy-traffic analysis of cloud provisioning. In *2012 24th International Teletraffic Congress (ITC 24)*, pages 1–8. IEEE, 2012.

[187] Z. Z. Tian and N. Wu. Probabilistic model for signalized intersection capacity with a short right-turn lane. *Journal of Transportation Engineering*, 132(3):205–212, 2006.

[188] H. C. Tijms. *Stochastic Models: An Algorithmic Approach*, volume 303. Wiley New York, 1994.

[189] H. C. Tijms. *A First Course in Stochastic Models*. John Wiley and Sons, 2003.

[190] H. C. Tijms and M. C. T. van de Coevering. A simple numerical approach for infinite-state Markov chains. *Probability in the Engineering and Informational Sciences*, 5(3):285–295, 1991.

[191] H. C. Tijms and D. J. van Vuuren. Markov processes on a semi-infinite strip and the geometric tail algorithm. *Annals of Operations Research*, 113(1-4):133–140, 2002.

[192] Transportation Research Board. *Highway Capacity Manual 5th Edition HCM2010*. Transportation Research Board, Washington D.C., 2010.

[193] M. Treiber and A. Kesting. *Traffic Flow Dynamics: Data, Models and Simulation*. Springer Berlin Heidelberg, 2013.

[194] H. Tu. *Monitoring Travel Time Reliability on Freeways*. PhD thesis, Delft University of Technology, 2008.

[195] u0192. Wat was het weer? – Weerverleden.nl. `https://weerverleden.nl/`, 2021. Date accessed: 2021-04-12, in Dutch.

[196] M. van den Berg, J. Francke, M. de Haas, M. Hamersma, O. Huibregtse, O. Jonkeren, P. Jorritsma, M. Knoope, S. Moorman, F. Savelberg, J. Visser, and H. Wüst. Mobiliteitsbeeld 2019. Technical report, KiM Netherlands Institute for Transport Policy Analysis, 2019, in Dutch.

[197] M. S. van den Broek. Traffic signals: optimizing and analyzing traffic control systems. Master's thesis, Eindhoven University of Technology, 2004.

[198] M. S. van den Broek, J. S. H. van Leeuwaarden, I. J. B. F. Adan, and O. J. Boxma. Bounds and approximations for the fixed-cycle traffic-light queue. *Transportation Science*, 40(4):484–496, 2006.

[199] W. B. van den Hout. *The Power-Series Algorithm. A Numerical Approach to Markov Processes*. PhD thesis, Tilburg University, 1996.

[200] R. D. van der Mei. Towards a unifying theory on branching-type polling systems in heavy traffic. *Queueing Systems*, 57(1):29–46, 2007.

[201] N. M. van Dijk. Perturbation theory for unbounded Markov reward processes with applications to queueing. *Advances in Applied Probability*, 20(1):99–111, 1988.

[202] B. van Houdt. Numerical solution of polling systems for analyzing networks on chips. In *Proceedings of NSMC 2010*, pages 90 – 93, 2010.

[203] G. J. J. A. N. van Houtum. *New Approaches for Multi-Dimensional Queueing Systems*. PhD thesis, Eindhoven University of Technology, 1995.

[204] G. J. J. A. N. van Houtum, I. J. B. F. Adan, J. Wessels, and W. H. M. Zijm. The compensation approach for three or more dimensional random walks. In *DGOR/ÖGOR*, pages 342–349. Springer, 1993.

[205] G. J. J. A. N. van Houtum, W. H. M. Zijm, I. J. B. F. Adan, and J. Wessels. Bounds for performance characteristics: a systematic approach via cost structures. *Stochastic Models*, 14(1-2):205–224, 1998.

[206] J. S. H. van Leeuwaarden. Delay analysis for the fixed-cycle traffic-light queue. *Transportation Science*, 40(2):189–199, 2006.

[207] J. S. H. van Leeuwaarden, D. Denteneer, and J. A. C. Resing. A discrete-time queueing model with periodically scheduled arrival and departure slots. *Performance Evaluation*, 63(4-5):278–294, 2006.

[208] J. S. H. van Leeuwaarden, B. W. J. Mathijsen, and A. P. Zwart. Economies-of-scale in many-server queueing systems: tutorial and partial review of the QED Halfin-Whitt heavy-traffic regime. *SIAM Review*, 61(3):403–440, 2019.

[209] J. S. H. van Leeuwaarden and J. A. C. Resing. A tandem queue with coupled processors: computational issues. *Queueing Systems*, 51(1-2):29–52, 2005.

[210] P. van Mieghem. The asymptotic behavior of queueing systems: large deviations theory and dominant pole approximation. *Queueing Systems*, 23(1-4):27–55, 1996.

[211] M. van Vuuren and E. M. M. Winands. Iterative approximation of k-limited polling systems. *Queueing Systems*, 55(3):161–178, 2007.

[212] D. A. J. van Zwieten. *Fluid Flow Switching Servers: Control and Observer Design*. PhD thesis, Technische Universiteit Eindhoven, 2014.

[213] S. A. Vaqar and O. Basir. Traffic pattern detection in a partially deployed vehicular ad hoc network of vehicles. *IEEE Wireless Communications*, 16(6):40–46, 2009.

[214] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S-PLUS*. Springer Science and Business Media, 2013.

[215] V. M. Vishnevskii and O. V. Semenova. Mathematical methods to study the polling systems. *Automation and Remote Control*, 67(2):173–220, 2006.

[216] F. Viti. *The Dynamics and the Uncertainty of Delays at Signals*. PhD thesis, Delft University of Technology, 2006.

[217] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325–362, 1952.

[218] F. V. Webster. Traffic signal settings. Technical report, Road Research Board, 1958.

[219] L. M. Wein. Capacity allocation in generalized Jackson networks. *Operations Research Letters*, 8(3):143–146, 1989.

[220] E. M. M. Winands, I. J. B. F. Adan, and G. J. J. A. N. van Houtum. The stochastic economic lot scheduling problem: a survey. *European Journal of Operational Research*, 210(1):1–9, 2011.

[221] Wolfram Research, Inc., Mathematica, Version 12.2, Champaign, Ilinois, 2020.

[222] J. Wu, F. Yan, and A. Abbas-Turki. Mathematical proof of effectiveness of platoon-based traffic control at intersections. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 720–725. IEEE, 2013.

[223] N. Wu. Capacity of shared-short lanes at unsignalized intersections. *Transportation Research Part A: Policy and Practice*, 33(3-4):255–274, 1999.

[224] N. Wu. Modelling blockage probability and capacity of shared lanes at signalized intersections. *Procedia-Social and Behavioral Sciences*, 16:481–491, 2011.

[225] Q. Yang, Z. Shi, S. Yu, and J. Zhou. Analytical evaluation of the use of left-turn phasing for single left-turn lane only. *Transportation Research Part B: Methodological*, 111:266–303, 2018.

[226] R. Yao and H. Michael Zhang. Optimal allocation of lane space and green splits of isolated signalized intersections with short left-turn lanes. *Journal of Transportation Engineering*, 139(7):667–677, 2013.

[227] K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk. A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys (CSUR)*, 50(3):1–38, 2017.

[228] V. J. Yohai. High breakdown-point and high efficiency robust estimates for regression. *The Annals of Statistics*, 15(2):642–656, 1987.

[229] Y. Zhang and J. Tong. Modeling left-turn blockage and capacity at signalized intersection with short left-turn bay. *Transportation Research Record*, 2071(1):71–76, 2008.

[230] N. Zou. *Queuing Models and Analyses of Traffic Control*. PhD thesis, The University of Arizona, 2007.

List of publications

[231] M. A. A. Boon, A. J. E. M. Janssen, J. S. H. van Leeuwaarden, and R. W. Timmerman. Pollaczek contour integrals for the fixed-cycle traffic-light queue. *Queueing Systems*, 91(1-2):89–111, 2019.

[232] M. A. A. Boon, A. J. E. M. Janssen, J. S. H. van Leeuwaarden, and R. W. Timmerman. Optimal capacity allocation for heavy-traffic fixed-cycle traffic-light queues and intersections. *In preparation*, 2021.

[233] B. Klaasse, R. W. Timmerman, T. van Ballegooijen, M. A. A. Boon, and G. Eijkelenboom. A novel data-driven algorithm for the automated detection of unexpectedly high traffic flow in uncongested traffic states. In *European Workshop on Performance Engineering*, pages 65–83. Springer, 2019.

[234] R. W. Timmerman and M. A. A. Boon. New vehicle-actuated access algorithms for intersections close to oversaturation. In *2020 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 299–304. IEEE, 2020.

[235] R. W. Timmerman and M. A. A. Boon. A novel approximation scheme for multidimensional queueing models. *In preparation*, 2021.

[236] R. W. Timmerman and M. A. A. Boon. Platoon forming algorithms for intelligent street intersections. *Transportmetrica A: Transport Science*, 17(3):278–307, 2021.

[237] R. W. Timmerman and M. A. A. Boon. The fixed-cycle traffic-light queue with multiple lanes and temporary blockages. *In preparation*, 2021.

Summary

Congestion is a common phenomenon in road traffic. Despite all sorts of countermeasures, congestion remains an enormous societal problem, giving rise to e.g. large costs and a lower quality of living. Mitigating the negative effects of congestion is a hard task and therefore a substantial body of research has been devoted to the prevention and reduction of congestion. Common sources of congestion are intersections, where many vehicles have to share a common scarce resource. Traffic-light control might be used to reduce some of the congestion if the control is well-adapted to the various traffic streams leading to the intersection. However, a good traffic-light control is typically difficult to obtain, e.g. due to stochasticity in the arrival times of vehicles at the intersection. Motivated by these observations, we (among other things) deepen and extend the knowledge regarding traffic-light models that aim to represent such stochastic influences.

In Chapter 2, we study the so-called Fixed-Cycle Traffic-Light (FCTL) queue. We derive an alternative expression for the Probability Generating Function (PGF) of the steady-state queue-length distribution at the end of the green period for the FCTL model. This PGF enables us to derive a plethora of performance measures, such as the mean delay experienced by vehicles and the queue-length distribution at an arbitrary time. Our alternative expression for the PGF of the queue-length distribution at the end of the green period has several advantages compared to the commonly used expression. For example, for our alternative expression, there is no need to compute roots of a certain equation and to solve a set of equations, which both have been considered to be inevitable parts in the performance analysis of the FCTL queue.

In Chapter 3, we use the expression for the PGF of the queue-length distribution at the end of the green period for the FCTL queue that we derived in Chapter 2 to obtain a Halfin-Whitt type of heavy-traffic scaling. Furthermore, this

heavy-traffic scaling leads to approximations for, e.g., the mean queue length at the end of the green period, which turns out to be accurate in many circumstances. We leverage these approximations to obtain traffic-light settings that are close to the optimal settings, that are easy to compute, and that allow for an intuitive interpretation.

In Chapter 4, we study the same type of heavy-traffic scaling as in Chapter 3, but apply it to different traffic-light control strategies. Instead of the fixed settings which are studied in the FCTL queue, we allow the green periods to end early, e.g. when the queue in front of the traffic light is dissolved. We leverage several simulation techniques to demonstrate that similar behavior can be observed as for the FCTL queue.

We also study an extension of the FCTL queue in Chapter 5. Here, vehicles may be blocked, e.g. because of pedestrians that block turning vehicles when both pedestrians and turning vehicles receive a green light at the same time. We derive the PGF of the steady-state queue-length distribution at the end of the green period for this extension of the FCTL queue and study several performance measures. We illustrate the impact of such pedestrian crossings. The extension that we formulated also allows us to model vehicle streams which are spread over several lanes. In particular, we can also study the FCTL queue with several lanes and allow for blockages (if desired).

In Chapter 6, we present another contribution of this thesis, which is an approximation scheme that allows us to obtain performance measures for several queueing models for which analytical results are scarce. Some of those models directly relate to several classical traffic-light control strategies. The key step taken in the scheme is an approximation for several unknown functions in the PGF of the joint queue-length distribution for the queueing model at hand. By using polynomials and roots of a certain equation, we are able to approximate the unknown functions, which leads to an approximation of the joint queue-length PGF. From this PGF various performance measures can be derived. The scheme may be leveraged to study e.g. models for vehicle-actuated traffic lights.

In Chapter 7, we turn our focus to autonomous vehicles, which are expected to occupy the road in the near future. We demonstrate how significant performance improvements can be obtained when certain criteria are met. We investigate how vehicles should approach the intersection and what the effect of platoon forming of arriving vehicles is on several performance measures. We obtain closed-form expressions for the trajectories that vehicles should drive such that they either minimize spillback effects or minimize the amount of acceleration (leading to energy-efficient approaches). Moreover, we demonstrate the benefit of platoon forming for autonomous vehicles compared to current-

day traffic.

Finally, in Chapter 8, we also study congestion at highways. We develop a data-driven algorithm that enables us to automatically identify so-called high-performance days, days with both a high traffic flow and no congestion. The algorithm employs historic loop-detector data, common concepts in traffic engineering, such as the fundamental diagram, and robust regression. The resulting high-performance days can be investigated further to get a sense for underlying factors that cause a day to be a high-performance day. Ultimately, this might lead to countermeasures against congestion on highways.

About the author

Rik Timmerman was born in Drunen, the Netherlands, on August 7, 1994. After completing his secondary education at d'Oultremontcollege in Drunen in 2012, he studied Applied Mathematics at Eindhoven University of Technology. In 2015, he received his Bachelor's degree with highest honors (cum laude). Afterwards, he pursued a Master's degree in Industrial and Applied Mathematics at Eindhoven University of Technology and obtained his degree with highest honors (cum laude) in 2017.

On September 1, 2017, he started his PhD research project at the Department of Mathematics and Computer Science of Eindhoven University of Technology under the supervision of Johan van Leeuwaarden, Ivo Adan, and Marko Boon. Rik's research has primarily focused on queueing-theoretic models for road traffic. The results of this research are presented in this dissertation.

During his PhD project, Rik was involved in teaching several courses, both as instructor and lecturer. He has been a member of the Department Council of the Department of Mathematics and Computer Science for two years. He served as a referee for several scientific journals and conferences. Further, he co-organized the hybrid workshop "Road Traffic Flow: Analysis, Optimization and Control" hosted at EURANDOM in 2021. Besides attending several workshops and seminars at EURANDOM, he also visited and presented his work at several national and international conferences, such as ECQT 2018 (Jerusalem), MATTS 2018 (Delft), INFORMS APS 2019 (Brisbane), YEQT XIII (Eindhoven), EPEW 2019 (Milan), IEEE ITSC 2020 (Rhodes), EURO 2021 (Athens), and the Beta Symposium 2021 (Soesterberg).

Rik will defend his thesis on January 28, 2022.