Networks of Fixed-Cycle Traffic-Lights

Klaas Maes
0822469
February 23, 2015
Networks of Fixed-Cycle Traffic-Lights

Author: K.J. Maes
0822469

Supervisor: dr. ir. M.A.A. Boon

A thesis submitted in partial fulfillment of the requirements of the degree of

Master of Science

February 23, 2015
Summary

In this thesis we generalized existing models to analyze the Fixed-Cycle Traffic-Light queue, such that we can calculate the probability generating functions of the queue length throughout the cycle and the delay of an arbitrary arriving vehicle. We did this in such a way that we are now able to manage having multiple green periods in one cycle, handle with different arrival processes during the cycle and to take into account that the first vehicle to depart after the traffic light turns green needs some more time to depart.

Furthermore we constructed techniques that allow us to use the model to analyze not only isolated intersections, but also networks of intersections. We can now find the output process of an intersection, such that this process can be used in some way as the arrival process of the next intersection and we can calculate the probability that a queue gets larger than a given threshold length, which would mean that the previous intersection would get blocked and traffic would be jammed. We can as well find the probability generating function of the delay of an arbitrary vehicle throughout a network of intersection. Finally one of the generalizations in the model makes it possible to take into account different number of lanes for a traffic lights in the different intersections of a network.

All these new possibilities come with a numerical downside. To do all this one needs to find some roots of an equation, which might be time consuming in some cases. To analyze the probability generating functions it might be useful to extract the probabilities from the probability generating functions and for some cases these probability functions become quite complex, which implies that extracting these probabilities is sometimes a hard problem to solve.

Besides from these new possibilities, we showed that when we construct a network of identical intersections, possibly shifted, under certain conditions the output processes will become Bernoulli processes and most of the time converge to certain Bernoulli Processes.
Acknowledgments

During this nine months master's project of my study Industrial and Applied Mathematics at the Eindhoven University of Technology i have learned many new things. I have some people to thank for giving me the opportunity to learn these things and all their help during the project. First of all i need to thank my supervisor Marko Boon, he was always available for guidance, gave me new ideas and the opportunity to chase the directions that interested me the most. He inspired me with his enthusiasm for the subject and encouraged me to keep pushing myself for better results. Furthermore i want to thank him for reading my thesis multiple times and giving me advice in how to improve it.

Onno Boxma and Johan van Leeuwaarden were always prepared to share their thoughts and by doing that they gave me many ideas to continue with, for which i want to thank them as well. Furthermore i would like to thank the complete assessment committee, consisting of Marko Boon, Johan van Leeuwaarden and Adrian Muntean, for reading my thesis and judging its value. Finally i want to thank all my friends and family for the ideas that they gave me without realizing it themselves.

Klaas Maes
February 2015
# Table of contents

**Title**
Networks of Fixed-Cycle Traffic-Lights

**Summary** ii

**Acknowledgments** iii

**Table of contents** iv

1 **Introduction** 1

2 **Literature Review** 2

3 **Model Description** 5
   3.1 First Generalization: Green Variable . . . . . . . . . . . . . . . 5
   3.2 Second Generalization: Arrival Process per Time slot . . . . . 5
   3.3 Third Generalization: Departure Variable . . . . . . . . . . . . . 6
   3.4 Assumptions . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

4 **Queue Length Analysis** 8
   4.1 Probability Generating Function of the Queue Length . . . . . . 8
   4.2 Expected Queue Length . . . . . . . . . . . . . . . . . . . . . . 11

5 **Delay Analysis** 13
   5.1 Probability Generating Function of the Delay . . . . . . . . . . 13
   5.2 Expected Delay . . . . . . . . . . . . . . . . . . . . . . . . . . 15

6 **Comparison of the Models** 17
   6.1 Comparison Expected Overflow Queue . . . . . . . . . . . . . . 17
   6.2 Comparison Expected Delay . . . . . . . . . . . . . . . . . . . . 18

7 **Simple Examples** 19
   7.1 No Departure in a Green Time Slot . . . . . . . . . . . . . . . 19
   7.2 Multiple Green Periods in a Cycle . . . . . . . . . . . . . . . 20
   7.3 Road Construction Example: Optimizing Signal Settings . . . . 23
# Table of contents

- **Title**
  - Networks of Fixed-Cycle Traffic-Lights

- **8 Networks of Intersections**
  - 8.1 Tail Probability for Transient and Steady State
    - 8.1.1 Example
  - 8.2 Output Process
  - 8.3 Delay in a Network
  - 8.4 Network of identical intersections
    - 8.4.1 Cycle length of three
    - 8.4.2 Arbitrary cycle length
  - 8.5 Simple Example of a Network
  - 8.6 Realistic Example of a Network

- **9 Numerical Issues**
  - 9.1 Root Finding
    - 9.1.1 Exact Solution Method
    - 9.1.2 Fixed Point Method
    - 9.1.3 Fourier Series Method
    - 9.1.4 FindRoot in Mathematica
    - 9.1.5 Root Finding Examples
  - 9.2 System of Equations
  - 9.3 Inversion of a Probability Generating Function

- **10 Conclusions and Suggestions for Further Research**

- **References**
1 Introduction

In 1868 the first traffic lights were installed at an intersection in London, this to guarantee a safe crossing for the military police. After that more traffic lights followed. In the Netherlands the first traffic lights were installed in the 30s in the 20th century. It is not completely sure which city was the first, but there are claims for Amsterdam, The Hague and Eindhoven. Nowadays there are about 5500 intersections with traffic lights in the Netherlands. Many of them make use of dynamic systems which detect waiting vehicles in the different lanes, but also still quite a few make use of fixed cycle settings. Every day many people get irritated in traffic, because of the fact that they have to wait in front of such a traffic light. And already much research is spent to try to set these traffic lights as optimal as possible. A main goal in these optimizations is most of the time to minimize the delay of an arbitrary vehicle.

The fixed-cycle traffic-light queue (FCTL), the model where the traffic light signal alternates between green and red periods of some fixed durations, is well studied. In most of these studies it is assumed that the vehicles arrive at an intersection according to a Poisson process. We generalize existing models in multiple ways. A first generalization is that we allow for different arrival processes in each time slot. Furthermore we allow multiple green periods in one cycle, by deciding for each slot if the traffic light is green or red. We also allow the possibility that no vehicle can depart during a green slot, making it possible to model the possibility that the first vehicles need more time to leave the system.

In this thesis we explore the possibilities to extend existing models, in particular the model of Van Leeuwaarden [9], such that we can use them to analyze more general fixed-cycle traffic-light intersections. Furthermore we construct mathematical techniques that can be used together with our model to analyze networks of intersections.

In Chapter 2 we take a look at existing research and we give an overview of relevant results. We describe our model in Chapter 3 what are the differences with the model of Van Leeuwaarden [9] and discuss our assumptions. After that we derive the distributions for the queue length and the delay in respectively Chapter 4 and 5. In Chapter 6 we compare our model with the models described in Chapter 2 and in Chapter 7 we show some simple examples that make use of our generalizations. After that in Chapter 8 we construct some techniques that make it possible to use our model to analyze networks of intersections and we use these techniques to analyze what happens if we consider a network of multiple intersections with the same settings. In Chapter 9 we take a look at the numerical difficulties of our model and show some methods that can tackle these in some situations. We finish this thesis with some conclusions and recommendations for further research.
2 Literature Review

In this chapter we briefly describe some relevant results of earlier research on the queue length and delay at traffic lights. We work towards a model by Van Leeuwaarden [9], which is the starting point for our more generalized model. All the models in the earlier research make use of a fixed cycle of length $C$. Furthermore this cycle is split into a green period of length $G$ and a red period of length $R$. In real life most traffic lights have an amber period as well. These models do not consider this, since one could easily see the amber period as green or red period or one could divide it over the two. In Figure 2.1 such a cycle is illustrated and also an indication of how the expected queue length looks over time is given. This expected queue length is obtained using simulation. It has been a challenge to obtain a mathematical expression for this.

\begin{equation}
\text{Expected Queue Length}
\end{equation}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{queue_length.png}
\caption{Expected Queue Length over time.}
\end{figure}

Much research has been done to analyze the overflow queue, the queue at the end of the green period, and the delay of a vehicle at a traffic light. Webster [10] gives a formula for the mean delay of a vehicle in closed form, in this expression the third term is based on simulation.

\begin{equation}
\mathbb{E} D = \frac{C}{2} \left( 1 - \frac{C}{E_q} \right)^2 + \frac{q \left( \frac{C}{G} \right)^2}{2(1 - \frac{C^2}{G^2})} - 0.65 \left( \frac{C}{q^2} \right)^2 \left( \frac{Cq}{G^3} \right)^{2+\frac{q^2}{G^2}}.
\end{equation}

In this equation $q$ is the flow of arriving vehicles per second and $s$ the saturation flow, which can be considered as the maximum capacity of a given flow.

McNeill [7] provides an exact expression for the expected delay up to one unknown: the mean length of the overflow queue (the mean stationary queue length at the end of a green period $\mathbb{E} X_g$). The formula holds for compound Poisson arrivals.

\begin{equation}
\mathbb{E} D = \frac{R}{2C(1 - \frac{1}{\mu})} \left( R + 2\lambda^{-1} \mathbb{E} X_g + \mu^{-1} \left( 1 + I \left( 1 - \frac{\lambda}{\mu} \right)^{-1} \right) \right).
\end{equation}

In this formula $\lambda$ is the arrival rate, $\mu^{-1}$ is the constant length of the service periods and $I$ is the dispersion index for the arrival process, the variance of the number of arrivals in an interval.
divided by the expected number of arrivals in that interval.

Most of the research is focused on deriving formulas for the mean queue length and delay. Meissl [8] and Darroch [3] independently derived the probability generating function (pgf $X_g(z)$) of the stationary overflow queue. Darroch [3] used a discrete-time setting and his approach is analyti-
cally and computationally involved. The cycle time $C$ is split into $c$ time slots of the same size, $g$ is the number of contiguous time slots for which the traffic light is green.

$$X_g(z) = \frac{Y^g(z)\left(\frac{\theta(z)}{1-\alpha+\alpha z}Y(z) - 1\right)}{z^g - (1 - \alpha + \alpha z)Y(z)} \sum_{k=0}^{g-1} \frac{q_kz^k}{(1-\alpha+\alpha z)^g Y^c(z)}.$$  

In this equation $Y(z)$ is the pgf of the arrival process in every time slot, $\alpha$ is the probability that a vehicle that can depart the queue due to a green light, still needs to wait for a vehicle from an conflicting oncoming traffic stream. $\theta(z)$ is the pgf of the queue length at the end of a time slot, given that at the end of the previous time slot there are no vehicles in the queue. To calculate the probability generating function of the overflow queue one needs to find $g$ roots ($q_k$) of some equation. In those times it was not that easy to find those roots, which may have been the reason why there has not been that much research on the probability generating function of the overflow queue after this. Nowadays with the more powerful computers it is significantly easier to find these roots. Darroch however constructed formulas to calculate a lower and upper bound for the expected length of the overflow queue and the delay. For this he mainly uses an lower and upper bound on $\sum_{k=1}^{g-1} kq_k$ stated below in Equation (2.1).

$$\frac{1}{2}(g + 1)K - \frac{1}{2}p_0'[(g - 1)(\theta(0) + rp_1(1 - \lambda)) + g + 1] < \sum_{k=1}^{g-1} kq_k < \frac{1}{2}K'\{}(2g - K' - 1) + (g - K' - 1)(K - K')\},$$  

with $K = \frac{g(1-\lambda)1-(\alpha-L)}{1-\overline{\alpha}}$, $K'$ the integer part of $K$, $p_0 = \mathbb{P}(Y = 0)$ and $p_1 = \mathbb{P}(Y = 1)$.

Using this bound together with Equations (2.2) and (2.3), one can find a lower and upper bound for the expected length of the overflow queue and the expected delay.

$$\mathbb{E}X_g = (1 - \overline{\alpha} - \lambda)(1 - \overline{\alpha} - \overline{\lambda} + \overline{\theta}'(1))\sum_{k=1}^{g-1} kq_k + B,$$  

$$\mathbb{E}D = \frac{r(1-\lambda)}{(g + r)\alpha} \mathbb{E}X_g + \frac{B}{(g + r)\alpha},$$  

with

$$A = \frac{1}{2}g(g - 1)(1 - \overline{\alpha} - \overline{\lambda}) + \frac{1}{2}r(r + 1)\alpha,$$

$$B = \frac{1}{2}\left[\theta''(1) + 2\theta'(1)(1 - \overline{\alpha} - \overline{\lambda} - 2(\overline{\alpha} + \overline{\lambda})(1 - \overline{\alpha} - \overline{\lambda}) - 2\lambda\alpha - Y''(1)\right]K$$

$$- \frac{1}{2}
\left[\theta''(1)(1 - \overline{\lambda} - 2\overline{\alpha}) - g(1 + \lambda)^2 + \theta'(1)(1 - \overline{\lambda} - 2\overline{\alpha}) - g(1 + \overline{\lambda} - 2\overline{\alpha}) + (g + r)(\overline{\alpha} - Y(1)).\right]$$

Van Leeuwaarden [9] showed that the result of Darroch [3] holds for more general discrete arrival processes, he showed how to derive the pgf of the queue length throughout the whole cycle and derived not only the pgf of the overflow queue, but as well of the stationary delay. His results use

3 Networks of Fixed-Cycle Traffic-Lights
\[ \alpha = 0, \text{ which implies that when the traffic light is green a vehicle can depart and will not conflict with an oncoming traffic stream. The results are summarized in (2.4)-(2.9) below.} \]

\[ X_g(z) = \frac{Y_g(z) \sum_{k=0}^{g-1} q_k z^k}{z^g - Y_c(z)}, \quad (2.4) \]

\[ D(z) = \frac{1}{c} \sum_{m=1}^{c} D_{[m]}(z), \quad (2.5) \]

with

\[ D_{[m]}(z) = \frac{z^{(c-m+1)g}}{g} \sum_{i=0}^{e-1} U_{[m]}^r(a^i z^i) \frac{1 - (z^{-r}a^{-i})^g}{1 - z^{-r}a^{-i}}, \quad (2.6) \]

\[ U_{[m]}^r(z) = X_g(z)Y(z)^m - Y(z) \frac{1 - Y(z)}{(1 - z)E_Y}, \quad (2.7) \]

for \( m = g + 1, ..., c \) (the red time slots) and the following for \( m = 1, ..., g \) (the green time slots),

\[ D_{[m]}(z) = q_{m-1} + (1 - q_{m-1}) \frac{z^{(c-m+1)g}}{g} \sum_{i=0}^{e-1} U_{[m]}^g(a^i z^i) \frac{1 - (z^{-r}a^{-i})^g}{1 - z^{-r}a^{-i}}, \quad (2.8) \]

\[ U_{[m]}^g(z) = \left( \frac{X_{m-1}(z) - q_{m-1}}{1 - q_{m-1}} \right) \frac{1 - Y(z)}{(1 - z)E_Y} z^{m-1}, \quad (2.9) \]

and \( a = e^{2\pi i / c} \).

The results in Van Leeuwaarden [9] form the starting point of this research thesis.
3 Model Description

After studying earlier research on the Fixed-Cycle Traffic-Light queue in Chapter 2, we use the model by Van Leeuwaarden [9] to construct a more general model. In his model, he uses time slots of the same size, the first \( g \) are green time slots and the other \( (c - g) \) in a cycle are red time slots. In our model, we make extensions, such that we can model more general real life traffic light systems. Our extensions provide solutions to possible practical issues in particular when applying the model in a network setting.

1. It might be efficient to have multiple green periods of different length in one cycle.
2. In a network, the arrival process might be identical for each cycle, but exhibits variation within a cycle.
3. It might be the case that when the traffic light turns green, the first vehicle needs more time to depart the queue than those behind him.
4. It might be desirable to have a way to model different number of lanes for different traffic lights in a network.

In our model, we again use a cycle with \( c \) time slots of the same size, but due to the first generalization, we will not have that we have first \( g \) green time slots followed by \( c - g \) red time slots.

3.1 First Generalization: Green Variable

First, we introduce a variable \( g_i \) which states if the traffic light is green in the \( i \)-th time slot \( (g_i = 1) \) or red \( (g_i = 0), \ i = 1, 2, ..., c \). By setting the first \( g \) time slots to 1 and the others to 0, we cover all the possibilities in the model of Van Leeuwaarden (see Figure 3.1a). This extension allows us to model various more complicated situations that arise in practice. For example, we can model traffic lights with multiple green periods in a cycle (see Figure 3.1b). This makes it possible to model more realistic examples. In real life, networks of fixed-cycle traffic lights are connected in a network, you can keep working in the same defined cycle, but with other green time slots. For consistency with the model of Van Leeuwaarden, we define \( g = \sum_{k=1}^{c} g_k \) as the total number of time slots in a cycle in which the traffic light is green.

3.2 Second Generalization: Arrival Process per Time slot

A second generalization is that we allow for different arrival processes in every time slot in the cycle (see Figure 3.2). This makes it possible to model more realistic examples. In real life...
the arrival process may for instance depend on the previous intersections. In Section 8.2 we construct the output process of an intersection, this process can be used to construct the arrival process for the next intersection.

<table>
<thead>
<tr>
<th>$g_i$</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

(a) Equivalent with Van Leeuwaarden's model

<table>
<thead>
<tr>
<th>$g_i$</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

(b) Possibility with multiple green periods of different size

Figure 3.1: First Generalization: Green Variable

3.3 Third Generalization: Departure Variable

A third generalization is that we introduce a variable $d_i$ which determines whether a vehicle can leave the queue when a traffic light is green. Again $d_i$ equals 1 if a vehicle can leave the queue, 0 otherwise. This can be used to model a situation where the first vehicle leaving the queue takes more time than the other ones (see Figure 3.2b). Another possible application for this generalization is to model multiple lanes. For instance if we have two connected intersection where we have three lanes in the first intersection and this for the direction towards the next intersection and that next intersection has four lanes in the direction we want to analyze, one could set three of every four $d_i$'s to one and one to zero, to model the fact that in the time that four vehicles depart in the second intersection only three can depart from the first intersection. Furthermore one could use these $d_i$'s to model lanes for which it might be possible that vehicles can be blocked by crossing lanes and only a certain partial of the time slots a vehicle can leave the queue. Let $d = \sum_{i=1}^{c} d_i$ the total number of slots in a cycle in which a vehicle can depart from the queue. Obviously, a practical restriction is that $d_i$ cannot be 1 if $g_i = 0$, because no vehicle is allowed to depart, when the traffic light is red.

<table>
<thead>
<tr>
<th>$d_i$</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

(a) Equivalent with the model in Figure 3.2

<table>
<thead>
<tr>
<th>$d_i$</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

(b) Possibility that the first vehicle needs more time to depart

Figure 3.3: Third Generalization: Departure Variable

6 Networks of Fixed-Cycle Traffic-Lights
3.4 Assumptions

Darroch [3] and Van Leeuwaarden [9] made three assumptions for their models. We generalized these models, requiring slightly different (or new) assumptions. We state the assumptions below and explain the differences with Darroch [3] and Van Leeuwaarden [9].

**Assumption 1.** The time axis is divided into constant time intervals of unit length, so-called slots, where in each slot at most one delayed vehicle can depart from the queue. The cycle time is assumed to be a fixed integer multiple $c$ of one slot. Those vehicles that arrive at the queue and are delayed join the queue at the end of the slot in which they arrive.

The difference with Darroch [3] and Van Leeuwaarden [9] is that they assumed that in each slot exactly one delayed vehicle can depart from the queue. In our model it is possible that $g_i = 1$ and $d_i = 0$, the traffic light is green, but no vehicle will depart the queue. This assumption also makes sure that all the time slots are of the same size and together are exactly one cycle. Furthermore this assumption states that during one time slot at most one vehicle can leave the queue.

**Assumption 2.** Let $Y_{k,n}$ denote the number of vehicles that arrive at the intersection during slot $k$ of cycle $n$. The random variables $Y_{k,n}$ are assumed to be discrete and i.i.d. for all $k, n$ with pgf $Y_k(z)$.

This assumption states that all the arrival processes are distributed like some set of random variables, more specific that they do depend on the time slot, but not on the current cycle. Darroch [3] and Van Leeuwaarden [9] assumed that all arrival processes have the same distribution $Y$, it does not matter which time slot.

**Assumption 3.** For those time slots in which the queue is empty and the traffic light is green, all vehicles that arrive during this slot pass through the system and experience no delay.

This assumption makes sure that all vehicles that find a green traffic light and no waiting vehicles in front of him, will not experience a delay and can pass through the system immediately. Darroch [9] and Van Leeuwaarden [9] used exactly the same assumption.

**Assumption 4.** In every last time slot of a green period a vehicle can depart the queue. So $g_i = 1$ and $g_{i+1} = 0 \Rightarrow d_i = 1 \forall i \in \{1, 2, ..., c - 1\}$ and also $g_c = 1$ and $g_1 = 0 \Rightarrow d_c = 1$.

This assumption is required to assure computability of our model. Furthermore it is a realistic assumption, because if in the last time slot of a green period no vehicle can leave the queue, it would be advisable to make that last slot red instead of green.
4 Queue Length Analysis

In this chapter we derive the probability generating function of the queue length and a formula for the expected queue length in every time slot in steady state. The system will reach a steady state when the expected number of arrivals in a cycle is less than the maximal number of departures in a cycle ($\sum_{m=1}^{c} EY_m < d$, where $Y_m$ denotes the number of arrivals in the $m$-th time slot).

4.1 Probability Generating Function of the Queue Length

Let $X_{k,n}$ denote the queue length at time $k$ in cycle $n$. Then the following recursive relation holds.

\[
X_{k+1,n} = \begin{cases} 
X_{k,n} + Y_{k+1,n} - 1 & \text{if } X_{k,n} \geq 1 \text{ and } d_{k+1} = 1, \\
0 & \text{if } X_{k,n} = 0 \text{ and } g_{k+1} = 1, \\
X_{k,n} + Y_{k+1,n} & \text{otherwise.} 
\end{cases} \tag{4.1}
\]

For all slots where $d_{k+1} = 1$ we have the following probabilities for $X_{k+1,n}$.

\[
\mathbb{P}(X_{k+1,n} = j) = \sum_{p=1}^{j+1} \mathbb{P}(X_{k,n} = p) \mathbb{P}(Y_{k+1,n} = j - p) \text{ if } j = 1, 2, ..., \tag{4.2}
\]

\[
\mathbb{P}(X_{k+1,n} = 0) = \mathbb{P}(X_{k,n} = 0) + \mathbb{P}(X_{k,n} = 1) \mathbb{P}(Y_{k+1,n} = 0). \tag{4.3}
\]

Let $X_{k+1,n}(z)$ denote the probability generating function (pgf) of $X_{k+1,n}$ it then follows from (4.2) and (4.3) that

\[
X_{k+1,n}(z) = \sum_{j=0}^{\infty} \mathbb{P}(X_{k+1,n} = j) z^j = z^{-1} Y_{k+1,n}(z) X_{k,n}(z) + \mathbb{P}(X_{k,n} = 0) \left( 1 - z^{-1} Y_{k+1,n}(z) \right). 
\]

We do the same for all slots where $d_{k+1} = 0$ and $g_{k+1} = 1$, indicating that the traffic light is green, but no vehicles can depart from the queue.

\[
\mathbb{P}(X_{k+1,n} = j) = \sum_{p=1}^{j} \mathbb{P}(X_{k,n} = p) \mathbb{P}(Y_{k+1,n} = j - p) \text{ if } j = 1, 2, ..., \\
\mathbb{P}(X_{k+1,n} = 0) = \mathbb{P}(X_{k,n} = 0).
\]

In this case the pgf $X_{k+1,n}(z)$ of $X_{k+1,n}$ is given by

\[
X_{k+1,n}(z) = Y_{k+1,n}(z) X_{k,n}(z) + \mathbb{P}(X_{k,n} = 0) \left( 1 - Y_{k+1,n}(z) \right).
\]
Finally we consider the case where $g_{k+1} = 0$, which implies that $d_{k+1} = 0$.

$$
P(X_{k+1,n} = j) = \sum_{p=0}^{j} P(X_{k,n} = p) P(Y_{k+1,n} = j - p) \text{ if } j = 0, 1, 2, \ldots,
$$

$$X_{k+1,n}(z) = X_{k,n}(z) Y_{k+1,n}(z).$$

We now define

$$\zeta_{k+1,n} = z^{-d_{k+1}} Y_{k+1,n}(z),$$

which enables us to rewrite $X_{k+1,n}(z)$,

$$X_{k+1,n}(z) = \zeta_{k+1,n}(z) X_{k,n}(z) + (1 - \zeta_{k+1,n}(z)) P(X_{k,n} = 0) g_{k+1}. \quad (4.4)$$

**Theorem 1.** Under the conditions stated in Chapter 3 and 4, we have that the probability generating function of the queue length in the k-th time slot in the n-th cycle is given by:

$$X_{k,n}(z) = X_{0,n}(z) \prod_{i=0}^{k-1} \zeta_{i+1,n}(z) + \sum_{j=0}^{k-1} \prod_{i=j+1}^{k-1} \zeta_{i+1,n}(z) (1 - \zeta_{j+1,n}(z)) P(X_{j,n} = 0) g_{j+1}. \quad (4.5)$$

**Proof.** We will proof this by induction. For $k = 1$ we have:

$$X_{1,n}(z) = X_{0,n}(z) \zeta_{1,n}(z) + (1 - \zeta_{1,n}(z)) P(X_{0,n} = 0) g_{1} \quad (4.6)$$

Suppose that it holds for $k$. We will show that it holds for $k + 1$ as well. So we have

$$X_{k,n}(z) = X_{0,n}(z) \prod_{i=0}^{k-1} \zeta_{i+1,n}(z) + \sum_{j=0}^{k-1} \prod_{i=j+1}^{k-1} \zeta_{i+1,n}(z) (1 - \zeta_{j+1,n}(z)) P(X_{j,n} = 0) g_{j+1} \quad (4.7)$$

and also

$$X_{k+1,n}(z) = \zeta_{k+1,n}(z) X_{k,n}(z) + (1 - \zeta_{k+1,n}(z)) P(X_{k,n} = 0) g_{k+1} \quad (4.8)$$

Combining these 2 equations gives:

$$X_{k+1,n}(z) = \zeta_{k+1,n}(z) \left( X_{0,n}(z) \prod_{i=0}^{k-1} \zeta_{i+1,n}(z) + \sum_{j=0}^{k-1} \prod_{i=j+1}^{k-1} \zeta_{i+1,n}(z) (1 - \zeta_{j+1,n}(z)) P(X_{j,n} = 0) g_{j+1} \right) + (1 - \zeta_{k+1,n}(z)) P(X_{k,n} = 0) g_{k+1} \quad (4.9)$$

$$= X_{0,n}(z) \prod_{i=0}^{k} \zeta_{i+1,n}(z) + \sum_{j=0}^{k} \prod_{i=j+1}^{k} \zeta_{i+1,n}(z) (1 - \zeta_{j+1,n}(z)) P(X_{j,n} = 0) g_{j+1} + (1 - \zeta_{k+1,n}(z)) P(X_{k,n} = 0) g_{k+1} \quad (4.10)$$

$$= X_{0,n}(z) \prod_{i=0}^{k} \zeta_{i+1,n}(z) + \sum_{j=0}^{k} \prod_{i=j+1}^{k} \zeta_{i+1,n}(z) (1 - \zeta_{j+1,n}(z)) P(X_{j,n} = 0) g_{j+1} \quad (4.11)$$

Note that the queue length at the end of the n-th cycle is the same as the queue length at the beginning of the $(n+1)$-th cycle $(X_{n,n} = X_{0,n+1})$. In equilibrium $X_{0,n} = X_{0,n+1}$. Denote by $X_k$ the stationary distribution of $X_{k,n}$ with corresponding pgf $X_k(z)$.

$$X_0(z) = X_0(z) \prod_{i=0}^{c-1} \zeta_{i+1}(z) + \sum_{j=0}^{c-1} \prod_{i=j+1}^{c-1} \zeta_{i+1}(z) (1 - \zeta_{j+1}(z)) q_j g_{j+1}.$$
After some rewriting we obtain the following formula for the pgf of $X_0$.

$$X_0(z) = \frac{z^d \sum_{j=0}^{c-1} q_j s_{j+1}(1 - \zeta_{j+1}(z)) \prod_{i=0}^{c-1} \zeta_{i+1}(z)}{z^d - \prod_{i=0}^{c-1} Y_{i+1}(z)},$$  \hspace{1cm} (4.12)

where $q_k = \mathbb{P}(X_k = 0)$. We used the fact that $\prod_{i=0}^{c-1} \zeta_{i+1}(z) = \prod_{i=0}^{c-1} Y_{i+1}(z)$. This expression still contains $g$ unknowns $\{q_k: g_k = 1\}$. We now state a Lemma that will give us conditions such that the denominator of (4.12) has $d$ roots on or within the unit circle $|z| = 1$.

**Lemma 1.** Let $\Psi(z)$ denote a probability generating function for which $\Psi(z)$ is analytic in $|z| < 1 + \Delta$ for some $\Delta > 0$. Then provided that $\Psi'(z) < d$ the equation

$$z^d - \Psi(z) = 0$$  \hspace{1cm} (4.13)

has $d$ distinct roots, $z_0, z_1, \ldots, z_{d-1}$, within the unit circle, where $z_0 = 1$.

**Proof.** We have that $\Psi(z)$ is analytic in $|z| < 1 + \Delta$ for some $\Delta > 0$ and that $\Psi'(z) < d$. For some $\delta > 0$ small enough, we have $\Psi(1 + \delta) < \Psi(1) + d\delta = 1 + d\delta \leq (1 + \delta)^d$. Furthermore we have, because of the fact that $\Psi(z)$ is a pgf and using the triangle inequality, that

$$|\Psi(z)| = |\sum_{k=0}^{\infty} p_k z^k| \leq \sum_{k=0}^{\infty} p_k |z|^k = \Psi(|z|).$$

For $|z| = 1 + \delta$ we have that $|z|^d > \Psi(|z|) \geq |\Psi(z)|$ or equivalently $|z| > |\Psi(z)|^{\frac{1}{d}}$. By Rouché’s Theorem it follows that the equation

$$z - e^{2\pi i k/d} \Psi(z)^{\frac{1}{d}} = 0$$  \hspace{1cm} (4.14)

has exactly one root $z_k$ in $|z| < 1 + \delta$, $k = 0, 1, \ldots, d - 1$. Clearly $z_0 = 1$ and furthermore $z_k \neq z_{k'}$ for $k \neq k'$ since $e^{2\pi i k/d} \neq e^{2\pi i k'/d}$. Finally we notice that the roots of (4.13) are the roots of the $d$ equations in (4.14). \hfill \Box

Using this Lemma with $\Psi(z) = \prod_{i=0}^{c-1} Y_{i+1}(z)$, we find that the denominator of (4.12) has $d$ roots on or within the unit circle $|z| = 1$. Necessary conditions for this are that $Y_{i+1}(z)$ is analytic in $|z| < 1 + \Delta$ for every $i$ in $\{0, 1, \ldots, c - 1\}$ and $\sum_{j=0}^{c-1} E Y_j < d$, the stability condition. The numerator of (4.12) should vanish at each of the roots. This gives $d$ equations. One of the roots equals 1, and yields a trivial equation. However, the normalization condition $X_0(1) = 1$ provides an additional equation. This system has $d$ equations with $d$ unknowns $\{q_k: g_k = 1\}$.

We still need to find $g - d$ other unknowns $q_k$’s. We can use the following approach. For every $k$ with $d_k = 0$ and $g_k = 1$, we have that $q_{k-1} = \mathbb{P}(X_{k-1} = 0) = \mathbb{P}(X_k = 0) = q_k$, because if $X_{k-1} = 0$ and $g_k = 1$ we know that all arriving vehicles can pass through the system without experiencing any delay by Assumption 3. There are $g - d$ values for $k$ such that $d_k = 0$ and $g_k = 1$, so we can construct $g - d$ equations, which will set $q_{k-1}$ equal to $q_k$. Now there are two options: either $d_k = 0$ and $g_k = 1$ which means we can do the same again for $k = k + 1$, or $d_k = 1$ which implies that we can determine $q_k$ by solving the system of linear equations above. It is not possible that $g_k = 0$, because by Assumption 4 we know that every green period ends with a slot where a departure is possible.

10 Networks of Fixed-Cycle Traffic-Lights
4.2 Expected Queue Length

To find the expected length of the queue at the beginning of a cycle in steady state, we need to take the derivative of the probability generating function of \( X_0 \) in \( z \). For notational reasons we define \( N(z) := \sum_{j=0}^{c-1} q_j g_{j+1} (1 - \zeta_j (z)) \prod_{i=0}^{j} \zeta_i (z) \) the numerator of \( X_0(z) \) divided by \( z^d \) and \( D(z) := 1 - \prod_{j=0}^{c-1} \zeta_j (z) \) the denominator of \( X_0(z) \) divided by \( z^d \). Which implies that \( X_0(z) = \frac{N(z)}{D(z)} \).

By taking the derivative once and applying l'Hôpital's rule twice, we obtain

\[
X'_0(z) = \frac{D'(z)N''(z) + D(z)N'''(z) - D''(z)N(z) - D'(z)N'(z)}{2D'(z)^2 + 2D(z)D''(z)}.
\]

This implies that

\[
\mathbb{E}X_0 = X'_0(1) = \frac{D'(1)N''(1) - D''(1)N'(1)}{2D'(1)^2}.
\]

To make it more specific we will now calculate \( N'(1), N''(1), D'(1) \) and \( D''(1) \).

\[
N'(1) = - \sum_{j=0}^{c-1} q_j g_{j+1} \zeta'_j(1),
\]

\[
N''(1) = - \sum_{j=0}^{c-1} q_j g_{j+1} \left[ 2\zeta'_j(1) \sum_{i=j+1}^{c-1} \zeta'_{i+1}(1) + \zeta''_{j+1}(1) \right],
\]

\[
D'(1) = - \sum_{i=0}^{c-1} \zeta'_{i+1}(1),
\]

\[
D''(1) = - \sum_{i=0}^{c-1} \zeta''_{i+1}(1) + \zeta'_i(1) \sum_{j=0}^{c-1} \zeta'_{j+1}(1).
\]

We will now calculate \( \zeta'_{i+1}(1) \) and \( \zeta''_{i+1}(1) \).

\[
\zeta'_{k+1}(1) = \mathbb{E}Y_{k+1} - d_{k+1},
\]

\[
\zeta''_{k+1}(1) = \text{Var}Y_{k+1} - \mathbb{E}Y_{k+1} + \mathbb{E}Y^2_{k+1} + 2d_{k+1}(1 - \mathbb{E}Y_{k+1}).
\]

More detailed calculations of the derivatives above can as well be found in the report of the three months project [6].

Finally we derive a formula to calculate the expected queue length at the end of the \( k \)-th time slot \( \mathbb{E}X_k \). For this we use the derivative of the probability generating function of \( X_k \).

\[
X'_k(z) = X'_0(z) \prod_{i=0}^{k-1} \zeta_i(z) + X_0(z) \sum_{i=0}^{k-1} \zeta_i'(z) \prod_{j=0, j \neq i}^{k-1} \zeta_j(z)
\]

\[
+ \sum_{j=0}^{k-1} q_j g_{j+1}(1 - \zeta_j(z)) \sum_{i=j+1}^{k-1} \zeta_i'(z) \prod_{m=j+1, m \neq i}^{k-1} \zeta_m(z)
\]

\[
- \sum_{j=0}^{k-1} q_j g_{j+1}(\zeta_j'(z)) \prod_{i=j+1}^{k-1} \zeta_i(z).
\]

\[
\mathbb{E}X_k = X'_k(1) = \mathbb{E}X_0 + \sum_{j=0}^{k-1} (1 - q_j g_{j+1}) \zeta'_j(1).
\]
The mean queue length at the end of an arbitrary slot is given by

$$EX = \frac{1}{c} \sum_{k=0}^{c-1} EX_k.$$  

In this chapter we found a method to calculate the probability generating function of the queue length and the expected queue length at the end of every time slot in steady state. In the next chapter, we study the distribution of the delay. In Section 8.1 we will use formulas from this chapter that hold in transient state.
5 Delay Analysis

In this chapter we derive the probability generating function of the delay and a formula for the expected delay in steady state, for more details we refer to the three months project [6]. Before we can do that, we need a definition of delay.

Definition 1 (Delay at an Intersection). The delay $D$ of a vehicle at an intersection is defined as the number of time slots from the beginning of the first slot after the slot in which the vehicle arrives, until the end of the slot in which the delayed vehicle departs from the queue.

Figure 5.1 illustrates the definition of delay.

5.1 Probability Generating Function of the Delay

We tag a vehicle that arrives in time slot $m$ and introduce $U_{[m]}$ as the number of vehicles that depart before the tagged vehicle, counted from the beginning of slot $m$ and given that the tagged vehicle is delayed.

$U_{[m]} = \begin{cases} X_{m-1} | X_{m-1} > 0 + Z_m & \text{if } g_m = 1, \\ X_{m-1} + Z_m & \text{if } g_m = 0. \end{cases}$

That is $U_{[m]}$ consists of those vehicles present at the end of time slot $m - 1$ (cannot be 0 when $g_m = 1$ because the tagged vehicles is delayed) and $Z_m$ defined as the number of vehicles that arrive in the same time slot as the tagged vehicle, but before it. The pgf of $Z_m$ is given by (see Bruneel and Kim [2])

$$Z_m(z) = \frac{1 - Y_m(z)}{(1 - z)\mathbb{E}Y_m}$$

and the pgf of $U_{[m]}$ thus satisfies (since $Z_m$ and $X_{m-1}$ are independent)

$$U_{[m]}(z) = \left(\frac{X_{m-1}(z) - q_{m-1} g_m}{1 - q_{m-1} g_m}\right) \left(\frac{1 - Y_m(z)}{(1 - z)\mathbb{E}Y_m}\right).$$ (5.1)
We express $U_{[m]}$ in terms of two integer random variables $F_{[m]}$ and $R_{[m]}$:

$$U_{[m]} = d(F_{[m]} - 1) + \sum_{i=m}^{c} d_i + R_{[m]}, \quad F_{[m]} \geq 0, \quad 0 \leq R_{[m]} \leq d - 1$$

$$= d F_{[m]} + R_{[m]} - \sum_{i=1}^{m-1} d_i, \quad F_{[m]} \geq 0, \quad 0 \leq R_{[m]} \leq d - 1,$$

where $d = \sum_{i=1}^{c-1} d_i$ denotes the maximum number of delayed vehicles that can leave the system in one cycle and will leave in the situation above, because the tagged vehicle is a candidate to leave. The integer random variable $F_{[m]}$ denotes the number of complete cycles enclosed in the tagged vehicles delay and the integer random variable $R_{[m]}$ denotes the number of vehicles that will depart during the same cycle as the tagged vehicle, but before it. Let now $D_{[m]}$ denote the delay of the tagged vehicle:

$$D_{[m]} = \begin{cases} c F_{[m]} + S^{-1}(R_{[m]} + 1) - m & \text{w.p. } 1 - q_{m-1} g_m, \\ 0 & \text{w.p. } q_{m-1} g_m, \end{cases}$$

where $S^{-1}$ is the inverse function of $S : \{1, 2, \ldots, c\} \rightarrow \{0, 1, \ldots, d\} : s \mapsto \sum_{k=1}^{s} d_k$. The function $S^{-1}$ returns the slot number in which the $m$-th departure takes place, so this is $\min\{s : \sum_{k=1}^{s} d_k = m\}$.

$$D_{[m]}(z) = \sum_{i=0}^{\infty} \mathbb{P}(D_{[m]} = i) z^i$$

$$= q_{m-1} g_m + (1 - q_{m-1} g_m) z^{-m} \sum_{j=0}^{d-1} \sum_{k=0}^{\infty} \mathbb{P}(U_{[m]} = dj + k - \sum_{i=1}^{m-1} d_i) z^{j+S^{-1}(k+1)}$$

$$= q_{m-1} g_m + (1 - q_{m-1} g_m) z^{-m} \sum_{k=0}^{d-1} z^{S^{-1}(k+1)} \theta_{mk}(z),$$

where

$$\theta_{mk}(z) = \sum_{j=0}^{\infty} \mathbb{P}(U_{[m]} = dj + k - \sum_{i=1}^{m-1} d_i) z^j$$

$$= \sum_{l=0}^{\infty} \mathbb{P}(U_{[m]} = l) z^{(l-k+\sum_{i=1}^{m-1} d_i)} \sum_{j=-\infty}^{\infty} \delta(l - dj - k + \sum_{i=1}^{m-1} d_i),$$

where $\delta(.)$ is the Dirac-delta function.

The problem is now that we can not directly substitute the pgf of $U_{[m]}$. We will use the following property, such that we can replace the Dirac-delta functions by summations, which allows us to substitute the pgf of $U_{[m]}$.

**Property 1.**

$$\frac{1}{d} \sum_{i=0}^{d-1} a^{im} = \sum_{j=-\infty}^{\infty} \delta(m - dj)$$

(5.2)

where $a = e^{2\pi i / d}$ and $m$ and $d$ integer values. The sum on the left-hand side is zero unless $m$ is a multiple of $d$. 

14 Networks of Fixed-Cycle Traffic-Lights
Using this property we obtain
\[
\theta_{mk}(z) = \sum_{j=0}^{\infty} \mathbb{P}(U_{[m]} = l) z^{l-k + \sum_{i=1}^{m-1} d_i} \frac{1}{d} \sum_{t=0}^{d-1} a^t z \frac{d}{d} \sum_{j=0}^{\infty} \mathbb{P}(U_{[m]} = l) z^l a^l
\]
\[
= \frac{e^{\left(\sum_{i=1}^{m-1} d_i \cdot k\right)}}{d} \sum_{t=0}^{d-1} a^t \left(\sum_{i=1}^{m-1} d_i \cdot k\right) U_{[m]} \left(z \frac{d}{d} a^t\right).
\]

\[
D_{[m]}(z) = q_{m-1}g_m + (1 - q_{m-1}g_m) z^{-m} \sum_{k=0}^{d-1} z^{-S^{-1}(k+1)} \frac{e^{\left(\sum_{i=1}^{m-1} d_i \cdot k\right)}}{d} \sum_{t=0}^{d-1} a^t \left(\sum_{i=1}^{m-1} d_i \cdot k\right) U_{[m]} \left(z \frac{d}{d} a^t\right)
\]
\[
= q_{m-1}g_m
\]
\[
+ (1 - q_{m-1}g_m) \frac{z^{-m} \sum_{k=0}^{d-1} U_{[m]} \left(z \frac{d}{d} a^t\right)}{d} \sum_{k=0}^{d-1} z^{-S^{-1}(k+1)} + \frac{e^{\left(\sum_{i=1}^{m-1} d_i \cdot k\right)}}{d} \sum_{t=0}^{d-1} a^t \left(\sum_{i=1}^{m-1} d_i \cdot k\right).
\]

(5.3)

And finally after deconditioning
\[
D(z) = \frac{\sum_{m=1}^{c} \mathbb{E}Y_m D_{[m]}(z)}{\sum_{m=1}^{c} \mathbb{E}Y_m}.
\]

(5.4)

5.2 Expected Delay

We calculate the derivative of (5.4) and evaluate it in \(z = 1\) to find an expression for the expected delay.

\[
D'(z) = \frac{\sum_{m=1}^{c} \mathbb{E}Y_m D'_{[m]}(z)}{\sum_{m=1}^{c} \mathbb{E}Y_m}
\]

\[
\mathbb{E}D = \frac{\sum_{m=1}^{c} \mathbb{E}Y_m D_{[m]}(z)}{\sum_{m=1}^{c} \mathbb{E}Y_m}.
\]

We calculate the derivative of (5.3) and evaluate it in \(z = 1\) to find an expression for the expected delay knowing a vehicle arrives in a certain time slot.

\[
D'_{[m]}(z) = (1 - q_{m-1}g_m) \frac{z^{-m} \sum_{k=0}^{d-1} U_{[m]} \left(z \frac{d}{d} a^t\right) \sum_{k=0}^{d-1} z^{-S^{-1}(k+1)} + \frac{e^{\left(\sum_{i=1}^{m-1} d_i \cdot k\right)}}{d} \sum_{t=0}^{d-1} a^t \left(\sum_{i=1}^{m-1} d_i \cdot k\right)}{d}
\]
\[
+ (1 - q_{m-1}g_m) \frac{z^{-m} \sum_{k=0}^{d-1} U_{[m]} \left(z \frac{d}{d} a^t\right)}{d} \sum_{k=0}^{d-1} z^{-S^{-1}(k+1)} + \frac{e^{\left(\sum_{i=1}^{m-1} d_i \cdot k\right)}}{d} \sum_{t=0}^{d-1} a^t \left(\sum_{i=1}^{m-1} d_i \cdot k\right)
\]
\[
+ (1 - q_{m-1}g_m) \frac{z^{-m} \sum_{k=0}^{d-1} U_{[m]} \left(z \frac{d}{d} a^t\right)}{d} \sum_{k=0}^{d-1} \left(S^{-1}(k+1) + \frac{e^{\left(\sum_{i=1}^{m-1} d_i \cdot k\right)}}{d} \sum_{t=0}^{d-1} a^t \left(\sum_{i=1}^{m-1} d_i \cdot k\right) \right).
\]

15 Networks of Fixed-Cycle Traffic-Lights
Furthermore we calculate the derivative of (5.1).

\[ \mathbb{E}D_{[m]} = D'_{[m]}(1) \]

\[ = (1 - q_{m-1}g_m) \sum_{i=0}^{d-1} U_{[m]}(d') \sum_{k=0}^{d-1} a' \left( \sum_{i=1}^{m-1} d_i - k \right) \]

\[ + (1 - q_{m-1}g_m) \frac{1}{d} \sum_{t=0}^{d-1} U'_{[m]}(d') \sum_{k=0}^{d-1} a' \left( \sum_{i=1}^{m-1} d_i - k \right) \]

\[ + (1 - q_{m-1}g_m) \frac{1}{d} \sum_{t=0}^{d-1} U_{[m]}(d') \]

\[ \cdot \sum_{k=0}^{d-1} \left( S^{-1}(k + 1) + \frac{c \left( \sum_{i=1}^{m-1} d_i - k \right)}{d} \right) a' \left( \sum_{i=1}^{m-1} d_i - k \right). \]

Furthermore we calculate the derivative of (5.1).

\[ U'_{[m]}(z) = \left( \frac{X'_{m-1}(z) - q_{m-1}g_m}{1 - q_{m-1}g_m} \right) \left( \frac{1 - Y_m(z)}{(1 - z)\mathbb{E}Y_m} \right) \]

\[ + \left( \frac{X_{m-1}(z) - q_{m-1}g_m}{1 - q_{m-1}g_m} \right) \frac{1}{\mathbb{E}Y_m} \left( \frac{1 - Y_m(z)}{(1 - Y_m(z)) - (1 - z)Y'_m(z)} \right). \]

We use l'Hôpital on \( \frac{1 - Y_m(z)}{(1 - z)\mathbb{E}Y_m} \) and \( \frac{1 - Y_m(z) - (1 - z)Y'_m(z)}{(1 - z)^2} \) for the limit of \( z \) to one.

\[ \lim_{z \to 1} \frac{1 - Y_m(z)}{(1 - z)\mathbb{E}Y_m} = \lim_{z \to 1} -\frac{Y'_m(z)}{\mathbb{E}Y_m} \]

\[ = \lim_{z \to 1} \frac{Y'_m(z)}{\mathbb{E}Y_m} \]

\[ = \frac{Y''_m(1)}{\mathbb{E}Y_m}. \]

\[ \lim_{z \to 1} \frac{1 - Y_m(z) - (1 - z)Y'_m(z)}{(1 - z)^2} = \lim_{z \to 1} \frac{(-Y'_m(z) - (1 - z)Y''_m(z) + Y''_m(z))}{-2(1 - z)} \]

\[ = \lim_{z \to 1} \frac{Y''_m(z)}{2} \]

\[ = \frac{Y''_m(1)}{2}. \]

So for the limit of \( z \) to one we have:

\[ U'_{[m]}(1) = \left( \frac{X'_{m-1}(1) - q_{m-1}g_m}{1 - q_{m-1}g_m} \right) \left( \frac{Y'_m(1)}{\mathbb{E}Y_m} \right) + \left( \frac{X_{m-1}(1) - q_{m-1}g_m}{1 - q_{m-1}g_m} \right) \left( \frac{1}{\mathbb{E}Y_m} \right) \left( \frac{Y''_m(1)}{2} \right) \]

\[ = \left( \frac{\mathbb{E}X_{m-1} - q_{m-1}g_m}{1 - q_{m-1}g_m} \right) \left( \frac{\mathbb{E}Y_m}{\mathbb{E}Y_m} \right) + \left( \frac{1}{\mathbb{E}Y_m} \right) \left( \frac{Y''_m(1)}{2\mathbb{E}Y_m} \right) \]

\[ = \frac{\mathbb{E}X_{m-1} - q_{m-1}g_m}{1 - q_{m-1}g_m} + \frac{Y''_m(1)}{2\mathbb{E}Y_m}. \]

Alternatively, one can apply Little’s Law to the mean queue length to find an expression for the expected delay \( \mathbb{E}D = c\mathbb{E}X / \sum_{k=0}^{\infty} \mathbb{E}Y_{k+1}. \)

16 Networks of Fixed-Cycle Traffic-Lights
6 Comparison of the Models

In this Chapter we compare the models discussed in Chapter 2 as well as the model introduced in Chapters 3, 4 and 5. We will compare their results for the expected overflow queue and the expected delay. In these comparisons the arrival process is the same for every time slot, this because only our model can allow different arrival processes.

6.1 Comparison Expected Overflow Queue

In this section we will compare the expected overflow queue for the model of Darroch [3] and the model described in Chapters 3, 4 and 5. Notice that we have a lower and upper bound by Darroch's model. In Table 6.1 we compare the expected overflow queue of these models for Poisson, Bernoulli and Geometric arrival processes with different expectations and this for a cycle with a green and red period of size five. The model by Van Leeuwaarden [9] would give the same results, since it is a special case of our model. We notice that the result of our model and equivalently Van Leeuwaarden's model, fits perfectly into the interval found by Darroch's model and all this for Poisson, Bernoulli and Geometric arrival process.

<table>
<thead>
<tr>
<th>$E[Y]$</th>
<th>Darroch</th>
<th>Our Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>lower bound</td>
<td>upper bound</td>
</tr>
<tr>
<td>Poisson</td>
<td>0.30</td>
<td>0.0909</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.3070</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>0.8924</td>
</tr>
<tr>
<td></td>
<td>0.45</td>
<td>3.0141</td>
</tr>
<tr>
<td></td>
<td>0.49</td>
<td>21.7746</td>
</tr>
<tr>
<td>Bernoulli</td>
<td>0.30</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.0805</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>0.3481</td>
</tr>
<tr>
<td></td>
<td>0.45</td>
<td>1.3857</td>
</tr>
<tr>
<td></td>
<td>0.49</td>
<td>10.6466</td>
</tr>
<tr>
<td>Geometric</td>
<td>0.30</td>
<td>0.1994</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0.5518</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>1.4620</td>
</tr>
<tr>
<td></td>
<td>0.45</td>
<td>4.6829</td>
</tr>
<tr>
<td></td>
<td>0.49</td>
<td>33.5442</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison of the expected overflow queue for a cycle with $g = r = d = 5$
6.2 Comparison Expected Delay

In this section we will compare the expected delay for the model of Webster [10], McNeill [7], Darroch [3] and our model. Notice that we have again a lower and upper bound by Darroch’s model. In Table 6.2 we now compare the expected delay of these models for Poisson, Bernoulli and Geometric arrival processes with different expectations and this again for a cycle with a green and red period of size five.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>2.9720</td>
<td>3.0816</td>
<td>2.5123</td>
<td>3.5357</td>
<td>2.7245</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.6900</td>
<td>3.8662</td>
<td>3.1895</td>
<td>4.4212</td>
<td>3.4816</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.1383</td>
<td>5.4801</td>
<td>4.6370</td>
<td>6.1667</td>
<td>5.0634</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>49.3937</td>
<td>50.3707</td>
<td>46.9788</td>
<td>51.0604</td>
<td>49.8805</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bernoulli</td>
<td>2.6812</td>
<td>2.4129</td>
<td>2.0000</td>
<td>2.3240</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.1266</td>
<td>2.4847</td>
<td>3.5238</td>
<td>3.7420</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.9925</td>
<td>3.2251</td>
<td>4.5000</td>
<td>3.5758</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.5303</td>
<td>5.5267</td>
<td>7.2222</td>
<td>6.0759</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26.5660</td>
<td>24.2428</td>
<td>27.0408</td>
<td>26.0759</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geometric</td>
<td>3.5204</td>
<td>2.9237</td>
<td>4.0714</td>
<td>3.1633</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.6571</td>
<td>3.9347</td>
<td>5.3187</td>
<td>4.2724</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7.0322</td>
<td>6.1015</td>
<td>7.8333</td>
<td>6.6156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14.3917</td>
<td>12.9314</td>
<td>15.4040</td>
<td>13.9371</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>74.2647</td>
<td>70.9984</td>
<td>75.0800</td>
<td>73.7745</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Comparison of the expected delay for a cycle with $g = r = d = 5$

We see that the results of our model fit in the interval constructed by the lower and upper bound by Darroch's model. And also the approximation of Webster and the solution of McNeill fit into this interval by Darroch. To calculate the result of McNeill a value for the expected overflow queue is needed, in this table we used the result of our model to calculate this.

We conclude by giving an advice for when to use which model. When we just want a good indication of the expected delay we can use the formula of webster in case of Poisson arrivals. When we have an indication of the expected overflow queue we can use McNeill's model to quickly find the expected delay, but the arrival process needs to be Compound Poisson. If the arrival process is not a Compound Poisson process and we still want just an indication of the expected overflow queue and the expected delay for an arrival process, one could use the lower and upper bound provided by Darroch, which is again an easy computation. If we want to know the expected overflow queue and the expected delay quite precise or if we want some more information about the overflow queue, which can be derived from its probability generating function, we can use Darroch's, Van Leeuwaarden’s or our model. For all three of them the calculation become more complex. Actually it is as hard as finding the roots of (4.12). If we want more information about the delay, which can be derived from its probability generating function, only Van Leeuwaarden’s and our model remain a possibility. Finally if the cycle has multiple green periods and/or we have different arrival processes for every time slot and/or we have green time slot where no vehicle can depart only our model can give a result.
7 Simple Examples

In this section we will show some examples for which we use two of the extensions discussed in Chapter 3. We will use the possibility to have a green time slot where no vehicle can leave the system and only arriving vehicles that find an empty queue can pass through. This to model that the first departing vehicle, when the traffic light turns green may need some more time to depart. We will compare this with the case that the first vehicle has no delay and the case that we have the red period a time slot longer. Furthermore we will use the possibility to have multiple green periods in a cycle to see if there are advantages of using this and finally we we show a road construction example.

7.1 No Departure in a Green Time Slot

In this section we will compare the characteristics for a cycle of length ten with time slot two till five green \( g_i = 1 \) and \( d_i = 1 \) as well. Time slot six till ten are red \( g_i = 0 \) and therefor we have \( d_i = 0 \). For the first time slot we consider all the possibilities for \( g_1 \) and \( d_1 \).

(a) First time slot green

(b) First time slot green but no departure

(c) First time slot red

Figure 7.1: Three situations analyzed in Table 7.1 and 7.2

The situation in Figure 7.1b can be used to model the situation that the first vehicle to depart form a queue when the traffic light turns green, needs more time to depart the queue than the vehicle departing after him. We compare the characteristics for the overflow queue and the delay with the other two situations to see if this makes a big difference in the characteristics. This to conclude if modeling such a situations is useful or if it can be ignored.

19 Networks of Fixed-Cycle Traffic-Lights
In Table 7.1 we see that not allowing a vehicle to depart in time slot one has a big negative influence on the length of the overflow queue. Furthermore we notice that if $d_1 = 0$ changing $g_1$ from one to zero has a small but significant negative influence for the length of the overflow queue, in this case the only difference in the model is that a vehicle arriving in time slot one and finds an empty queue cannot depart anymore.

In Table 7.2 we can see the same as for the overflow queue. Not allowing a vehicle to depart in time slot one has a big negative influence on the expected delay. Furthermore we notice that if $d_1 = 0$ changing $g_1$ from one to zero has a small again significant negative influence for the delay, in this case the only difference in the model is that a vehicle arriving in time slot one and does find an empty queue can not depart anymore. We conclude that it might be useful to model situations where the first vehicle to depart needs more time to depart the queue to reach more precise results.

**Table 7.1: Characteristics of the overflow queue $X_g$ with $g = r = 5$**

<table>
<thead>
<tr>
<th>$E Y$</th>
<th>$d_1$</th>
<th>$g_1$</th>
<th>$E X_g$</th>
<th>$Var X_g$</th>
<th>$P(X_g \geq 10)$</th>
<th>$P(X_g \geq 20)$</th>
<th>$P(X_g \geq 30)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>0.30</td>
<td>1</td>
<td>1</td>
<td>0.1800</td>
<td>0.4285</td>
<td>2.92 \cdot 10^{-2}</td>
<td>2.25 \cdot 10^{-9}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>1</td>
<td>0.7677</td>
<td>2.3994</td>
<td>2.46 \cdot 10^{-3}</td>
<td>1.00 \cdot 10^{-5}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>0</td>
<td>0.7720</td>
<td>2.4070</td>
<td>2.47 \cdot 10^{-3}</td>
<td>1.01 \cdot 10^{-5}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>1</td>
<td>1</td>
<td>0.4399</td>
<td>1.2707</td>
<td>5.72 \cdot 10^{-4}</td>
<td>6.66 \cdot 10^{-7}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>1</td>
<td>2.5767</td>
<td>13.4347</td>
<td>5.72 \cdot 10^{-2}</td>
<td>4.19 \cdot 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>0</td>
<td>2.5844</td>
<td>13.4496</td>
<td>5.74 \cdot 10^{-2}</td>
<td>4.20 \cdot 10^{-3}</td>
</tr>
<tr>
<td>Geometric</td>
<td>0.30</td>
<td>1</td>
<td>1</td>
<td>0.3000</td>
<td>0.9509</td>
<td>4.69 \cdot 10^{-4}</td>
<td>6.19 \cdot 10^{-7}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>1</td>
<td>1.1065</td>
<td>4.5538</td>
<td>1.03 \cdot 10^{-2}</td>
<td>1.84 \cdot 10^{-4}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>0</td>
<td>1.1177</td>
<td>4.5810</td>
<td>1.04 \cdot 10^{-2}</td>
<td>1.85 \cdot 10^{-4}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>1</td>
<td>1</td>
<td>0.7055</td>
<td>2.7914</td>
<td>4.42 \cdot 10^{-3}</td>
<td>4.16 \cdot 10^{-5}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>1</td>
<td>3.6492</td>
<td>25.8171</td>
<td>1.16 \cdot 10^{-1}</td>
<td>1.75 \cdot 10^{-2}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>0</td>
<td>3.6690</td>
<td>25.8716</td>
<td>1.17 \cdot 10^{-1}</td>
<td>1.76 \cdot 10^{-2}</td>
</tr>
</tbody>
</table>

**Table 7.2: Characteristics of the delay $D$ with $g = r = 5$**

<table>
<thead>
<tr>
<th>$E Y$</th>
<th>$d_1$</th>
<th>$g_1$</th>
<th>$E D$</th>
<th>$Var D$</th>
<th>$P(D = 0)$</th>
<th>$P(D \geq 10)$</th>
<th>$P(D \geq 20)$</th>
<th>$P(D \geq 30)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>0.30</td>
<td>1</td>
<td>1</td>
<td>2.7245</td>
<td>6.5537</td>
<td>2.86 \cdot 10^{-1}</td>
<td>1.83 \cdot 10^{-4}</td>
<td>1.56 \cdot 10^{-6}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>1</td>
<td>5.3502</td>
<td>23.3558</td>
<td>1.65 \cdot 10^{-1}</td>
<td>1.57 \cdot 10^{-1}</td>
<td>1.73 \cdot 10^{-2}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>0</td>
<td>5.3681</td>
<td>23.1871</td>
<td>1.50 \cdot 10^{-1}</td>
<td>1.57 \cdot 10^{-1}</td>
<td>1.73 \cdot 10^{-2}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>1</td>
<td>1</td>
<td>3.4817</td>
<td>10.6534</td>
<td>2.31 \cdot 10^{-1}</td>
<td>5.33 \cdot 10^{-2}</td>
<td>1.79 \cdot 10^{-3}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>1</td>
<td>10.2952</td>
<td>93.7505</td>
<td>7.69 \cdot 10^{-2}</td>
<td>4.05 \cdot 10^{-1}</td>
<td>1.42 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>0</td>
<td>10.2604</td>
<td>93.9388</td>
<td>8.84 \cdot 10^{-2}</td>
<td>4.04 \cdot 10^{-1}</td>
<td>1.42 \cdot 10^{-1}</td>
</tr>
<tr>
<td>Geometric</td>
<td>0.30</td>
<td>1</td>
<td>1</td>
<td>3.1632</td>
<td>10.3205</td>
<td>2.86 \cdot 10^{-1}</td>
<td>4.72 \cdot 10^{-2}</td>
<td>1.69 \cdot 10^{-2}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>1</td>
<td>6.5606</td>
<td>39.9897</td>
<td>1.43 \cdot 10^{-1}</td>
<td>2.32 \cdot 10^{-1}</td>
<td>4.63 \cdot 10^{-2}</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0</td>
<td>0</td>
<td>6.4927</td>
<td>40.2026</td>
<td>1.68 \cdot 10^{-1}</td>
<td>2.30 \cdot 10^{-1}</td>
<td>4.60 \cdot 10^{-2}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>1</td>
<td>1</td>
<td>4.2724</td>
<td>19.1995</td>
<td>2.31 \cdot 10^{-1}</td>
<td>1.11 \cdot 10^{-1}</td>
<td>1.06 \cdot 10^{-2}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>1</td>
<td>13.4043</td>
<td>176.0569</td>
<td>7.69 \cdot 10^{-2}</td>
<td>4.96 \cdot 10^{-1}</td>
<td>2.33 \cdot 10^{-1}</td>
</tr>
<tr>
<td></td>
<td>0.35</td>
<td>0</td>
<td>0</td>
<td>13.3312</td>
<td>176.2451</td>
<td>9.05 \cdot 10^{-2}</td>
<td>4.93 \cdot 10^{-1}</td>
<td>2.31 \cdot 10^{-1}</td>
</tr>
</tbody>
</table>

### 7.2 Multiple Green Periods in a Cycle

In this section we compare what happens if we have one green period in a cycle or two green periods with the same total size in a cycle. What is better for the delay of a vehicle and what is
better for the queue length? To analyze this we will take a look at model with a cycle length of ten
time slots with in total five green slots in a cycle. In one model we will consider one green period
of size five as shown in Figure 7.2a, in the other model we will consider two green periods, the
first one of size three, followed by three red time slots again followed by two green time slots, this
is shown in Figure 7.2b.

<table>
<thead>
<tr>
<th>$d_i$</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_i$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

(a) Situation with one green period

<table>
<thead>
<tr>
<th>$d_i$</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_i$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

(b) Situation with two green periods

Figure 7.2: Two situations analyzed in Table 7.3 and 7.4

In this section we consider $X_0$ to be the queue length at the beginning of the corresponding green
period and $X_g$ the queue length at the end of that green period (the overflow queue).

<table>
<thead>
<tr>
<th></th>
<th>$E X_0$</th>
<th>$\text{Var} X_0$</th>
<th>$E X_g$</th>
<th>$\text{Var} X_g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>5.6498</td>
<td>24.0046</td>
<td>3.3998</td>
<td>21.7546</td>
</tr>
<tr>
<td>Geometric</td>
<td>7.4307</td>
<td>51.3861</td>
<td>5.1807</td>
<td>48.1236</td>
</tr>
</tbody>
</table>

(a) One green period with $g = 5$

<table>
<thead>
<tr>
<th>Green Period</th>
<th>$E X_0$</th>
<th>$\text{Var} X_0$</th>
<th>$E X_g$</th>
<th>$\text{Var} X_g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>1</td>
<td>4.8125</td>
<td>23.6299</td>
<td>3.5036</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4.8536</td>
<td>23.3225</td>
<td>3.9125</td>
</tr>
<tr>
<td>Geometric</td>
<td>1</td>
<td>6.6305</td>
<td>50.8392</td>
<td>5.3142</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6.6642</td>
<td>50.4616</td>
<td>5.7305</td>
</tr>
</tbody>
</table>

(b) Two green periods with $g^{(1)} = 3$, $r^{(1)} = 3$ and $g^{(2)} = 2$

Table 7.3: Expectation and variance of the queue length at the beginning and end of the green
periods

In Table 7.3 we see that for one contiguous green period the expected overflow queue is smaller
than the expected overflow queue at the end of the green periods in a cycle with two green
periods. For the expected queue length at the beginning of the green period we see the opposite
relation. For the contiguous green period it is bigger than for the two green periods.
In Table 7.4 we see the same relation for the Expected queue length and the delay, between the situations with the contiguous green period and the situation with two green periods. This is obvious by the fact that the two are related by Little's Theorem. We see that the expectations are lower for the situation where we have multiple green periods. Therefore one would conclude that the more green periods in a cycle, the lower the delay. This is true, but one should keep in mind that there are multiple traffic streams at an intersection and that when a certain traffic stream gets a red light all the other traffic lights need to stay red for a while, such that the intersection can be cleared. So in practice one needs to find a balance such that an optimal delay over all the traffic streams can be accomplished.

<table>
<thead>
<tr>
<th></th>
<th>$\mathbb{E}X$</th>
<th>$\text{Var}X$</th>
<th>$\mathbb{E}D$</th>
<th>$\text{Var}D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>4.4954</td>
<td>23.6325</td>
<td>9.9675</td>
<td>94.6784</td>
</tr>
<tr>
<td>Geometric</td>
<td>6.2717</td>
<td>50.5891</td>
<td>13.9372</td>
<td>204.9138</td>
</tr>
</tbody>
</table>

(a) One green period with $g = 5$

<table>
<thead>
<tr>
<th></th>
<th>$\mathbb{E}X$</th>
<th>$\text{Var}X$</th>
<th>$\mathbb{E}D$</th>
<th>$\text{Var}D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>4.2375</td>
<td>23.1302</td>
<td>9.4167</td>
<td>93.4275</td>
</tr>
<tr>
<td>Geometric</td>
<td>6.0536</td>
<td>50.0608</td>
<td>13.4524</td>
<td>203.8670</td>
</tr>
</tbody>
</table>

(b) Two green periods with $g^{(1)} = 3, r^{(1)} = 3$ and $g^{(2)} = 2$

Table 7.4: Expectation and variance of the queue length and delay

In Figure 7.3 one can see the delay distribution for a vehicle arriving in the first time slot in a traffic light situation as described in this section with two green periods. The blocks with not bars correspond nicely with the red periods, while the blocks with bars correspond with the green periods.

Figure 7.3: Delay distribution of a vehicle that arrives in the first time slot

In Figure 7.3 one can see the delay distribution for a vehicle arriving in the first time slot in a traffic light situation as described in this section with two green periods. The blocks with not bars correspond nicely with the red periods, while the blocks with bars correspond with the green periods.

22 Networks of Fixed-Cycle Traffic-Lights
7.3 Road Construction Example: Optimizing Signal Settings

In this section we take a look at a road construction example. A road is under construction and all vehicles in both directions have to drive on the same lane as illustrated in Figure 7.4. So they have to take turns by traffic lights.

![Figure 7.4: Situation of the road construction example](image)

In this example the vehicles from one direction arrive according to a Poisson process with arrival rate $0.10$ and the vehicles of the other direction arrive according to a Poisson process as well, but with an arrival rate of $0.30$. When the traffic light turns red for one direction, it takes some time until the lane is clear for the other direction to start driving. In this example is the time both traffic lights have to be red two time slots. Notice that we have two queues building and that making the green period corresponding to one queue larger implies that the red period of the other queue becomes larger, or equivalent to this the green period becomes smaller. When the length of the green period of the first queue is set to $g^{(1)}$, the length of the green period of the other queue is known immediately to be $g^{(2)} = c - g^{(1)} - 4$. This means we have a conflict of interest. An obvious question to ask for this example is for which cycle length and length of the green periods is the delay of an arbitrary vehicle optimal. In Figure 7.5 we show one example of the settings for the green periods of the road construction example. Notice that like needed the green periods of the two queues do not overlap and there is a gap of two time slots between the green periods to make sure the lane is cleared before vehicles form the other direction start driving.

| Queue 1: $g^{(1)}$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Queue 2: $g^{(2)}$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

![Figure 7.5: Green periods for $c = 9$ and the length of the green period of the first traffic light is 2](image)

In Table 7.5 we get an overview of the expected delay for an arbitrary arriving vehicle for all the different possible settings of the traffic lights. A value of $\infty$ implies that one of the two traffic light queues is unstable. We can see that the optimal is when we use a cycle length of 15, a green period of size 3 for the first traffic light and a green period of size 8 for the second traffic light. The set of solutions is in principle an convex set which should make it easy to find the optimal without having to calculate all the results above, but due to the fact we work with discrete time slots, it is not convex anymore. Notice the optimal for a cycle length 18 is lower than the optimal for a cycle length of 17, while the overall optimal is for cycle length 15.

23 Networks of Fixed-Cycle Traffic-Lights
<table>
<thead>
<tr>
<th>Cyclelength</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>9.8988</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>9</td>
<td>13.8338</td>
<td>12.5517</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>10</td>
<td>∞</td>
<td>5.4404</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>11</td>
<td>∞</td>
<td>4.4958</td>
<td>7.3413</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>12</td>
<td>∞</td>
<td>4.3561</td>
<td>4.9230</td>
<td>12.7191</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>13</td>
<td>∞</td>
<td>4.5412</td>
<td>4.2586</td>
<td>6.1610</td>
<td>50.2890</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>14</td>
<td>∞</td>
<td>4.9712</td>
<td>4.0263</td>
<td>4.8593</td>
<td>8.3814</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>15</td>
<td>∞</td>
<td>5.6887</td>
<td>3.9708</td>
<td>4.3766</td>
<td>5.7936</td>
<td>13.1764</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>16</td>
<td>∞</td>
<td>6.8515</td>
<td>4.0129</td>
<td>4.1675</td>
<td>4.9583</td>
<td>7.1699</td>
<td>32.0385</td>
<td>∞</td>
</tr>
<tr>
<td>17</td>
<td>∞</td>
<td>8.8661</td>
<td>4.1239</td>
<td>4.0827</td>
<td>4.5912</td>
<td>5.7253</td>
<td>9.3869</td>
<td>∞</td>
</tr>
</tbody>
</table>

Table 7.5: Expected delay for an arbitrary vehicle in a road construction $\lambda_1 = 0.10, \lambda_2 = 0.30$ (Poisson arrivals)
8 Networks of Intersections

In this chapter we will construct some techniques that can be used to analyze methods that can be used in networks of intersections. The first one in Section 8.1 is to find the probability that some queue in front of a traffic light becomes larger than some given threshold length. This can for instance be used as probability that the queue building in front of the traffic light will block the previous intersection, which is something one would like to avoid. The second technique is a way to construct output processes of an intersection. This can be used to connect intersections into networks and to construct more realistic arrival processes for the queues. The third and final technique is a way to calculate the probability generating function of the delay in a network of intersection. After that we will consider networks of multiple intersections with the same settings. We will show what happens with the output processes over time.

8.1 Tail Probability for Transient and Steady State

In a network of intersections it might be the case that the distance between two consecutive intersection is small, therefore it is useful to have a method that can find the probability that the queue length becomes more than some threshold length within some number of cycles \( f \), given the settings of some traffic light. As a threshold one could use the largest number of vehicles that can be waiting in the queue without blocking the previous intersection. To find such a probability, we will find the probability that the queue length stays beneath the threshold length \( t \) within a predefined number of cycles \( f \) \( (P(X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c,f} \leq t)) \), which is the complementary event of the event of which we want to find the probability. By using the formula of the conditional probability and the fact that \( X_{k,n} \) only depends on \( X_{k-1,n} \) the queue length of the previous time slot, we can find a recursive formula. Remember that \( X_{0,n} = X_{c,n-1} \).

\[
P(X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c,f} \leq t) = P(X_{c,f} \leq t | X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c-1,f} \leq t) \cdot P(X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c-1,f} \leq t) = P(X_{c,f} \leq t | X_{c-1,f} \leq t) P(X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c-1,f} \leq t).
\]

The first factor in this equation is a conditional probability which we will simplify below, the second factor is the same probability as we started with for one time slot less. By using this formula multiple times recursively, we can construct an exact formula with a multitude of those conditional formulas. Furthermore this makes it easy to use in an algorithm and construct the desired solution by starting in the first time slot and continue until the desired time slot.

\[
P(X_{k,n} \leq t | X_{k-1,n} \leq t) = \frac{P(X_{k,n} \leq t, X_{k-1,n} \leq t)}{P(X_{k-1,n} \leq t)} = \frac{\sum_{i=0}^{t} P(X_{k,n} \leq t | X_{k-1,n} = i) P(X_{k-1,n} = i)}{\sum_{j=0}^{t} P(X_{k-1,n} = j)}.
\]
The probabilities \( P(X_{k-1,n} = j) \) can easily be calculated from the probability generating function of the \( k-1 \)-th time slot \( (X_{k-1,n}(z)) \). The conditional probabilities we split into the sum of conditional probabilities which are more easy to derive.

\[
P(X_{k,n} \leq t | X_{k-1,n} = i) = \sum_{l=0}^{t} P(X_{k,n} = l | X_{k-1,n} = i).
\]

By Equation (4.4) and using the fact that we know that \( X_{k-1,n}(z) = z^i \). We can find a conditional probability generating function from which we can derive the individual conditional probabilities. Furthermore these conditional probabilities only depend on the slot number in the cycle and the condition and not on the current cycle. Thanks to this, we only need to calculate these probabilities for one cycle and remember those probabilities, because we can reuse them in every cycle.

\[
\sum_{i=0}^{\infty} P(X_{k,n} = l | X_{k-1,n} = i)z^i = \begin{cases} 
\xi_{k,n}(z) + (1 - \xi_{k,n}(z)) g_k, & \text{if } i = 0, \\
\xi_{k,n}(z)z^i, & \text{if } i > 0.
\end{cases}
\]

Quite easily we can now calculate \( P(X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c,f} \leq t) \) by a recursive algorithm. Notice that one should still need a starting point. One could use any number of vehicles in the queue for the first queue in time \( X_{1,0} \), but of course taking a value more than \( t \) will already be a situation outside our event, which means that our probability will become one. Remember that \( 1 - P(X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c,f} \leq t) \) is the probability we are looking for. One could also use a direct formula. But we would advise to use it recursive, because in this way one could save memory space during calculations. For completion we state the direct formula below.

\[
P(X_{1,0} \leq t, X_{2,0} \leq t, \ldots, X_{c,f} \leq t) = P(X_{1,0} \leq t) \left( \prod_{k=2}^{c} \frac{\sum_{j=0}^{t} P(X_{k,0} \leq t | X_{k-1,0} = i) P(X_{k-1,0} = j)}{\sum_{j=0}^{t} P(X_{k-1,0} = j)} \right) \cdot \left( \prod_{n=1}^{f} \prod_{k=1}^{c} \frac{\sum_{l=0}^{t} P(X_{k,n} \leq t | X_{k-1,n} = i) P(X_{k-1,n} = j)}{\sum_{l=0}^{t} P(X_{k-1,n} = j)} \right).
\]

### 8.1.1 Example

We will now consider a traffic light with a cycle length of 60 seconds divided into 60 time slots. The first half of the time slots is green, the other half red. We will now keep track of the tail probability over time. During the first 8 minutes the vehicles arrive according a Poisson process with arrival rate \( \lambda = 0.45 \) for every time slot, after that the arrival rate increases to \( \lambda = 0.55 \) for 8 minutes. And we will finish again with an arrival rate of \( \lambda = 0.45 \) for 8 minutes. We will as well keep track of the expected queue length over time. In Figure 8.1a we illustrate how the arrival rate changes over time. The red line represents the switching point between a stable and an unstable arrival rate for the system. For \( \lambda = 0.5 \) or more the system will be unstable. Fortunately we can still use the equations of this chapter in an unstable situation. We assume that in front of the traffic light there is space for 40 vehicles in the queue. If more vehicles would joint the queue, an early intersection could be blocked. Therefore we will use \( t = 40 \) as our threshold value for the tail probabilities.
(a) Arrival rate over time

(b) Expected queue length over time

(c) Tail probability over time
In Figure 8.1b we can see the expected queue length over time. We can see clearly the cycles in the graph, small upward movements during the red periods and downward movements during the green periods. During the first 8 minutes, with a stable arrival rate, we see that we have an upward trend in the beginning. The slope however gets less over time. It is going to a stable situation, but does not reach it completely after 8 minutes. After those first 8 minutes the arrival rate changes and we notice that from there on we have an upward trend and the slope of this trend stays the same for the whole 8 minutes. This happens because of the unstable arrival rate. The system will not converge to some stable situation, instead the queue keeps growing cycle after cycle. In the final 8 minutes we have again a stable arrival rate and notice that a negative trend starts immediately. We also see that the slope of this trend becomes less negative over time. Actually the system is again converging to the same stable situation it was converging to in the first 8 minutes.

In Figure 8.1c we see what happens to the tail probability, stating the probability that from time zero until the current time the length of the queue becomes larger than our threshold length $t = 40$. The probability starts at zero, because we started from an empty system. We see that in the stable situation the probability increases, but it is that slow that it is almost not noticeable. When the arrival rate increases after 8 minutes, the probability starts increasing faster and faster until it gets quite close to one. Therefor it has to slow down a bit. When the arrival rate decreases again, the increasing of the probability keeps going down faster and faster, but the harm is already done by the second 8 minutes.

8.2 Output Process

With our model described in Chapter 3, 4 and 5 we are able to analyze the queue length and delay of an intersection for any given set of arrival processes. In practice these arrival processes depend on the previous intersection. therefor we are going to describe a way to find the output process of an intersection, such that we can use that output process for the arrival process of the next intersection.

Given the arrival process for every time slot, we can find the probability generating function of the queue length and the delay in every time slot. For the output process $O_m$ we distinguish three cases:

$$O_m = \begin{cases} 
0, & \text{w.p. } 1 - P(X_m-1 = 0)g_m \text{ if } g_m = 0 \text{ or } (g_m = 1 \text{ and } d_m = 0), \\
1 - P(X_m-1 = 0) & \text{w.p. } \text{ if } g_m = 1, \\
1, & \text{w.p. } 1 - P(X_m-1 = 0) \text{ if } g_m = 1 \text{ and } d_m = 1.
\end{cases} \quad (8.1)$$

When $g_m = 0$ no vehicle will leave the queue in time slot $m$. When $g_m = 1$ and the queue $X_m$ is empty, all the arriving vehicles will immediately pass through by assumption 3 independent of the value of $d_m$. When the queue $X_{m-1}$ is not empty, there will be one vehicle that leaves the system if $d_m = 1$ and there will be no vehicle that leaves the system if $d_m = 0$. This gives the following probability generating function for the output process of slot $m$.

$$O_m(z) = 1 + (z - 1)(1 - q_{m-1})d_m + (Y_m(z) - 1)q_{m-1}g_m. \quad (8.2)$$

These output processes are quite easy to calculate, the only thing really needed that might take some time are the $q_i$’s we already described in Chapter 4. These parameters are as easy to find as finding the roots of the denominator of equation (4.12) in the same chapter.

In a system of intersections, the output processes can be used to construct the arrival processes for other intersections. For instance when the two intersections are close together one could
just use the output process of the first intersection as arrival process for the next intersection. Possibly multiplied with the output process of another stream of vehicles from the previous intersection when such a stream exists. When the intersections are not that close together, the output process can probably not be used that strictly in the existing intervals, because the vehicles probably will not drive all with the same speed and therefore have different travel times in between the intersections. So it might be necessary to transform these output processes in some way before using them as arrival processes of the next intersection. Furthermore we assumed that the arrival processes are independent and due to that assumption the output processes are independent, while in practice they probably are dependent in some way, there we do not know how dependent they are, we do not know how much influence this would have. Therefor it might be an option to construct something similar without that assumption and to work with a joint probability distribution instead. This could be an interesting subject for further research.

8.3 Delay in a Network

At the moment we can already analyze many things. We can calculate the probability generating function of the queue length at the end of every time slot of an intersection as well as the probability generating function of the delay at an intersection. Besides that we can construct an output process for every intersection and use this in some way as the arrival process for other intersection, but at the moment we can not analyze the delay of an arbitrary vehicle in a network of intersections. In this section we will construct a way to calculate the probability generating function of the delay in a network of two intersections and we will explain as well how we can use this to calculate the probability generating function of the delay in a network of any given number of intersections.

Before we can start constructing the pgf of the delay of a network of two intersections, we need a definition for the delay in a network of intersections.

**Definition 2** (Delay in a Network of Intersections). *The delay $D_N$ of a vehicle in a network of intersections is defined as the sum of the number of time slots a vehicle is delayed at everyone of the intersections of the network.*

Furthermore we assume that a vehicle that leaves the queue of the first intersection at the end of the $m$-th time slot, will arrive at the second intersection during the $(m + j \mod c)$-th time slot, where $j \in \{0, 1, ..., c - 1\}$. Similar as in Chapter 5 we start by tagging a vehicle that arrives in the $m$-th time slot. Let now $D_N[m]$ denote the delay in the network of the tagged vehicle and let us define the event

$$E_{m,k} := \{\text{The tagged vehicle in time slot } m \text{ arrives at the second intersection during time slot } k\}.$$

We say that

$$D_N[m] = \begin{cases} D^{(1)}[m] E_{m,1} + D^{(2)}[m] & \text{when } E_{m,1}, \\ \vdots & \vdots \\ D^{(1)}[m] E_{m,k} + D^{(2)}[m] & \text{when } E_{m,k}, \\ \vdots & \vdots \\ D^{(1)}[m] E_{m,c} + D^{(2)}[m] & \text{when } E_{m,c}, \end{cases}$$

where $D^{(1)}[m]$ is the delay in the first intersection of a vehicle tagged in the $m$-th time slot of the first intersection and $D^{(2)}[m]$ is the delay in the the second intersection of a vehicle tagged in the
\(m\)-th time slot of the second intersection. The probability generating function of these random variables can easily be calculated by the method from Chapter 5. We only need to find a way to find the probability generating function of the random variable \(D^{(1)}_{m,k}\), which is the delay of a tagged vehicle in the \(m\)-th time slot given that the tagged vehicle will arrive at the second intersection during the \(k\)-th time slot.

We know that we can write \(D^{(1)}_{m}\) in the form \(\sum_{j=0}^{\infty} p_{m,i}^{(1)} z^j\), where \(p_{m,i}^{(1)}\) are just probabilities that sum to one. Suppose now that the delay of the tagged vehicle is \(i\). Knowing that it arrived in time slot \(m\), we know that it will depart the queue of the first intersection at the end of time slot \((m + i + j) \mod c\), but we know it arrives during the \(k\)-th time slot, therefore we know that the only possible delays are the ones such that the following equation holds for \(i\).

\[
k \equiv (m + i + j) \mod c. \tag{8.3}
\]

This happens when \(i = (k - m - j) + sc, s \in \mathbb{N}\). Therefore we have that when \(k \geq m + j\)

\[
\left( D^{(1)}_{m} \mid E_{m,k} \right) (z) = \sum_{j=0}^{\infty} p_{m,(k-m-j)+jc}^{(1)} z^{(k-m-j)+jc}.
\]

When \(k < m + j\) the first value of \(s\) needs to be larger such that \(0 \leq (k - m - j) + sc < c\). For simplicity we just define \(p_{m,i}^{(1)} := 0\) for every \(i < 0\). Notice that the event \(E_{m,k}\) happens with probability \(\sum_{j=0}^{\infty} p_{m,(k-m-j)+jc}^{(1)}\). Using this we can find the probability generating function of \(D_{N[m]}\) to be

\[
D_{N[m]}(z) = \sum_{k=1}^{c} D_{[k]}^{(2)}(z) \left( D^{(1)}_{m} \mid E_{m,k} \right) (z) \mathbb{P}(E_{m,k})
\]

\[
= \sum_{k=1}^{c} D_{[k]}^{(2)}(z) \sum_{j=0}^{\infty} p_{m,(k-m-j)+jc}^{(1)} z^{(k-m-j)+jc}. \tag{8.4}
\]

Similar as in \(5\) we can now find the probability generating function of the delay of an arbitrary vehicle in a network.

\[
D_{N}(z) = \sum_{m=1}^{c} \frac{\mathbb{E}Y^{(1)}_{m} D_{N[m]}(z)}{\sum_{m=1}^{c} \mathbb{E}Y^{(1)}_{m}}. \tag{8.5}
\]

When calculating this delay in practice one needs to replace infinity in Equation (8.4) by some integer number, such that the probabilities \(p_{m,(k-m-j)+jc}^{(1)}\) we discard are significantly small.

If we now want to know the delay of an arbitrary vehicle in a network of \(n\) intersections, we simply use the above method \((n - 1)\) times. First calculate the delay of an arbitrary vehicle the network consisting of the first two intersection. We then simply see the network of the first two intersections as one new first intersection and redefine \(D^{(1)}_{m} := D_{N[m]}(z)\) for every \(m\) and \(D^{(1)}_{m} := D_{N[m]}^{(1)}\). You could say we now have an network of \((n - 1)\) intersections where the first two intersections are merged. We continue this process until all intersections are merged. We can then simply use formula (8.5) to calculate the probability generating function of the delay of an arbitrary vehicle throughout the complete network, by using the \(D_{N[m]}(z)\) found in the last step.

### 8.4 Network of identical intersections

In this section we will analyze what happens when we have a network of multiple intersections with identical settings. Does the output process converge? We will use some arbitrary probability
generating function $Y(z)$ as arrival process for every time slot at the first intersection. We assume that in these networks the travel time from one intersection to another is constant, which means that the output process of one intersection can be used exactly as arrival process for the next intersection. In the case of three time slots we will have the following arrival process.

\[
Y_1^0(z) = Y(z), \\
Y_2^0(z) = Y(z), \\
Y_3^0(z) = Y(z).
\]

### 8.4.1 Cycle length of three

In this section we will analyze what happens in a cycle of length $c = 3$ and a green period of length 2.

**No shift**

First we will use the output process of the $i$-th slot as the arrival process for the $i$-th slot of the next intersection.

\[
O_0^1(z) = z(1 - q_0) + Y(z)q_0, \\
O_2^0(z) = z(1 - q_1) + Y(z)q_1, \\
O_3^0(z) = 1,
\]

\[
Y_1^1(z) = z(1 - a_0) + Y(z)a_0, \\
Y_2^1(z) = z(1 - a_1) + Y(z)a_1, \\
Y_3^1(z) = 1.
\]

Given the arrival process we can use the result of chapter [8.2] to find the output process of the first intersection. These processes we use as arrival processes for the next intersection. For simplicity we introduce for every arrival process new parameters. For example $a_0 := q_0$. The parameters $q_0$ and $q_1$ are the parameters as introduced in chapter [4]

\[
O_0^1(z) = z(1 - q_0) + Y(z)q_0, \\
O_0^2(z) = z(1 - q_0a_0) + Y(z)a_0q_0, \\
O_3^0(z) = 1,
\]

\[
Y_1^1(z) = z(1 - a_0) + Y(z)a_0, \\
Y_2^1(z) = z(1 - a_1) + Y(z)a_1, \\
Y_3^1(z) = 1.
\]

Similar we construct the output processes for the second intersection and use these for the arrival process of the third intersection. Again we use new parameters for simplicity, for example $b_0 := q_0a_0$.

\[
O_1^1(z) = z(1 - q_0) + Y(z)q_0 \\
= z(1 - q_0a_0) + Y(z)a_0q_0, \\
O_2^1(z) = z(1 - q_1) + Y(z)q_1 \\
= z(1 - q_1a_1) + Y(z)a_1q_1, \\
O_3^1(z) = 1,
\]

\[
Y_1^2(z) = z(1 - b_0) + Y(z)b_0, \\
Y_2^2(z) = z(1 - b_1) + Y(z)b_1, \\
Y_3^2(z) = 1.
\]

We notice that after the first intersection the output process is of the form $z(1 - a) + Y(z)a$ for a green slot and it stays of this form after that. We will now calculate the values of $q$ for such an
arrival process to see if the output process converges.

\[ Y_1(z) = z(1 - a_0) + Y(z)a_0, \]
\[ Y_2(z) = z(1 - a_1) + Y(z)a_1, \]
\[ Y_3(z) = 1. \]

First we calculate the roots of the numerator of \( X_0(z) \):

\[
0 = 1 - \frac{a_0 Y(z) + (1 - a_0)z a_1 Y(z) + (1 - a_1)z}{z} = \left( a_0 + a_1 - a_0a_1 + \frac{a_0a_1 Y(z)}{z} \right) \left( 1 - \frac{Y(z)}{z} \right).
\]

For the specific case that \( Y(z) \) is exponential this leads to the roots:

\[
z = \begin{cases} 
1, \\
- \frac{-W \left( \frac{a_0a_1 e^{-\lambda z}}{a_0 + a_1 - a_0a_1} \right)}{\lambda}
\end{cases},
\]

where \( W(\theta) \) is the Lambert \( W \) function, which is defined as the inverse function of \( f(W) = We^W \).

By using the root \( z = \frac{-W \left( \frac{a_0a_1 e^{-\lambda z}}{a_0 + a_1 - a_0a_1} \right)}{\lambda} \) we can construct an equation for our system of equations as shown in chapter 4.

\[
0 = q_0a_0 \left( a_1 e^{\frac{-W \left( \frac{a_0a_1 e^{-\lambda z}}{a_0 + a_1 - a_0a_1} \right)}{\lambda} - 1} \right) + q_1a_1 \frac{-W \left( \frac{a_0a_1 e^{-\lambda z}}{a_0 + a_1 - a_0a_1} \right)}{\lambda}.
\]

The second equation is constructed by setting the limit of \( X_0(z) \) to one to one. By using l’Hopital we find:

\[
1 = \lim_{z \to 1} X_0(z) = \frac{a_0q_0 + a_1q_1}{a_0 + a_1}.
\]

Solving this system of equations to \( (q_0, q_1) \), one can find that \( q_0 = 1 \) and \( q_1 = 1 \), which implies that for an Poisson arrival process the output process will be of the form \( z(1 - a) + Y(z)a \) for the green slots and 1 for the red slot for every intersection and this for in a system as described above. The values for \( a \) can shown to be \( \frac{W \left( \frac{n}{\lambda} \right) - \frac{n}{\lambda}}{W \left( \frac{n}{\lambda} \right) + \left( \frac{n}{\lambda} \right)} \) for the first green slot and

\[
\frac{\frac{3n}{\lambda} - \frac{2n}{\lambda}}{\lambda - 1} - \frac{W \left( \frac{3n}{\lambda e} - \frac{3n}{\lambda} \right)}{W \left( \frac{3n}{\lambda e} - \frac{3n}{\lambda} \right) + \left( \frac{3n}{\lambda} \right)} \frac{1}{\lambda - 1} \] for the second green slot.

32 Networks of Fixed-Cycle Traffic-Lights
Shift of one time slot

In this subsection the output process of the $i$-th time slot of the $l$-th intersection is the arrival process for the $(i+1)$-th time slot of the $(l+1)$-th intersection.

\begin{align*}
O_0^0(z) &= z(1-q_0) + Y(z)q_0, & Y_1^1(z) &= 1, \\
O_0^1(z) &= z(1-q_1) + Y(z)q_1, & Y_2^2(z) &= z(1-a_0) + Y(z)a_0, \\
O_0^2(z) &= z(1-q_2) + Y(z)q_2, & Y_3^3(z) &= z(1-a_1) + Y(z)a_1, \\
O_0^3(z) &= 1, & Y_4^4(z) &= 1, \\
O_1^0(z) &= z(1-q_0) + Y_1(z)q_0 & & Y_2^2(z) &= z(1-b_0) + b_0, \\
O_1^1(z) &= z(1-q_1) + Y_2(z)q_1 & & Y_3^3(z) &= z(1-b_1) + Y(z)b_1, \\
O_1^2(z) &= z(1-q_2) + Y_3(z)q_2 & & Y_4^4(z) &= 1, \\
O_1^3(z) &= 1, & Y_3^3(z) &= z(1-c_0) + c_0, \\
O_2^0(z) &= z(1-q_0) + Y_1(z)q_0 & & Y_3^3(z) &= z(1-c_1) + c_1, \\
O_2^1(z) &= z(1-q_1) + Y_2(z)q_1 & & Y_4^4(z) &= 1, \\
O_2^2(z) &= z(1-q_2) + Y_3(z)q_2 & & Y_2^2(z) &= z(1-d_0) + d_0, \\
O_2^3(z) &= 1, & Y_3^3(z) &= z(1-d_1) + d_1, \\
O_3^0(z) &= z(1-q_0) + Y_1(z)q_0 & & Y_3^3(z) &= z(1-d_1) + d_1, \\
O_3^1(z) &= z(1-q_1) + Y_2(z)q_1 \text{ for a green slot and it stays of this form after that. We will now calculate the values of } q \text{ for such an arrival process to see if the output process converges.}
\end{align*}

We notice that after the third intersection the output process is of the form $z(1-a) + a$ for such an arrival process to see if the output process converges.
First we calculate the roots of the numerator of $X_0(z)$:

\begin{align*}
0 &= z^2 - (a_0 + (1 - a_0)z)(a_1 + (1 - a_1)z) \\
&= (z(a_0 + a_1 - a_0 a_1) + a_0 a_1)(z - 1).
\end{align*}

This leads to the roots:

\[ z = \left\{ 1, \frac{-a_0 a_1}{a_0 + a_1 - a_0 a_1} \right\}. \]

By using the root $z = \frac{-a_0 a_1}{a_0 + a_1 - a_0 a_1}$ we can construct an equation for our system of equations as shown in chapter 4:

\[ 0 = q_0(a_0 + (1 - a_0) \frac{-a_0 a_1}{a_0 + a_1 - a_0 a_1}) + q_1 a_0 \frac{-a_0 a_1}{a_0 + a_1 - a_0 a_1}. \]

The second equation is constructed by setting the limit of $X_0(z)$ to one to one. By using l’Hopital we find:

\[ z = \lim_{z \to 1} X_0(z) = \frac{q_0 + a_0 q_1}{a_0 + a_1}. \]

Solving this system of equations to $\{q_0, q_1\}$, one can find that $q_0 = a_1$ and $q_1 = 1$, which implies that during one intersection $\{a_0, a_1\}$ will be transformed to $\{a_1, a_0\}$. We finally see that this systems converges to an alternating sequence where the output of the two green periods switches.

Shift of two time slots

In this subsection the output process of the $i$-th time slot of the $l$-th intersection is the arrival process for the $(i+2)$-th time slot of the $(l+1)$-th intersection.

\[ O_0^i(z) = z(1 - q_0) + Y(z)q_0, \quad Y_1^i(z) = z(1 - a_1) + Y(z)a_1, \]
\[ O_2^i(z) = z(1 - q_1) + Y(z)q_1, \quad Y_2^i(z) = 1, \]
\[ O_3^i(z) = 1, \quad Y_3^i(z) = z(1 - a_0) + Y(z)a_0, \]

34 Networks of Fixed-Cycle Traffic-Lights
First we calculate the roots of the numerator of $X_0(z)$:

$$\begin{align*}
O_1^1(z) &= z(1-q_0) + Y_1(z)q_0 \\
&= z(1-q_0a_1) + Y(z)q_0a_1, \\
O_2^1(z) &= z(1-q_1) + Y_2(z)q_1 \\
&= z(1-q_1) + q_1, \\
O_3^1(z) &= 1,
\end{align*}$$

$$\begin{align*}
Y_1^2(z) &= z(1-b_1) + b_1, \\
Y_2^2(z) &= z(1-b_0) + Y(z)b_0, \\
Y_3^2(z) &= z(1-b_0) + Y(z)b_0.
\end{align*}$$

We notice that after the third intersection the output process is of the form $z(1-a) + a$ for a green slot and it stays of this form after that. We will now calculate the values of $q$ for such an arrival process to see if the output process converges.

$$\begin{align*}
Y_1(z) &= z(1-a_1) + a_1, \\
Y_2(z) &= 1, \\
Y_3(z) &= z(1-a_0) + a_0.
\end{align*}$$

First we calculate the roots of the numerator of $X_0(z)$:

$$\begin{align*}
0 &= z^2 - (a_0 + (1-a_0)z)(a_1 + (1-a_1)z) \\
&= (z(a_0 + a_1 - a_0a_1) + a_0a_1)(z-1). \\
\end{align*}$$

This leads again to the roots:

$$z = \left\{ 1, \frac{-a_0a_1}{a_0 + a_1 - a_0a_1} \right\}.$$

By using the root $z = \frac{-a_0a_1}{a_0 + a_1 - a_0a_1}$ we can construct an equation for our system of equations as shown in chapter 4.
\[ 0 = q_0 a_1 + q_1 \frac{-a_0 a_1}{a_0 + a_1 - a_0 a_1}. \]

The second equation is constructed by setting the limit of \( X_0(z) \) to one to one. By using l’Hôpital we find:

\[ z = \lim_{z \to 1} X_0(z) = \frac{0}{a_0 + a_1}. \]

Solving this system of equations to \( \{q_0, q_1\} \), one can find that \( q_0 = a_0 \) and \( q_1 = a_0 + a_1 - a_0 a_1 \), which implies that during one intersection \( \{a_0, a_1\} \) will be transformed to \( \{a_0 a_1, a_0 + a_1 - a_0 a_1\} \). By induction it can be shown that the output process will converges.

### 8.4.2 Arbitrary cycle length

Suppose now we have a cycle of length \((g + r)\) and a green period of length \(g\), which implies a red period of length \(r\) and let us consider a shift of \(k\) time slots. We state that the following expression is true if and only if at some finite intersection the output processes are reduced to only Bernoulli processes.

\[
\text{lcm}(\min(k, g + r - k), g + r) r \geq \min(k, g + r - k)(g + r). \tag{8.10}
\]

**Proof.** First notice that shifting \(k\) is equivalent to shifting \(g + r - k\) by the fact that shifting to the left or to the right is symmetric for the output process. Furthermore notice that an output process becomes Bernoulli if and only if the arrival process was Bernoulli or the traffic light was red for the corresponding traffic light. From this we can deduce that if the arrival process for the first intersection was not Bernoulli for every time slot and the output process is Bernoulli for the \( n \)-th time slot after the \( n \)-th intersection, the \((i)\)-th time slot had to be red somewhere along the way.

Suppose now that we have that \( \text{lcm}(\min(k, g + r - k), g + r) r < \min(k, g + r - k)(g + r) \). This is equivalent to \( \frac{\text{lcm}(\min(k, g + r - k), g + r) r}{\min(k, g + r - k)} < (g + r) \). Notice now that \( \frac{\text{lcm}(\min(k, g + r - k), g + r) r}{\min(k, g + r - k)} \) is the number of shifts until the red slots are back on their beginning position. From this it follows that the left-hand side of the equation above represents the number of time slots that have been red during the \( k \) shifts of \( k \) time slots, or equivalent the number of time slots that will become red at some point. By the equation above we know that that number is less than \((g + r)\), which implies that there are slots that never become red. From this we know that when \( \text{lcm}(\min(k, g + r - k), g + r) r < \min(k, g + r - k)(g + r) \) we do not have that at some finite intersection the output processes are reduced to only Bernoulli processes.

Suppose now that \( \text{lcm}(\min(k, g + r - k), g + r) r \geq \min(k, g + r - k)(g + r) \) and suppose that \( \text{lcm}(\min(k, g + r - k), g + r) = g + r \). From this it follows that \( r \geq \min(k, g + r - k) \). From this it easily follows that by shifting by \( \min(k, g + r - k) \) the red periods of consecutive intersections overlap. So it follows that all time slots become red at some finite time, which implies that the output process will become a Bernoulli process for every time slot after some finite time.

Suppose now that \( \text{lcm}(\min(k, g + r - k), g + r) r \geq \min(k, g + r - k)(g + r) \) and suppose that \( \text{lcm}(\min(k, g + r - k), g + r) > g + r \). Like before \( \frac{\text{lcm}(\min(k, g + r - k), g + r) r}{\min(k, g + r - k)} \) is the number of shifts until the red slots are back on their beginning position. So we can be sure that there are at least enough red slots as needed to cover all the slots. But suppose now that by an overlap of two red periods some time slot will never be covered. Well suppose that the red period of the \(i\)-th intersection overlaps with the red period of the \((i + l)\)-th intersection. Then we now that the red period of the \((i + l)\)-th intersection overlaps again with the red period of the \((i + l)\)-th intersection.
at the other side. Furthermore if \((i + 2l)\) is more than the number of shifts until we are back at the starting point, we can calculate it modulo that number of shifts and find an intersection number with the same red period. So we can use this method to cover every slot by a red period and get to a contradiction. We find that it is not possible that by an overlap of two red periods some time slot will never be covered.

When the condition above is not met, not every time slot will become red at some finite time, but all the time slots who do will go to a Bernoulli distribution as output process and will stay in this way. The other time slots will be of the form \((1 - q_{m-1})z + q_{m-1}Y_m(z)\) where \(Y_m(z)\) is of the same form, going back until \(Y_m(z) = Y(z)\), the original arrival process.

In the special case where we have no shift between the different intersections, we

Suppose that at some point the output process will be of the form \(z(1 - a) + a\) for a green slot. From that step on the process will stay of that form for every type of shift.

**Conjecture 1.** Let \(a_i\) be the coefficient such that \(z(1 - a_i) + a_i\) is the output process of the \(i\)-th green slot and let \(b_{k,l}\) be the output of the \(l\)-th green period at the next iteration, given that the \(i\)-th slot is send to the \((i + k)\)-th slot. Then the following formulas find the \(b_{k,l}\) in function of the \(a_i\)'s.

\[
\begin{align*}
\begin{cases}
  a_i - k & \text{if } k < l \leq g, \\
  1 - \prod_{i=g-k+1}^{g-1}(1-a_i) & \text{if } k = l \leq g, \\
  a_{g-k+l+1} \left[1 - \prod_{i=g-k+1}^{g-l}(1-a_i)\right] & \text{if } g \geq k > l,
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\begin{cases}
  1 - \prod_{i=1}^{g-1}(1-a_i) & \text{if } k > l = g,
  a_{i+1} \left[1 - \prod_{i=1}^{g-l}(1-a_i)\right] & \text{if } k > g > l.
\end{cases}
\end{align*}
\]

**Argument.** For every \(g, r, k\) and \(l\), satisfying (8.10), and which we tested with computer software, the formulas above holds.

The following table gives the \(b_{k,l}\) in a matrix form.

<table>
<thead>
<tr>
<th>(k)</th>
<th>(1)</th>
<th>(\ldots)</th>
<th>(\ldots)</th>
<th>(g)</th>
<th>(\ldots)</th>
<th>(g + r - 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(a_g)</td>
<td>(a_g a_{g-1})</td>
<td>(\ldots)</td>
<td>(a_2 a_1)</td>
<td>(\ldots)</td>
<td>(a_2 a_1)</td>
</tr>
<tr>
<td>2</td>
<td>(a_1)</td>
<td>(1 - (1 - a_g)(1 - a_{g-1}))</td>
<td>(\ldots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
</tr>
<tr>
<td>(l)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\ldots)</td>
<td>(a_g [1 - \prod_{i=1}^{g-1}(1-a_i)])</td>
<td>(\ldots)</td>
<td>(a_g [1 - \prod_{i=1}^{g-1}(1-a_i)])</td>
</tr>
<tr>
<td>(g)</td>
<td>(a_{g-1})</td>
<td>(\ldots)</td>
<td>(a_1)</td>
<td>(1 - \prod_{i=1}^{g-1}(1-a_i))</td>
<td>(\ldots)</td>
<td>(1 - \prod_{i=1}^{g-1}(1-a_i))</td>
</tr>
</tbody>
</table>

Table 8.1: \(b_{k,l}\) in matrix form

37 Networks of Fixed-Cycle Traffic-Lights
8.5 Simple Example of a Network

In this section we will consider an example of an network of intersections. In fact we have two intersections, the first one with two traffic lights and the second one with one, we will call them traffic light 1, 2 and 3. Vehicles that leave the queue of traffic light 1 or 2 will join the queue of traffic light 3. The situation is illustrated in Figure 8.5. In this section we will use indices \(i\) to indicate that we mean traffic light \(i\).

![Figure 8.5: Situation of the network examples](image)

The cycle length for all the traffic lights is \(c = 10\). Traffic light 1 has for every time slot a Poisson arrival process with arrival rate \(\lambda = 0.25\). The first four time slots are green. Traffic light 2 has as well Poisson arrival process, but with arrival rate \(\lambda = 0.08\). The seventh and eighth time slot are green for this traffic light. The arrival processes of traffic light 3 depend on the output processes of traffic light 1 and 2. We assume that the travel time from traffic light 1 and 2 till traffic light 3 is that small, that all departing vehicles at the end of the \(m\)-th time slot arrive during the next time slot at traffic light 3. According to the equations in Section 8.2 we have \(j = 1\). Threfor the arrival process of the \(m\)-th time slot of traffic light 3 is equal to the output processes of the \((m - 1)\)-th time slot of traffic light 1 and 2 multiplied. The traffic light is red during time slot 4 till 8 and green in the remaining ones.

We see that the different traffic lights have different number of lanes. We will use the variables \(d_i\) to take this into account. For all green time slots we set the \(d_i\) to one in a ratio of the number of lanes divided by four. Two lanes means alternating between one and zero. By assumption 4 we make sure that the last green time slot has \(d_i = 1\). In Figure 8.6 it is shown what this means for all of the traffic lights.

---

Networks of Fixed-Cycle Traffic-Lights
In Table 8.2 we show the different arrival processes for the different traffic lights. \( Y^{(1)} \) the arrival process for every time slot of traffic light 1, \( Y^{(2)} \) as the arrival process for traffic light 2 and \( Y^{(3)} \) the arrival processes for traffic light 3, the coefficients rounded to four digits accuracy. As earlier mentioned these arrival processes \( Y_i^{(3)} = O_{i-1}^{(1)} + O_{i-1}^{(2)} \) depend on the output processes of traffic light 1 and 2, with \( O_i^{(1)} \) and \( O_i^{(2)} \) the output processes of the \( i \)-th time slot of respectively traffic light 1 and 2. Figure 8.7a shows the distribution of the queue length at the end of the red period of traffic light 3. We see that the probability that the queue length is more than three is almost zero. Most of the times at the end of the red period there will be one or two vehicles waiting in the queue. Figure 8.7b shows the distribution of the queue length at the end of the green period. Here we see that with a probability of 0.998 there will be no vehicles waiting in the queue. It is not that clear to see but the other queue lengths are not completely zero but extremely small. The expected queue length at the end of the different time slots is shown in Figure 8.7c. We see that as expected the queue length starts decreasing at time slot 9, the first green time slot. At the end of time slot 3 the expected queue length is significantly small and starts increasing after that because of the red traffic light. during time slot 6 and 7 the expected queue length does not increase because no vehicles depart from the previous traffic lights and therefore no vehicles will arrive at traffic light 3. This can be seen at the values of \( Y_6^{(3)} \) and \( Y_7^{(3)} \). In Figure 8.7d we see the

<table>
<thead>
<tr>
<th>( d_i )</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_i )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
<td>( y_1^{(1)} )</td>
</tr>
</tbody>
</table>

(a) Settings of traffic light 1

<table>
<thead>
<tr>
<th>( d_i )</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_i )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
<td>( y_1^{(2)} )</td>
</tr>
</tbody>
</table>

(b) Settings of traffic light 2

<table>
<thead>
<tr>
<th>( d_i )</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_i )</td>
<td>( y_1^{(3)} )</td>
<td>( y_2^{(3)} )</td>
<td>( y_3^{(3)} )</td>
<td>( y_4^{(3)} )</td>
<td>( y_5^{(3)} )</td>
<td>( y_6^{(3)} )</td>
<td>( y_7^{(3)} )</td>
<td>( y_8^{(3)} )</td>
<td>( y_9^{(3)} )</td>
<td>( y_{10}^{(3)} )</td>
</tr>
</tbody>
</table>

(c) Settings of traffic light 3

Figure 8.6: Settings of the traffic lights

\[
\begin{align*}
Y^{(1)} &= e^{\frac{z}{3}(z^{-1})} \\
Y_1^{(3)} &= 1 \\
Y_3^{(3)} &= 0.8903z + 0.1097e^{\frac{25}{3}(z^{-1})} \\
Y_5^{(3)} &= 0.6346z + 0.3654e^{\frac{25}{3}(z^{-1})} \\
Y_7^{(3)} &= 1 \\
Y_9^{(3)} &= \frac{16}{21}z + \frac{5}{21}e^{\frac{z}{3}(z^{-1})} \\
Y^{(2)} &= e^{\frac{z}{3}(z^{-1})} \\
Y_2^{(3)} &= 0.8903 + 0.1097e^{\frac{25}{3}(z^{-1})} \\
Y_4^{(3)} &= 0.7719z + 0.2281e^{\frac{25}{3}(z^{-1})} \\
Y_6^{(3)} &= 1 \\
Y_8^{(3)} &= \frac{16}{21}z + \frac{5}{21}e^{\frac{z}{3}(z^{-1})} \\
Y_{10}^{(3)} &= 1
\end{align*}
\]

Table 8.2: Arrival processes of the different traffic lights
distribution of the delay of traffic light 3. In Table 3.3c we show its expected delay and variance. Furthermore we analyzed the delay at the other traffic lights and the delay through the network of an arbitrary vehicle arriving at traffic light 1 and 2. This can be seen in Tables 3.3a and 3.3b. Obviously for an arbitrary arriving vehicle at traffic light 1 and 2 the expected delay in the network is more than the expected delay at the first traffic light of the intersection. For a vehicle that starts at traffic light 1 the difference is a bit more than three time slots, there it is only about 1.5 more for a vehicle that started at intersection 2. Therefore we can conclude that the settings of traffic light 3 are more in favor of those vehicles. In the contrary those vehicles have a much longer delay at their first intersection. So the vehicles of traffic light 2 are expected to be the longest in the network, but mainly due to traffic light 2. Furthermore we notice that we have, as expected, the equality

$$E_D(3) = \frac{EY(1)}{EY(3) + EY(3)} (ED(1) - ED(2)) + \frac{EY(1)}{EY(3) + EY(3)} (ED(2) - ED(2)).$$

This equality holds in general due to the definition of the delay in a network, however it only holds for the expectation of the delay, not for the delay itself. When the arrival processes are different for every time slot the equation has to be changed a bit in the fractions.
Finally we show the distribution of the delay in the network for an arbitrary arriving vehicle at traffic light 1 and 2 in respectively Figure 8.8a and 8.8b. We see that in Figure 8.8a that there is a significantly low probability (0.0001788) of a network delay of one for an arbitrary arriving vehicle at traffic light 1. This is due to the fact there is only one possibility that a vehicle has a delay of one time slot in the network. That is when he arrives at traffic light 1 during time slot 1 and finds an empty queue, which means he can immediately pass through by Assumption 3, and therefore arrives at traffic light 3 during time slot 2 and finds one waiting vehicle in the queue and he can leave at the end of time slot 3. For a vehicle arriving at traffic light 2 there is only one possibility to have no delay in the network. This is when he arrives during time slot 8, finds an empty queue, therefore arrives during time slot 9 at traffic light 3 and finds again an empty queue. In Figure 8.8b we can see that there is a small probability (0.001179) of no delay in the network.

Table 8.3: Overview of the different expected delay and variance of the delay

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D^{(1)}$</td>
<td>10.4586</td>
<td>91.7811</td>
</tr>
<tr>
<td>$D_N^{(1)}$</td>
<td>13.5321</td>
<td>93.9038</td>
</tr>
<tr>
<td>$D^{(2)}$</td>
<td>24.2857</td>
<td>542.5533</td>
</tr>
<tr>
<td>$D_N^{(2)}$</td>
<td>25.8866</td>
<td>541.5750</td>
</tr>
</tbody>
</table>

(a) Arbitrary vehicle arriving at traffic light 1

(b) Arbitrary vehicle arriving at traffic light 2

(c) Arbitrary vehicle arriving at traffic light 3
(a) Delay distribution in the network for an arbitrary vehicle from traffic light 1

(b) Delay distribution in the network for an arbitrary vehicle from traffic light 2
8.6 Realistic Example of a Network

In this section we will consider a realistic example of a network of intersections. The situation of the network is the same as in the previous section, but now the cycle length is of a more realistic size, in this situation $c = 77$. For an illustration of the situation one can go back to Figure 8.5. In practice one could find such a network at the exit of a highway, which would be traffic light 2 in our situation. In practice there would be more traffic streams, but these streams would not influence this streams and therefore they can be analyzed separately.

At traffic light 1 there arrive on average 1360 vehicles in an hour, at traffic light 2 this are 460 vehicles in an hour. The green period of traffic light 1 starts at time slot 5 and ends at time slot 33. This means that we have 29 green time slots, but only three of every four time slots will have $d_i = 1$ due to the fact we have three lanes. Traffic light 2 has a green period of length 20, starting at time slot 56. Due to the fact it has two lanes only one of every two green time slots has $d_i = 1$. Finally traffic light 3 its green period starts at time slot 8 and continues for 30 time slots. Here we have that for every green time slot $d_i = 1$, there we have four lanes. We consider one time slot to refer to half a second. All this is shown in Figure 8.9.

In Table 8.4 we show a part of the different arrival processes for the different traffic lights. $Y^{(1)}$ the arrival process for every time slot of traffic light 1, $Y^{(2)}$ as the arrival process for traffic light 2 and the first ten $Y^{(3)}$'s, the arrival processes for traffic light 3 with the coefficients rounded to four digits accuracy. Like in the previous section these arrival processes $Y^{(3)} = O_{i-1}^{(1)} + O_{i-1}^{(2)}$ depend on the output processes of traffic light 1 and 2, with $O_{i}^{(1)}$ and $O_{i}^{(2)}$ the output processes of the $i$-th time slot of respectively traffic light 1 and 2.

Analyzing these traffic lights we can find the expected delay of all these traffic lights as well as the expected queue length at the end of every time slot of all these traffic lights. The expected delays are shown in Table 8.5. In Figure 8.10 we show the expected queue length at the end of every time slot for traffic light 3 in the vertical axis and the time slot number $m$ in the horizontal axis. We see that the expected queue length decreases during the green period, time slot 8 till 37, and increases during the other time slots. Furthermore we see that during the green period of traffic light 2. There is alternating a big and a small step. This due to the fact that traffic light 2 has two lanes and therefore has $d_i$ alternating between one and zero. We see as well that from time slot 77 till 5 the expected queue length stays the same, because traffic light 3 is red, which...
\[ Y^{(1)} = e^{\frac{1360}{7200} (z-1)} \]
\[ Y_1^{(3)} = 1 \]
\[ Y_3^{(3)} = 1 \]
\[ Y_5^{(3)} = 1 \]
\[ Y_7^{(3)} = 0.9990 + 0.0010 e^{\frac{1360}{7200} (z-1)} \]
\[ Y_9^{(3)} = 0.9963 + 0.0037 e^{\frac{1360}{7200} (z-1)} \]
\[ Y^{(2)} = e^{\frac{460}{7200} (z-1)} \]
\[ Y_2^{(3)} = 1 \]
\[ Y_4^{(3)} = 1 \]
\[ Y_6^{(3)} = 0.9999z + 0.0001 e^{\frac{460}{7200} (z-1)} \]
\[ Y_8^{(3)} = 0.9990z + 0.0010 e^{\frac{460}{7200} (z-1)} \]
\[ Y_9^{(3)} = 0.9884z + 0.0116 e^{\frac{460}{7200} (z-1)} \]

Table 8.4: Arrival processes of the different traffic lights

means no vehicles will depart, and there will arrive no new vehicles due to the fact that traffic light 1 and 2 are red from time slot 76 till 4.

\[ \mathbb{E}D^{(1)} = 21.1283 \]
\[ \mathbb{E}D^{(2)} = 25.8590 \]
\[ \mathbb{E}D^{(3)} = 8.6902 \]

Table 8.5: Expected delay at the three traffic lights

Next to these characteristics we can as well determine the probability generating functions of the queue length at every time slot and the delay and this for all three traffic lights. Unfortunately we are not able to distract the probability distribution out of these probability generating functions. This because the methods available to us to distract these probabilities are not able do this for such complex probability generating functions as in this example. In Section 9.3 we will explain the methods we used to do this, one possible subject for further research could be to construct a new method that is able to distract the probabilities from a complex probability generating function such as the ones in this example.
Figure 8.10: Expected queue length at traffic light 3
9 Numerical Issues

By using our model described in Chapter 3 and the techniques to analyze networks of intersections described in Chapter 8, there are a few steps that need numerical calculations that might be hard. In this chapter we will show some methods that might be able to do these calculations.

9.1 Root Finding

In all the examples except the ones in Section 8.1 we had to use methods to find the roots of the denominator of Equation (4.12). In some cases it is possible to do this exactly, in other cases we need to approximate this by using some algorithms. In this section we will discuss several possible methods and explain in which situations they can be used.

In this section we will define \( A(z) := \prod_{i=1}^{I} Y_i(z) \), such that the objective becomes to find the roots of \( z^d = A(z) \). By Janssen [4, 5] we know that the roots of this equation lie on the generalized Sezcô curve, which is defined by

\[
\mathcal{S}_{A,d} := \{ z \in \mathbb{C} | |z| \leq 1, |A(z)| = |z|^d \}.
\]

In Figure 9.1 we show three such a Sezcô curves. These curves are in the situation with ten

![Sezcô curves for \( A(z) = e^{10z(1-z)} \), with \( d = 5 \)]

Figure 9.1: Sezcô curves for \( A(z) = e^{10z(1-z)} \), with \( d = 5 \)

time slots and when the arrival processes are all Poisson with arrival rate \( \lambda \). In five time slots it is allowed that a vehicle can depart the queue (\( d = 5 \)). The green, red and blue curve have respectively \( \lambda \) equal to 0.25, 0.375 and 0.45. Notice that all these curves pass through the real valued point 1, which is the trivial root. It is always a solution, because \( Y(1) = 1 \) by definition.
9.1.1 Exact Solution Method

When the function $A(z)$ is not to complex, mathematical software package like Mathematica is able too find the roots of $z^d = A(z)$ exactly. This can be done with the function 'Solve' in Mathematica. Apparently the function $A(z)$ is quite fast to complex for Mathematica. When all the arrival processes are of the same type, say for instance all Poisson or all Geometric, Mathematica can find the roots as long as the number of time slots in which a vehicle can depart is not to large, for instance when all arrival processes have the same Poisson process 'Solve' will not be able to find all the roots for $d > 21$. The function 'NSolve' in Mathematica, which can be used to find the roots with numerical methods does not seem to be able to find the roots when 'Solve' can not handle it anymore.

For the special case were we have that $A(z) = e^{-\lambda(1-z)}$, which means that the arrival process is Poisson with parameter $\lambda$ for every time slot, we can find an exact expression for the roots. Obviously $z^d = A(z)$ is equivalent to $z = w e^{-\frac{d}{\lambda}(1-z)}$, with $w$ the solutions of the equation $w^d = 1$. We claim now that the roots of $z^d = e^{\lambda(1-z)}$ can be given in exact form by

$$z = -\frac{d}{c\lambda} W \left[ -\frac{c\lambda}{d} e^{-\frac{d}{\lambda}} \right].$$

(9.1)

In these equation $W(\theta)$ is again the Lambert W function, which is defined as the inverse function of $f(W) = We^W$. We already used this function in Section 8.4. It is easy to prove that (9.1) is a solution of $z = we^{-\frac{d}{\lambda}(1-z)}$. This can be done by substituting that solution in the equation and after rewriting a bit and using the fact that $W(\theta)e^{W(\theta)} = \theta$. Furthermore it is well known that the solutions of $w^d = 1$ are $w_k = e^{\frac{2\pi i k}{d}}$, for $k = 0, 1, ..., d - 1$. Using these $w_k$'s one can find with (9.1) all the roots of $z^d = A(z)$. Notice that $w_0$ gives the trivial root 1.

9.1.2 Fixed Point Method

In this subsection we will discuss a fixed point method that will converge to all of the roots under certain conditions. This method is already described in detail in Janssen [5], but we will still give the mean results below.

When the function $A(z)$ has no zeros for $|z| \leq 1$, the $d$ roots of $z^d = A(z)$ within the unit circle satisfy the equation $z = wG(z)$, with $w^d = 1$ and $G(z) = A(z)$. It can be shown that the equation $z = wG(z)$ has a unique solution in $|z| \leq 1$ for every feasible $w$. Using the following successive substitution equation, one could try to find the roots.

$$z_k^{(n+1)} = w_k G(z_k^{(n)}), \quad k = 0, 1, ..., d - 1,$$

(9.2)

with $w_k = e^{2\pi i k/d}$ and starting values $z_k^{(0)} = 0$.

In Janssen [5] there is a Lemma that states that when $A(z)$ has no zeros for $|z| \leq 1$ and $|G'(z)| < 1$, the fixed point equation (9.2) will converge to the desired roots. When all arrival processes are Poisson these conditions hold and therefor the fixed point method will find the desired roots. Unfortunately we already encountered several examples where these conditions will not hold. For this method one has to use a stop criteria, the most useful one in this situation would be to stop when $|z_k^{(n+1)} - z_k^{(n)}|$ becomes small enough. Be aware that stopping when $|z_k^{(n+1)} - z_k^{(n)}|$ is smaller than some value, it does not mean that a precision of that value is reached. At that moment one could say that there will be no bigger steps than the chosen value. So if one wants to be quite sure that some precision is reached, one needs to take an even quite smaller stopping value.
9.1.3 Fourier Series Method

In this subsection we will briefly explain another method from Janssen [5]. This method is based on the the Széchenyi curve $\mathcal{S}_{A,d}$, which we defined in the beginning of this section. First notice that the equation $z = wA^2(z)$ can be written as $zA^2(z) = w$. Applying the Lagrange inversion theorem on this equation, we can find that the solution $z_0(w)$ has the power series representation

$$z_0(w) = \sum_{l=1}^{\infty} c_l w^l,$$

for $w$ in a neighborhood of 0 and with

$$c_l = \frac{1}{l!} \left( \frac{d}{dz} \right)^{l-1} A^2(z) \bigg|_{z=0}.$$

Now using again the fact that we have a unique solution within the unit circle for every $w$ that satisfies $w^d = 1$, we can write a formula for every of the $d$ roots.

$$z_k = \sum_{l=1}^{\infty} c_l w_k^l, \quad (9.3)$$

with $w_k$ again equal to $e^{\frac{2\pi ki}{d}}$.

This method will find the roots whenever the coefficients $c_l$ exponentially decay. In practice it is of course not possible to calculate all the terms of (9.3), this would take infinite time. Therefore one needs to cut off the series at some point, therefor we need a stop criteria again, a useful one here would be to stop when the term currently calculating is small enough, there the terms exponentially decay. It does again not mean that a precision of that value is reached. At that moment one could say that there will be no bigger term than the last calculated term. So if one wants to be quite sure that some precision is reached, one needs to stop when an even quite smaller term is reached. This method is able to find the roots more often than the fixed point method, but we still encountered several examples for which this method encounters the problem that the conditions are not satisfied and therefore will not find the roots. For the details of this method we refer again to Janssen [5].

9.1.4 FindRoot in Mathematica

The software package Mathematica has a function called FindRoot. This function is quite good in finding roots with any given precision. The downside of this function is that it finds only one root close to a given start value. Nevertheless we constructed a way to use this function FindRoot to find all the roots of the equation $z^d = A(z)$ and this for every situation that the methods above could find the roots, most of the time even a bit faster. We do this by using the methods above to construct a start value for the FindRoot function, but we will not use the methods above to find the roots with a high precision, we only use it to find a raw approximation of the roots. By using these approximations as a start value FindRoot will be able to find the roots quite fast with any precision we want.

Furthermore we constructed a method, using FindRoot, which can find most of the time all the roots, sometimes a little human help is needed. To do this we need a way to find close enough starting values in every situation, even in the situations that the previous methods were not able to find the roots. We do this by using the function ContourPlot in Mathematica. It constructs a
plot of the Sezcö curve $\delta_{A,d}$ by trying points by a smart algorithm. From this plot we can extract many points that lie on this curve and because we know that all the desired roots lie on this Sezcö curve, we can easily use all these points as starting value. Obviously this would be a time consuming method, there it has to run the function FindRoot for many points. Nevertheless will it find a root close to everyone of the starting values, but of course most roots will be found several times. Therefore we will not use all the point as starting value, but only the ones for which the value of the denominator is in absolute values less than $10^{-\eta}$. For most examples a value of 1 or 2 will work fine for $\eta$. To make this algorithm even faster, one could consider start values that are rather close to be the same and just run FindRoot on one of these start values in that cluster. After plotting the distinct roots found by FindRoot, it is quite easy to see if there are obvious roots missing, there the roots lie on the Sezcö curve in a structured way and symmetric. When we notice by the structure that roots are missing, we can use a smaller value for $\eta$, when necessary one could make $\eta$ less than 0.

When we see that all the roots in the structure are found, it still can happen that some roots are missing, especially when the arrival processes are based on earlier output processes. The roots that are missing are real valued roots lying inside of the curve. Actually they lie on the Sezcö curve as well. Not on the outer curve of the Sezcö curve but on smaller inner curves. These inner curves are most of the times that small that ContourPlot will not find any points on them. To find these real roots we will use again FindRoot with some real starting values within the outer Sezcö curve. These roots will be less than 0 and the closer to 0 the more roots we will find. So we use start values closer together, the closer we get to 0 and we will stop when we found that many distinct real roots such that we found all our $d$ desired roots. In this process it might be necessary to manually help choosing the starting values at some point. This can be done by analyzing the real graph of the denominator.

### 9.1.5 Root Finding Examples

In this section we show for two examples how we find the roots of the equation $z^d = A(z)$. We do this for traffic light 3 of the example in Section 8.5 and traffic light 3 of the example in Section 8.6. To do this we use the method that uses FindRoot in Mathematica, which we described in Subsection 9.1.4. For traffic light 1 and 2 of these examples it is quite easy to find the roots, there the arrival processes are all the same Poisson process, therefor we can just use the exact solution method described in Subsection 9.1.1.

First we plot the Sezcö curves in respectively Figure 9.2a and 9.3a. In Figure 9.2b and 9.3b we show again these Sezcö curves, but now with the value of $z^d - A(z)$ on the vertical axis. We will now select the lowest points on these graphs. In Figure 9.2c and 9.3c we show the Sezcö curves again with these points in blue. We see these points are nicely clustered, therefor we will use one point of each of the clusters as start value for FindRoot. For the example of Section 8.5 we now found all the roots. They are showed on the Sezcö curve in Figure 9.2d. We can also see here how nicely structured they lie on the curve. For the example of Section 8.6 we found all the roots on the red line showed, but we miss some real roots lying inside this curve. However these roots do lie on the actual Sezcö curve but the method used by Mathematica to find points on this curve finds no point near those roots.

In Figure 9.2a we see a very small curve inside the big part of the Sezcö curve, in this situation Mathematica was lucky to find a few points. We can see as well in Figure 9.2b that Mathematica found much less points in that region compared to the other regions. To find the missing real roots we used different real start values inside the curve and less than zero, and the closer to zero the more start values we chose. Doing this we almost found all the roots, we found the last few roots by scanning the real graph of $z^d - A(z)$ for some more roots and used an approximation.
of them as start point. In Figure 9.2d we show now the Sezcō curve with all the roots and we can again notice the structure.

In practice method works quite well, but some improvements can be made. When one can construct a method that can find the roots as fast as this method and in no situation some help choosing the start values is needed, one has a full automatic algorithm that would make optimizing networks of intersections easier.
Figure 9.2: Root finding by Mathematica
Figure 9.3: Root finding by Mathematica

52 Networks of Fixed-Cycle Traffic-Lights
9.2 System of Equations

In Chapter 4 we constructed a system of equations that calculates the values for the \( q_i \)'s from the roots of the denominator of (4.12). For all the examples shown in this thesis we simply use the function ‘Solve’ in Mathematica to solve this system of equalities. We noticed that when this system gets rather large due to the fact we have a quite big cycle length and especially a rather large value for \( d \), the number of time slots in which a vehicle can depart the queue, the \( q_i \)'s at the end of a green period get rather close to 1.

If now furthermore the queue is close to unstable, which means that during one cycle the number of vehicles that arrive on average are almost as much as the number of vehicles that can depart during one cycle, it might happen that mathematica is able to solve the system of equations, but that the results found for the \( q_i \)'s are not all within the range \([0, 1]\) anymore, which is necessary because the \( q_i \) represent the probabilities \( P(X_i = 0) \). Therefore it might be interesting to find a way such that these systems can be solved in these bigger and close to unstable situations as well.

9.3 Inversion of a Probability Generating Function

With our model we can calculate the probability generating function for the queue length at the end of every time slot and the delay. To analyze these probability generating functions, it might be useful to have a method to find the individual probabilities, for instance to know a certain probability or to calculate tail probabilities. One way to do this is by using the method suggested by van Leeuwaarden [9], which is the method already described in Abate [1].

The distribution \( p_0, p_1, \ldots \), can be retrieved from the pgf \( P(z) = \sum_{k=0}^{\infty} p_k z^k \) by the formula

\[
p_k = \frac{1}{2\pi i} \oint_{C_r} \frac{P(z)}{z^{k+1}} dz,
\]

where \( C_r \) is a circle around the origin with radius \( r, 0 < r < 1 \). Abate and White use an approximation \( \hat{p}_k \) for the equation (9.4).

\[
\hat{p}_k = \frac{1}{2kr^k} \sum_{j=1}^{2k} (-1)^j \text{Re} \left( P(re^{i\pi j/2k}) \right),
\]

with an error bound of

\[
|p_k - \hat{p}_k| \leq \frac{r^{2k}}{1 - r^{2k}},
\]

for \( 0 < r < 1, k \geq 1 \). To have accuracy up to the \( \gamma \)th decimal, we let \( r = 10^{-\gamma/2k} \).

We consider another method to be of more value. With the computational power of computers nowadays, it is most of the time faster to calculate the Taylor series of the probability generating functions and simply take the coefficients to be the searched probabilities. In Mathematica the function ‘Series’ is quite good in handling this and almost always much faster than the method above. Unfortunately as seen in Section 8.6 both methods are not able to find the probabilities when the probability generating functions are as complex as in that realistic example. Therefore it would be useful for further research to find a new method which can handle such a probability generating functions.
10 Conclusions and Suggestions for Further Research

Darroch [3] first constructed a model with discrete time slots to find the probability generating function and expected length of the overflow queue and the expected delay. Van Leeuwaarden [9] made it possible to find the probability generating functions for queue length at the end of every time slot and even for more general arrival processes and also the probability generating function for the delay and its expectation.

In this thesis we were able to make some generalizations on their models. We made it possible to have different arrival processes for different time slots, to model multiple green periods in a cycle and to model that less vehicles can depart the queue in certain periods of the green period. This gives us the possibility to analyze much more situations of isolated intersections than before. The model described in Chapter 3 works for every set of arrival processes, given by probability generating functions, as long as it is possible to find the roots of the denominator of (4.12) within some acceptable time. Together with the techniques described in Chapter 8 we found a way to analyze networks of intersections, a way to compute the pgf of the delay and the queue length at every time slot of every traffic light of the network, as well as the pgf of the delay throughout the network.

But for large networks or smaller networks with large intersections, some numerical problems occur. With the methods we use, or at least the way we implemented them, it is sometimes hard to calculate the roots or to distract the probabilities out of the sometimes rater complex pgf’s. Therefore analyzing how to find the roots in an efficient way, seems to be a useful topics for further research. When one could find a way to find these roots quicker, it would be much easier to analyze traffic lights. Furthermore one needs to find a way to solve the system of equations that finds the $q_i$’s out of these roots in bigger and close to unstable systems. But finding a faster method or a faster implementation to deduce the probabilities from the sometimes quite complex probability generating function, would make it possible to interpret the results found in realistic networks of intersections. When one is able to solve these numerical issues in acceptable time, one has a good model to analyze the queue length and the delay in networks of intersections and might even be able to use this model to optimize a network of intersections.

We used the assumption that the arrival processes over the different time slots are independent, one could for instance by using simulations investigate how much influence this assumption has and conclude how acceptable it is to use this assumption in practice. One could also investigate the possibility of contracting a model with joint probability distributions, such that the assumption of independence would not be necessary. Furthermore a few things we stated are not proven yet. For instance Conjecture 1 where we stated for a general cycle length what the Bernoulli output process are when u have Bernoulli processes as arrival process, as well as the prove that iterating that process converges to specific Bernoulli Processes. These things could be an interesting subject for further research as well. An other interesting topic could be to construct a similar model, but with a joint probability distribution to take dependence into account.
References


