TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computer Science

MASTER'S THESIS

Traffic signals

Optimizing and analyzing traffic control systems

by

M.S. van den Broek

Eindhoven, June 4th, 2004

Supervisors: dr. ir. I.J.B.F. Adan
prof. dr. ir. O.J. Boxma
ing. Y.M. Aarts

VOORWOORD

Dit verslag is het eindresultaat van mijn afstuderen voor de studie Toegepaste Wiskunde, afstudeerrichting stochastische besliskunde, aan de Technische Universiteit Eindhoven. Mijn afstudeerproject heb ik uitgevoerd voor de Technische Dienst van de Gemeente Eindhoven, Sector Realisatie. Gedurende negen maanden heb ik gewerkt aan problemen die gerelateerd waren aan verkeerslichten regeling zoals die binnen de gemeente Eindhoven gehanteerd wordt. Hierbij heb ik verschillende systemen wiskundig geanalyseerd, maar tevens ook software ontwikkeld, waarbij wiskundige strategieën de basis waren. Ik heb mijn afstudeerwerk verricht op de Technische Universiteit Eindhoven, waar ik gedurende deze negen maanden een kamer heb gedeeld met enkele AIO's.

Over het algemeen kijk ik met tevredenheid terug op mijn afstudeerproject. Ik heb met plezier gewerkt aan de wiskundige problemen die voortkwamen uit het onderzoek, al zijn er ook, zoals dat bij ieder onderzoek gaat, minder productieve perioden geweest. Wel was het erg prettig om aan verschillende deelonderwerpen te werken, zodat de afwisseling gedurende deze hele periode groot is geweest. Uiteindelijk ben ik tevreden met het eindresultaat van mijn onderzoek en met dit verslag.

Gedurende de afgelopen negen maanden ben ik door meerdere mensen begeleid. Allereerst wil ik graag Ivo Adan en Onno Boxma, beiden verbonden aan de Technische Universiteit Eindhoven, hartelijk bedanken voor de prettig begeleiding tijdens mijn afstuderen. Zowel het meedenken over de verschillende problemen als het doorlezen en becommentariëren van mijn verslag zijn van grote invloed geweest op het eindresultaat. Verder zou ik Yvonne Aarts van de Technische Dienst van de gemeente Eindhoven hartelijk willen danken voor haar hulp. Gedurende mijn afstuderen heeft ze me van allerhande praktische informatie voorzien omtrent de verkeerslichten regeling in de gemeente Eindhoven. Ook kon ik bij haar altijd terecht voor praktische vragen. Dank gaat ook uit naar Ron Nohlmans, tevens van de Technische Dienst van de gemeente Eindhoven, die het mogelijk heeft gemaakt mijn afstuderen binnen de gemeente uit te voeren.

Omdat dit afstudeerwerk tevens de afronding is van mijn studie van ongeveer zes jaar, wil ik allereerst mijn ouders en broer ontzettend bedanken voor hun blijvende enthousiasme, interesse en hulp tijdens mijn studieperiode. Verder wil ik de mensen binnen mijn vakgroep stochastische besliskunde bedanken voor mijn gezellige verblijf van de afgelopen negen maanden. Als laatste dank ik mijn vrienden voor de mooie studieperiode die ik hiermee definitief afsluit.

Monique van den Broek

Eindhoven, juni 2004

CONTENTS

CHAPTER 1

INTRODUCTION

The history of traffic signal control started on December 1868 when the first traffic signal was installed in London. Traffic signal setting is an important tool for the circulation of traffic through cities. With increasing traffic and congested roads the scheduling of traffic signals becomes more and more important. Indeed, delays at intersections depend most of the times on signal setting. The classical objectives of traffic control are according to Gazis[13] the improvement of safety and the decrease of discomfort of drivers. In other words, signal control is used as a tool for increasing the performance of traffic networks without changing the road structure.

The main goal of this thesis is to analyze the different traffic control systems nowadays used in the Netherlands, especially in Eindhoven. We try to improve the current control systems by using other mathematical models and methods and try to implement these new signal control systems into new software, so that it can be used by traffic engineers.

In this chapter we will give some information about the topics that are of interest for the problems described in this thesis. The outline of this chapter is as follows: In the first section some background information is given about the traffic situation and traffic control in the city of Eindhoven. The terminology used in this thesis is listed in section 1.2. In section 1.3 the various system-types of traffic are explained. The various control-types of traffic management that can be distinguished are illustrated in the next section. We list several performance criteria for optimal adjustments in section 1.5. In the next section, the assumptions used throughout this thesis are given. A brief overview of relevant literature is given in section 1.7. The subjects of the research in this thesis are briefly described in section 1.8. Finally, in the last section, the outline of this thesis is given.

1.1 The city of Eindhoven

In this section some background information about the traffic (signal) situation in the city of Eindhoven is given. First, some facts and numbers are presented to sketch the traffic situation in this city. Finally, the role of the department of administration for which this thesis was carried out is described.

1.1.1 Current situation

With more than two hundred thousand inhabitants Eindhoven is the fifth biggest city in the Netherlands. All these inhabitants together possess almost one hundred thousands motorized vehicles. The presence of a couple of big multinational companies leads to a lot of commuters from surrounding villages visiting Eindhoven daily as well, and causing an extra number of vehicles in the city.

The last fifty years the population has increased enormously and the average number of motorized vehicles per inhabitant as well, resulting in an extraordinary increase of the total amount of vehicles in the city of Eindhoven. On the other hand, the infrastructure of Eindhoven didn't change dramatically. This results in congested roads, especially during the morning and evening rush-hours. To prevent having too many traffic-jams, over one hundred fifty intersections are signalized to regulate the traffic. By adjusting the signals on these intersections properly, the traffic can be controlled to some extent. When at the end of the year 2004 the construction of the highway A50 from Nijmegen to Eindhoven is completed even more vehicles will come into Eindhoven. This is making it even more important to have a good traffic signal control system in Eindhoven.

1.1.2 Department of Technical Service

The local government of Eindhoven consists of eight different service organizations, and the *Urban Development and Management Service* is one of them. This service is responsible for the development and adaptation of the city. Traffic and transportation play an important role here. The service can be divided into a couple of different fields, including the field Strategy. They formulate points of view regarding the development of the city and they make the policy for the Urban Development and Management Service. The aims of Eindhoven to create good living and working conditions are written down in development-views, policy documents and long-range plans.

The field Strategy also tries to find for the necessary (financial) resources, and determines the different steps needed to get the desired results. An important aspect of the realization of the development and management plans is the cooperation with other organizations, like government institutions. They monitor the realization of the policy, so they can modify the policy if needed.

1.2 Terminology

In this section the terminology used in this thesis is listed.

Intersection	A set of approaches and a common crossing area.
Approach	A part of a road leading to the intersection such that the vehicles on it, have right of way simultaneously.
Effective green-time	The time during which the vehicles can leave the queue.
Effective red-time	The time during which no departures occur.
Cycle time	The time during which all the lights shown on each approach of the intersection have had right of way at least once.
Incompatible	Two approaches that cannot safely cross the intersection at the same time are called incompatible.
Clearance-time	The minimum amount of time between the end of the amber period of one, and the beginning of the green period of another traffic signal.
Conflict-group	A set of approaches that are all incompatible with each other.
Maximal conflict-group	A conflict-group, but with the extra requirement that each approach that is added to this set is not incompatible with all of the approaches in the maximal conflict-group.
Queue	A line of vehicles waiting to proceed through an intersection.
Delay	The difference between the travel time of a vehicle crossing the signalized junction and the time it would have taken if no other traffic were present and the stream in which it travels had constant right of way.
Signal control	A control which specifies the duration of the effective green- and red-times and the sequence of these times for each intersection.
Fixed-time control	A signal control method that allows only a fixed sequence and fixed duration of the effective green-time.

Fully-actuated control	A control method that allows a variable sequence and variable duration of the effective green-time, depending on vehicle demands.
Platoon	A group of vehicles travelling together because of signal control, or other factors.
Detector	A device by which vehicles are registered.

1.3 System-types

With reference to the system to be controlled, we can distinguish three types of systems. The single intersection, when an intersection operates independently of the others; the arterial system when some intersections along a one-way or a two-way path are coordinated; the network, when the road system to be controlled has one or more loops. In this section the three systems will be described in more detail.

1.3.1 Single intersection

An intersection is defined isolated if there are no interactions with the other surrounding intersections. In practice, this means that the mutual distance must be sufficient to eliminate the platoon effect generated by the traffic signals.

In the early seventies global agreements were made about the coding of approaches on an intersection. With this standard coding the numbers 1 to 99 are used which are broken down according to table 1.1.

Reserved for	Numbers
Motorized traffic	1 - 12
Bicycles	21 - 28
Pedestrians	31 - 38
Public transport	41 - 52

Table 1.1: Standard coding of approaches on an intersection

In this thesis we will mainly focus on motorized traffic, because they form the major target group. In figure 1.1 a basis intersection with all possible motorized traffic approaches is given.

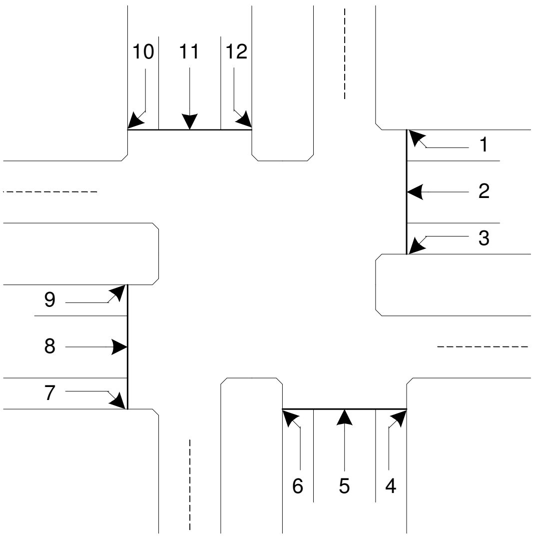


Figure 1.1: Standard coding of a single intersection

1.3.2 Arterial system

In an arterial system, adjacent intersections are connected by one-way or two-way paths. Traffic signals force vehicles to stop and to remain stopped for a certain time, and then release vehicles in platoons. The delays and speed changes caused by traffic signals considerably reduce the capacity of an arterial. Usually, the traffic signals of the intersections of the arterial system are phased. In figure 1.2 an example of two connected intersections is given.

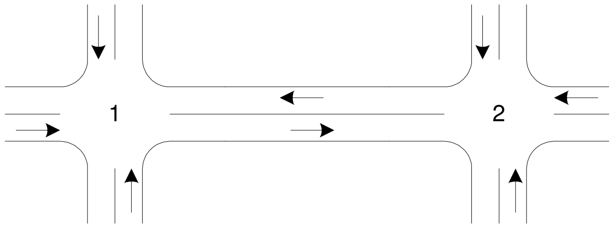


Figure 1.2: A simple example of an arterial system

Arterial systems occur often in big cities where the traffic is led around the city center to prevent getting too many cars in the center.

1.3.3 Network

In a network, a number of intersections are connected by one-way or two-way paths, such that the total system contains one or more loops. An urban signalized traffic network can be expressed by a directed graph. Traffic in a network is always moving in platoons, except for traffic arriving at the outside of the network. An arterial system may be considered to be a simple form of a network.

1.4 Control-types

Typically, traffic signals operate in one of three different control modes at a signalized intersection, namely, fixed-time control, pre-time control and fully-actuated control. In this section these three control-types will be explained. In the rest of this thesis we will only make use of the fixed-time and fully-actuated control, because the pre-time control can be modelled as a special case of the fixed-time control.

1.4.1 Fixed-time control

In fixed-time control, the traffic control calculated for average conditions, is independent of time and of the actual characteristics of the traffic flows. The effective green-times and red-times together with the cycle time are fixed. This is the most well-studied control. There are two related aspects of this kind of control. The first aspect is that of identifying the optimum signal sequence to be used at a given intersection, taking into account the characteristics of the intersection. The second aspect is the identification of optimal durations of the effective green-times of signals whose order is already determined by some method.

Numerical results have shown that this type of control is one of the best to apply at intersections with long queues and high utilization.

1.4.2 Pre-time control

In pre-time control the traffic control changes automatically in some times of the day, according to the amount of traffic present at that time of the day. Actually pre-time control is a form of fixed-time control, where for certain times of the day a different fixed-time control is determined. Up to ten years ago, most intersections in a city were functioning according to this type of control. Usually, five day-periods are distinguished, which are listed in table 1.2. For these periods a specific fixed-time control is determined, resulting in a pre-time control.

Period of the day	Point of time
Morning rush-hour	7.45 - 8.45
Daytime	8.45 - 16.45
Evening rush-hour	16.45 - 17.45
Evening	17.45 - 23.00
Night	23.00 - 7.45

Table 1.2: Division of a full day into five (pre-time) periods

Because pre-time control is a form of fixed-time control we leave this type of control in the remainder of this thesis out of consideration.

1.4.3 Fully-actuated control

Fully-actuated control takes into account the presence of vehicles to determine the adjustment of the traffic signals. The presence of vehicles is detected with a short loop (D1.1, D2.1 and D3.1) at the stop-line and a long loop (D1.2, D2.2 and D3.2) near the stop-line. In figure 1.3 a schematic view of the placement of these detection loops can be seen.

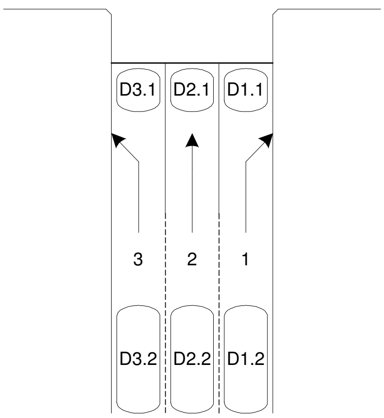


Figure 1.3: Placement of detection loops on an intersection

Nowadays, the most commonly used fully-actuated control works as follows: A signal only turns green, when vehicles are waiting in front of the stop-line. This can be detected by the short detection loop. When a signals turns green, it stays green for a minimum amount of time. After this time it stays green until the short loop (D1.1, D2.1, D3.1) has detected a gap time of duration at least γ. When no vehicles are detected by the short loop, it stays green until the long loop doesn't detect any vehicles. The effective green-time is bounded by a maximum time, when the signal (no matter what) turns red.

A special form of fully-actuated control is adaptive control. In normal fully-actuated control, the duration of the gap time and the maximum effective green-times are fixed and have to be specified by the traffic engineer. As traffic situations and conditions change, these timings should be changed as well. In Eindhoven, this is done not very frequently and certainly not systematically. With adaptive control, methods and algorithms are derived to automatically adjust these timings.

Results have shown that fully-actuated control is effective in case queues are short. It avoids that cars stop soon after the effective green-time is finished.

In the past, fully-actuated control strategies have been limited. The reason for this is that traffic engineers are sceptic about this type of control, because a characteristic of this sort of strategy is the reliance on heuristics rather than formal mathematical techniques to control the signals. But because of the increasing amount of vehicles on the road, most of the traffic engineers agree that in the future more and more intersections have to be regulated by fully-actuated control.

1.5 Performance criteria in traffic control

The main target of this thesis is to find good methods for adjusting traffic signals. Many performance criteria to evaluate traffic control systems can be proposed. Some of these criteria are listed below.

- (Weighted) average delay per vehicle.
- Maximum individual delay.
- (Weighted) average queue length.
- Maximum individual queue length.
- Throughput of the system.
- Percentage of the vehicles that are stopped as they go through the system.
- Average number of stops before a vehicle goes through the system.

The (weighted) average delay is commonly used as performance criterion, since this also expresses in some way the other criteria.

1.6 Assumptions

For clarity and simplicity, we make the following simplifying assumptions, valid for all the different problems handled in this thesis:

- All vehicles are identical.
- When driving, the vehicles travel with a constant speed.
- Queued vehicles accelerate instantaneously to their normal speed when activated and, vice versa, vehicles stop instantaneously when arriving at a queue.
- Vehicles waiting in the queue drive off with a fixed succession time that remains constant while driving.
- Vehicles (not in a network) arrive one by one according to a Poisson process.

1.7 Overview of literature

In this section we will discuss the literature already published about traffic control management. In each of the subsections literature about different traffic control systems will be handled separately.

1.7.1 Fixed-time control on a single intersection

The problem of optimizing signal control problems has been addressed by several researchers. One of the earliest results is the well known Webster's formula that relates the average delay at an intersection with the effective green-time. Based on this approximation, Webster [26] has developed a technique for finding a fixed-time control scheme.

Ramanathan et al. [20] presented a mathematical formulation for the problem of optimizing the operations of a modern traffic signal. The proposed formulation is a Mixed Integer Non-Linear program, which is solved with Branch and Bound techniques.

Riedel and Brunner [21] found a new approach for designing traffic control systems. First, the model of an intersection is derived by considering a simple intersection. Then, using a combination of Dynamic Programming and Branch and Bound, a control algorithm is developed.

1.7.2 Fully-actuated control on a single intersection

Fedotkin and Litvak [9] have studied a fully-actuated traffic control at an intersection. A new algorithm is introduced and analyzed. Connections between the new algorithm and already known algorithms are established.

In subsection 1.4.3 of this thesis the most commonly used fully-actuated control is described. A traffic signal stays green until the short loop has detected a headway of duration at least γ. Darroch et al. [7] investigated how one should choose the γ so as to minimize the average delay per vehicle at the intersection.

Taale [23] has presented two approaches to determine the settings in an adaptive traffic control and has compared the two of them. The first approach is based on Webster's formulas for the cycle time and the effective green-times. The second approach is based on evolutionary algorithms. By means of random mutations within solutions, random combinations of solutions and selection of the best solutions, a pre-defined performance function is optimized.

1.7.3 Signal control in a network

Improta [15] has given a good overview about basic arterial system control and network control. It deals with phasing intersections of an arterial. Furthermore a mathematical model for synchronizing the intersections within a network is given.

Finally, Shimizu et al. [22] have studied a signal control method which controls congestion lengths in a two-way traffic network systematically. A network control algorithm is presented, in which the three signal control parameters consisting of the cycle length, green split and offset are searched systematically so as to minimize the sum of congestion lengths in the traffic network.

1.8 Subjects of research

As we have seen in the previous section, a lot of research had been done in the field of traffic signal control. However, many improvements of existing techniques and development of new techniques are possible. In this section the subjects of interest in this thesis are described.

The main object of this thesis is to find good and efficient methods for adjusting traffic signals. This can be done for different types of systems and controls. Ramanathan et al. [20] formulated the problem of finding an optimal fixed control for a single intersection as a Mixed Integer Non-Linear program. We formulate the same model in a different way and solve the resulting Mixed Integer Program using new techniques. This results in a high quality fixed-time control.

The objective in this optimization program is given by the weighted average delay of vehicles on the intersection. But to calculate this average delay, and because no exact analytical expressions are known we have to make use of approximation formulas. There is a variety of approximation formulas that can be used, but in this thesis we will derive another accurate approximation formula of the average delay depending on different parameters.

A different view on the traffic signal issues can be obtained by modelling the traffic system as a polling system. The basic polling system is a system of multiple queues, attended to by a single server in a cyclic order. Many resemblances between a polling and a traffic system can be found. So methods used to optimize polling systems can be translated and used to optimize traffic systems. As we have mentioned in the previous section, Darroch et al. [7] have investigated the most commonly used fully-actuated control outlined in subsection 1.4.3. In this control, the effective green-time is bounded by a maximum time, when the signal (no matter what) turns red. Nowadays this maximum time is based on a fixed-time control, developed for the intersection. In this thesis we will investigate a new method of finding these maximum times based on k-limited polling.

When we take these maximum effective green-times to be infinite, the fully-actuated control corresponds to the exhaustive control. With exhaustive control, the effective green-time for each approach lasts until the queue has become empty. Exhaustive control is effective at intersections, where queues are not so long. It is a precursor of the current fully-actuated control as described in this chapter. In this thesis we will analyze this type of control mathematically and compare these results with results obtained via simulation.

There is an extensive literature on traffic signal control, but in the area of traffic networks the results are disappointing. A specific type of a network will be investigated: the arterial system. This type of system is less complex than the generical network by the absence of difficult routing issues. An additional reason for investigating an arterial system, is the fact that in Eindhoven a large project has been started at the beginning of 2004 to reschedule the main arterial in the city of Eindhoven. In this thesis we will compare two different methods of phasing the intersections of an arterial.

1.9 Outline of the thesis

In the previous section, we have described the subjects of research. In this section the main topics of this thesis are outlined. The remainder of this thesis is structured as follows.

In chapter 2 a method for finding a fixed-time control for an isolated single intersection is given. In the next chapter, we apply this method to an intersection in the city of Eindhoven. The results are compared with the current situation. In chapter 4 we shall determine an approximation formula for the average delay of vehicles on an isolated intersection, with a fixed-time control. Furthermore we compare the results of this new formula with existing formulas, that are nowadays world-wide used in traffic signal management. The exhaustive control system is analyzed by means of moment fitting techniques in chapter 5. For a simple isolated intersection we shall approximate the average effective green-time and the average delay and compare these results with results obtained via simulation. Subsequently, in chapter 6 we shall give a method for finding a good dynamic traffic control system. This method is applied to an intersection in the city of Eindhoven and the results are compared with results of the current situation. In the area of multiple intersections, we investigate in chapter 7 different ways how to phase the signals of different intersections. Finally, in the last chapter a practical case of phasing traffic signals in the city of Eindhoven is investigated.

CHAPTER 2

OPTIMIZING FIXED-TIME CONTROL

This chapter presents a solution method to find a fixed-time control for an isolated single intersection. The problem is formulated as a Mixed Integer Programming (MIP) problem. The objective to be minimized is the weighted average delay of vehicles on the intersection. This average delay is approximated by the formula of Webster, which is nonlinear. These nonlinear functions can be approximated by linear functions, the so-called piecewise linear formulation. The objective curve is divided into pieces that are approximated by straight lines. These straight lines are added in an iterative process described in this chapter.

The outline of this chapter is as follows: In section 2.1 the problem of determining a fixed-time control for an isolated single intersection is described. In section 2.2 a simplified problem is translated into a Mixed Integer Program. To find better and faster solutions an algorithm is introduced in section 2.3. In the next chapter we will apply the method described in this chapter to an intersection in the city of Eindhoven.

2.1 Problem description

This section introduces the problem of finding an optimal fixed-time control for an isolated single intersection. Consider a single intersection in a city. This intersection has no interactions with surrounding intersections. To regulate the traffic on this intersection, traffic signals are placed. These signals have a fixed-time cyclic control, that means that the effective green-times and the order in which signals turn green are fixed.

Some approaches on the intersection are incompatible. The signals cannot turn green at the same time because of safety reasons. For each pair of incompatible signals clearance-times are given. When signal i and signal j are incompatible, two clearance times s_{ij} and s_{ji} are introduced. For example s_{ij} indicates the minimum time between the end of the effective green-time of signal i and the beginning of the effective green-time of signal j.

When a signal switches from red to green, vehicles start to accelerate. Drivers don't pay attention to traffic signals during their acceleration. That's why the length of the effective green-time is bounded by a minimum value. When the signal of an approach turns green, all the incompatible approaches have to wait until the green-period is finished. To limit this waiting time, or delay, the length of the effective green-time is bounded by a maximum as well.

Only traffic control systems that yield a stable system are determined. The system is called stable, if for each approach the average amount of vehicles that can move on during the effective green-time is larger than the total average number of vehicles that arrives during a whole cycle.

Due to the fact that some approaches on an intersection are incompatible, not all signals can be turned green at the same time. Vehicles on an approach wait until signals of incompatible approaches turn red and by that reason are delayed. The delay as a function of the effective green-time can be approximated by the formula of Webster [26]. The main objective is to find a fixed-time control in such a way that the weighted average delay of vehicles on the intersection is minimized.

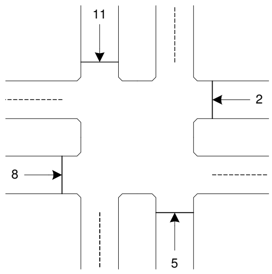


Figure 2.1: Example of a simple intersection

Example:
Consider the intersection as given in figure 2.1. Here the signals of the approaches 2 and 8 are green at the same time. The same holds for the signals of the approaches 5 and 11. Signals 2 and 8 are incompatible with signals 5 and 11. Note, that the approaches 2 and 8 are 'sub-incompatible'. This means that for example, vehicles turning left on approaches 2 and 8 have right of way together, but can come into conflict with each other. These vehicles have to obey the common traffic rules.

2.2 The initial model formulation

In this section the description of the problem in section 2.1 is translated into an optimization model. First, an informal verbal presentation of the model is provided, followed by the extensive notation needed to describe all aspects of the model. The objective function and each constraint is developed separately. A model summary is listed at the end.

2.2.1 Verbal model statement

The objective function and constraints are expressed in the following qualitative model formulation:

Minimize: *Weighted average delay of vehicles on the intersection.*
Subject to:

1. *For all traffic signals: Length of the effective green-time must be smaller than or equal to the maximum green-time and must be greater than or equal to the minimum green-time,*
2. *For all traffic signals: The beginning and the end of the effective green-times must fall in the range* $[0, Cycle)$,
3. *For all traffic signals: The delay of vehicles as a function of the length of the effective green-time is given by the formula of Webster,*
4. *For all traffic signals: The green-period cannot overlap the green-periods of incompatible signals and incompatible approaches have to take into account a clearance-time for safety,*
5. *For all traffic signals: The length of the effective green-time must be large enough to handle all the arriving vehicles at that signal,*
6. *For one traffic signal: The beginning of the effective green-time is equal to zero.*

2.2.2 Notation

The verbal model statement of the problem can be specified as a mathematic model using the following notation.

Indices:

I	set of traffic signals, index i (second index j)
Q_i	set of incompatible traffic signals of signal i

Parameters:

n	number of traffic signals on the intersection
c	cycle time (integer)
λ_i	arrival rate of vehicles at traffic signal i
μ_i	departure rate of vehicles at traffic signal i
ρ_i	occupation rate of traffic signal i ($\rho_i = \lambda_i/\mu_i$)
w_i	load/weight of traffic signal i
mg_i	minimum effective green-time of traffic signal i
Mg_i	maximum effective green-time of traffic signal i
s_{ij}	necessary time between the end of the green period of signal i and the beginning of signal j

Variables:

b_i	time when signal i turns green (range $[0, c)$)
e_i	time when signal i turns red (range $[0, c)$)
g_i	length of effective green-time of signal i ($g_i = (e_i - b_i) \bmod c$)
d_i	average delay of vehicles at traffic signal i
D	weighted average delay of vehicles at the intersection ($D = \sum_{i=1}^{n} w_i d_i$)
z_{ij}	1 if $e_i - b_j < 0$
	0 otherwise

2.2.3 Mathematical model

The constraints as formulated in the first section of this chapter can be translated into mathematical constraints, using the notation above. The constraints formulated in this subsection have the same numbers as the constraints in the verbal model statement earlier.

Constraint 1

The length of the effective green-time is bounded by a maximum and minimum. The resulting constraint assumes a simple format.

$$mg_i \leq g_i \leq Mg_i, \qquad i = 1, \ldots, n$$

Constraint 2

The variable z_{ij} is used to circumvent difficulties that appear by the cyclic behavior of the control. The lengths of specific intervals $e_i - b_j$ have to be calculated. When e_i occurs after b_j in the cycle there is no problem. On the other hand, if b_j occurs after e_i, the duration of the interval will be negative. In the latter case a cycle length has to be added to the solution $e_i - b_j$. This can be realized by always computing $e_i - b_j + cz_{ij}$ when a length of time has to be computed. The following holds for variable z_{ij}:

$$z_{ij} = \begin{cases} 1 & \text{if } e_i - b_j < 0 \\ 0 & \text{otherwise} \end{cases}, \qquad i, j = 1, \ldots, n$$

This is realized by adding the following constraints.

$$\begin{aligned} b_j - e_i &\leq cz_{ij}, && i, j = 1, \ldots, n \\ e_i - b_j &< c(1 - z_{ij}), && i, j = 1, \ldots, n \end{aligned}$$

So the length of the effective green-time is then given by:

$$g_i \quad = \quad e_i - b_i + cz_{ii}, \qquad i = 1, \dots, n$$

Furthermore, we have to claim that the beginning as well as the end of the effective green-period has to lie in the interval $[0, c)$. This is guaranteed by the following constraint:

$$\begin{array}{lllll} 0 & \leq & b_i & < & c, \qquad i = 1, \dots, n \\ 0 & \leq & e_i & < & c, \qquad i = 1, \dots, n \end{array}$$

Constraint 3

The average delay d_i of traffic signal i depends on the length of the effective green-time g_i of signal i. This function is not linear, because the d_i decreases slower while g_i increases. The first derivative is negative, while the second derivative is positive. A good approximation of the average delay of vehicles can be obtained by the formula of Webster. The formula given below is a part of the original formula, without correction term. In chapter 4 this formula is discussed and alternative and better formulas are derived.

$$d_i^{WEB}(g_i) \quad := \quad \frac{(c-g_i)^2}{2c(1-\rho_i)} + \frac{\rho_i c^2}{2g_i(\mu_i g_i - \lambda_i c)}, \qquad i = 1, \dots, n$$

This function is not linear and needs to be approximated by a number of piecewise linear equations (N_i). An example of the Webster function is given in figure 2.2. In this example, the function is bounded from below by four line pieces. Because of the fact that the objective function has to be minimized, the following constraints have to be added to the program.

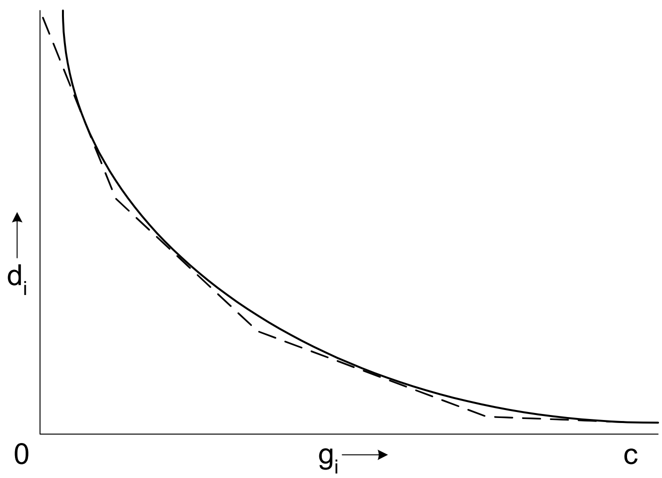


Figure 2.2: Example of delay d_i versus the effective green-time g_i according to Webster's formula

$$d_i \quad \geq \quad \alpha_{ik} g_i + \beta_{ik}, \qquad k = 1, ..., N_i, i = 1, \dots, n$$

Constraint 4

For each pair of incompatible traffic signals the green-periods cannot overlap. This corresponds to the fact that the b_j or the e_j cannot fall in the effective green-time of signal i. The translation of this requirement into mathematics is symbolized in figure 2.3. This is only true, when the dashed line plus the dotted line minus the solid lines are equal to a whole cycle length. This results in the following constraint:

$$(e_i - b_j + cz_{ij}) + (e_j - b_i + cz_{ji}) - (e_i - b_i + cz_{ii}) - (e_j - b_j + cz_{jj}) = c, \qquad i, j = 1, \ldots, n$$

which can be simplified to:

$$z_{ij} + z_{ji} - z_{ii} - z_{jj} = 1, \qquad i, j = 1, \ldots, n$$

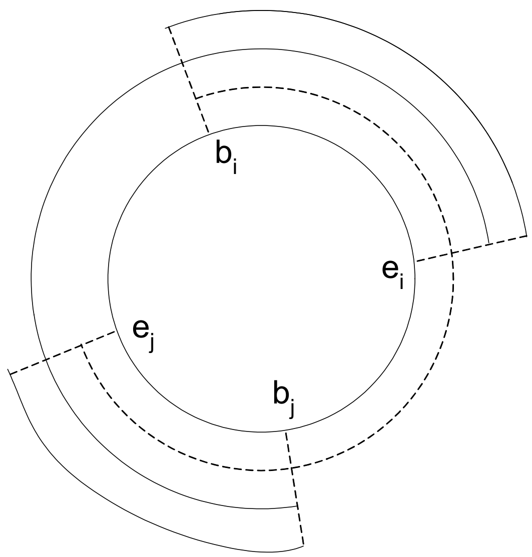


Figure 2.3: Explanation of constraints belonging to incompatible signals

Between the end of the effective green-time of traffic signal i and the beginning of the effective green-time of traffic signal j must be at least a clearance-time s_{ij}. This can be written as follows:

$$b_j - e_i + cz_{ij} \geq s_{ij}, \qquad i, j = 1, \ldots, n$$

Constraint 5

We require that the traffic control leads to a stable system. The system is called stable, if for each approach the average amount of vehicles that can leave the approach during the effective green-time is (strictly) larger than the total average amount of vehicles that arrive during a whole cycle. In other words, the number of vehicles does not explode. This can be easily translated into the following constraint:

$$\mu_i g_i > \lambda_i c, \qquad i = 1, \ldots, n$$

Constraint 6

The fixed-time control produces a cyclic scheme of times when signals turn green and red. When to all these times the same number $\delta \in [0, c)$ is added, the resulting scheme is identical to the first, with the small difference that all points of time are moved over a period δ. To reduce the computation time, we can simply fix the beginning of the effective green-time of the first signal to 0. The resulting constraint assumes a simple format:

$$b_{signal1} = 0$$

Objective

The objective function being the weighted average delay of all vehicles on the intersection, has to be minimized

Minimize:

$$\sum_{i=1}^{n} w_i d_i$$

The mathematical description of the model can now be summarized as follows. The constraints in the model hold for all i and j

Minimize:

$$\sum_{i=1}^{n} w_i d_i$$

Subject to:

$$\begin{array}{rcll}
b_j - e_i & \leq & cz_{ij} & \\
e_i - b_j & < & c(1 - z_{ij}) & \\
g_i & \leq & mg_i & \\
g_i & \geq & Mg_i & \\
d_i & \geq & \alpha_{ik} g_i + \beta_{ik}, & k = 1, ..., N_i \\
z_{ij} + z_{ji} - z_{ii} - z_{jj} & = & 1 & \\
b_j - e_i + cz_{ij} & \geq & s_{ij} & \\
\mu_i g_i & > & \lambda_i c & \\
b_i & < & c & \\
e_i & < & c & \\
g_i & < & c & \\
b_i & \geq & 0 & \\
e_i & \geq & 0 & \\
g_i & \geq & 0 & \\
d_i & \geq & 0 & \\
z_{ij} & \in & \{0, 1\} & \\
b_{signal1} & = & 0 &
\end{array}$$

The program described above (MainProgram) is called a Mixed Integer Program, because some of the variables used to formulate the problem are continuous and some of them are integer (binary). The MainProgram can be solved with AIMMS, a linear and mixed integer program solver.

2.3 Adding piecewise linear constraints

In this section, we present an algorithm to find more accurate and faster optimal solutions of the mathematical problem described in the previous section. For each signal i the delay as a function of the length of the effective green-time is approximated by N_i linear equations (see figure 2.2). The number of linear equations is given as a parameter. The problem is to find the number of linear approximation equations so that the accuracy ϵ of the solution is high enough. This number is hard to find, and the accuracy ϵ can only be achieved by trial and error.

2.3.1 Concept

As we have seen in the previous section, an optimal solution of the Mixed Integer Program can be determined given the linear constraints that approximate the Webster function. Here we assume that $N_i = 2$ for all i. To be more specific: The Webster function is approximated by two linear equations given by the tangent line of the Webster function at two chosen values. These values are chosen in the following way:

$$\begin{array}{rcll}
g_i^{Min} & := & \frac{\lambda_i c}{\mu_i} + 1, & i = 1, \ldots, n \\
g_i^{Max} & := & c, & i = 1, \ldots, n
\end{array}$$

So, the linear approximation constraints (given in the previous section) are of the form:

$$d_i \quad \geq \quad \alpha_{ik} b_i + \beta_{ik}, \qquad k = 1, 2$$

The α_{ik} and β_{ik} are given by:

$$\begin{aligned}
\alpha_{i1} &= d_i^{WEBACC}(g_i^{Min}), & i &= 1, \ldots, n \\
\beta_{i1} &= d_i^{WEB}(g_i^{Min}) - g_i^{Min}\alpha_{i1}, & i &= 1, \ldots, n \\
\alpha_{i2} &= d_i^{WEBACC}(g_i^{Max}), & i &= 1, \ldots, n \\
\beta_{i2} &= d_i^{WEB}(g_i^{Max}) - g_i^{Max}\alpha_{i2}, & i &= 1, \ldots, n
\end{aligned}$$

The derivative of the Webster function ($d_i^{WEBACC}(g_i)$) is given in Appendix A.
A pseudo-optimal solution can be easily determined. So the pseudo-optimal g_i denoted by $\bar{g}_i^*$ are known. Because the Webster function is bounded from below by the linear constraints, a lower bound (LB) for the real optimal solution is given by the objective function value of the Mixed Integer Program. An upper bound (UB) is given by the weighted sum of the Webster functions, with the $\bar{g}_i^*$ filled in. Now a new constraint can be added to the set of old constraints (so $N_i := 3$):

$$d_i \quad \geq \quad \alpha_{i3} g_i + \beta_{i3}, \qquad i = 1, \ldots, n,$$

where α_{ik} and β_{ik} are given by:

$$\begin{aligned}
\alpha_{i3} &= d_i^{WEBACC}(\bar{g}_i^*), & i &= 1, \ldots, n \\
\beta_{i3} &= d_i^{WEB}(\bar{g}_i^*) - \bar{g}_i^*\alpha_{i3}, & i &= 1, \ldots, n
\end{aligned}$$

In figure 2.4 the process of adding constraints is shown.

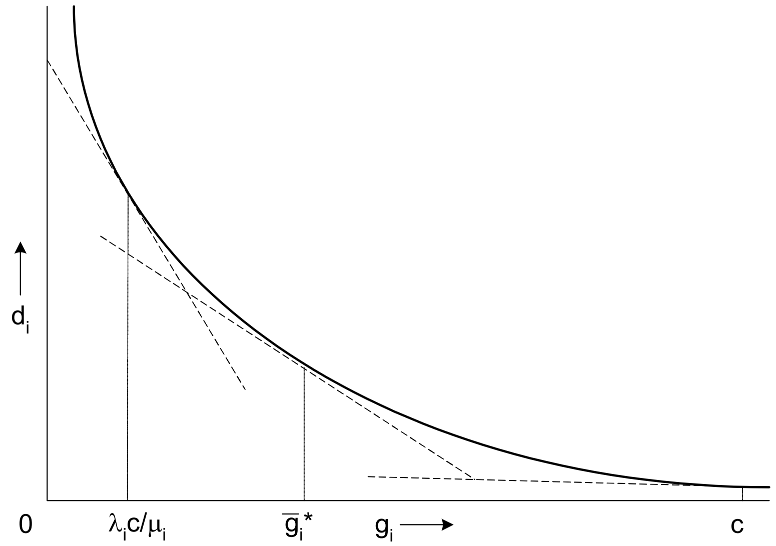


Figure 2.4: Piecewise linear approximation of the Webster function

By adding these new constraints the approximation of the Webster function by these linear equations will be better in the neighborhood of the optimal solution. The resulting Mixed Integer Program can again be solved. This procedure can be repeated until $UB - LB$ is small enough. So, we proceed with the algorithm until $UB - LB$ is smaller than a value ϵ times UB. We have chosen ϵ equal to 0.001. The iterative algorithm is summarized in the next subsection.

2.3.2 Algorithm

The algorithm described in the previous subsection is outlined below and determines the optimal sequence and durations of effective green-times.

```
N_i := 0;
Generate approximation constraints;
Solve MainProgram;
LB =MainProgram.Objective;
UB = sum_{i=1}^{n} w_i d_i^{WEB}(ḡ_i^*);
WHILE (UB − LB) > εUB DO
    FOR each traffic signal DO
        N_i := N_i + 1;
        α_{iN_i} = d_i^{WEBACC}(ḡ_i^*);
        β_{iN_i} = d_i^{WEB}(ḡ_i^*) − ḡ_i^* α_{iN_i};
        Add Constraint d_i ≥ α_{iN_i} g_i + β_{iN_i};
    ENDFOR;
    Solve MainProgram;
    LB =MainProgram.Objective;
    UB = sum_{i=1}^{n} w_i d_i^{WEB}(ḡ_i^*);
ENDWHILE.
```

The proposed algorithm relies on formal mathematical techniques rather than heuristics. Under the assumptions, it is expected that the optimal solution found by the algorithm is globally optimal. Since the algorithm is mathematically based, it may be shown that the solution produced is optimal for the conditions and within the constraints imposed. The solution can thus be used with confidence that there is no better alternative under the given circumstances.

CHAPTER 3

OPTIMIZING FIXED-TIME CONTROL: EXAMPLE

The intersection that is investigated in this chapter, is one of the busiest intersections in Eindhoven. During day time almost 30000 motorized vehicles pass the intersection. The intersection is part of the main arterial around the center of Eindhoven. Especially during the rush hours, in the morning from 7.45 AM until 8.45 AM and in the afternoon from 4.45 PM until 5.45 PM, the approaches are filled with vehicles. In this chapter we will investigate a fixed-time signal control during the rush hour in the late afternoon.

The outline of this chapter is as follows: First, in section 3.1 the input, as well as some background information of the intersection is given. The method described in chapter 2 is implemented with the optimization software package AIMMS. The current signal scheme and the new found signal scheme are presented in section 3.2 and 3.3 respectively. In section 3.4 the structure of the simulation program is discussed. In section 3.5 the newly found signal scheme is compared with the current scheme by simulation. Finally, a sensitivity analysis of the results is done in the last section.

3.1 Introduction

In this section all the necessary input and information of the investigated intersection is given.

3.1.1 Notation

The following notation is used in the rest of this chapter.

$$
\begin{aligned}
I &= \text{Set of traffic signals (index } i\text{)} \\
c &= \text{Cycle time} \\
\lambda_i &= \text{Arrival rate of vehicles at traffic signal } i \\
\mu_i &= \text{Departure rate of vehicles at traffic signal } i \\
b_i &= \text{Time when signal } i \text{ turns green (range } [0, c)) \\
e_i &= \text{Time when signal } i \text{ turns red (range } [0, c)) \\
g_i &= \text{Length of the effective green-time of signal } i \\
w_i &= \text{Weight of signal } i \text{ (used in the objective)} \\
d_i &= \text{The average delay of vehicles at signal } i \\
\rho_i^* &= \text{Degree of saturation of signal } i
\end{aligned}
$$

3.1.2 Input and background information

The investigated intersection is schematically presented in figure 3.1. As can be seen in this figure, there

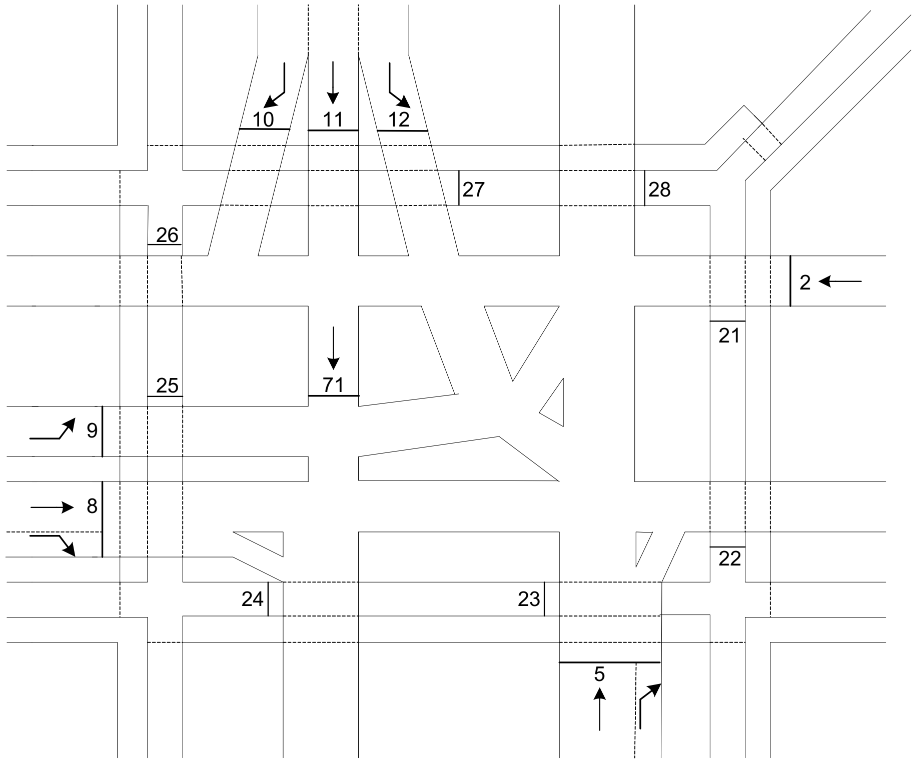


Figure 3.1: Overview of intersection (Scale 1:700)

are seven car signals (numbered 2, 5, 8, 9, 10, 11, 12). Furthermore, there are sixteen other signals for cyclists (numbered 21, 22, 23, 24, 25, 26, 27, 28) and pedestrians (numbered 31, 32, 33, 34, 35, 36, 37, 38). Finally, there is one special signal 71 for busses. The length of the effective green-time of the cyclist and pedestrian signals is fixed, and is based on the design of the intersection. When the average speed of the road user and the length of the traversed way is known, the fixed effective green-time can be easily computed. In table 3.1 the average speed of the different road users is given.

Road user	Average speed (m/s)
Car (to the left)	8.0
Car (to the right)	8.0
Car (straight on)	12.0
Bus	8.0
Cyclist	4.0
Pedestrian	1.2

Table 3.1: Average speed of different types of road users

In 1999 the number of arriving vehicles during one hour was counted at all signals. The arrival rate λ_i of signal i, is the traffic intensity given in number of arriving cars per second and can be directly calculated from this counting. The drive off capacity, given in the maximum number of cars leaving per hour, of vehicles for the different signals can be easily computed, based on the design of the approaches, resulting in the total amount of vehicles leaving the approach per second (departure rate μ_i). The objective that

i	w_i	λ_i	μ_i
2	0.121	0.0731	0.4722
5	0.158	0.0956	0.4722
8	0.153	0.0922	0.4722
9	0.175	0.1058	0.4722
10	0.068	0.0411	0.4722
11	0.203	0.1228	0.4722
12	0.119	0.0717	0.4722

Table 3.2: Input of intersection

is minimized is the weighted average delay of vehicles on the intersection. In table 3.2 the input of this intersection is given.

As the computation time of the AIMMS program increases exponentially with the number of traffic signals, we try to minimize this number. The intersection can be simplified by taking one cyclist/pedestrian signal instead of four signals. For example, signals 21, 22, 31 and 32 can be represented by signal 21, having the same set of incompatible signals. Later on, we can fit signals 22, 31 and 32 in the signal scheme manually.

From figure 3.1, we can derive which approaches (and signals) are incompatible. At this intersection, partial conflicts are permitted. For example, approach 8 and 24 can have right of way simultaneously. Possible conflicts between road users are solved by following the normal traffic rules. In Appendix C can be found, which signals are incompatible with each other. In the same Appendix the clearance-times of these incompatible signals are given.

3.2 Current situation

The current signal scheme is presented in table 3.3. The cycle time is 90 seconds.

i	b_i(s)	e_i(s)	g_i(s)	i	b_i(s)	e_i(s)	g_i(s)
2	16.5	32.5	16.0	25	62.0	80.5	18.5
5	67.0	85.0	18.0	26	57.0	73.5	16.5
8	23.5	55.0	31.5	27	17.0	33.0	16.0
9	36.0	57.0	21.0	28	12.0	26.5	14.5
10	78.0	12.0	24.0	31	-	-	-
11	65.0	11.5	36.5	32	-	-	-
12	0.0	12.0	12.0	33	24.0	48.0	24.0
71	64.0	17.0	43.0	34	26.0	47.5	21.5
21	70.0	85.0	15.0	35	-	-	-
22	69.0	75.5	6.5	36	-	-	-
23	23.5	61.0	37.5	37	16.5	31.0	14.5
24	23.5	51.5	28.0	38	12.0	22.0	10.0

Table 3.3: Current signal scheme

The intersection is part of the main arterial of the city. For a good circulation, the traffic signals of the intersections of this arterial are phased. As a result all intersections of this arterial have the same cycle time of 90 seconds. In table 3.3 no settings for traffic signals 31, 32, 35 and 36 are given. These signals for a pedestrian crossing turn green, when someone pushes a button.

3.3 Results

In figure 3.2 the output screen of the AIMMS program is displayed. On this screen, the input is given at the left-hand side. At the other side, the output is presented. A bar chart of the optimal signal scheme with a cycle time of 57 seconds is given. Furthermore, a graph as well as a table of the objective as function of the cycle time can be seen at the bottom of the screen. Increasing the optimal cycle time of 57 seconds, will gradually increase the objective. On the other hand, decreasing the cycle time will lead to much higher objective values.

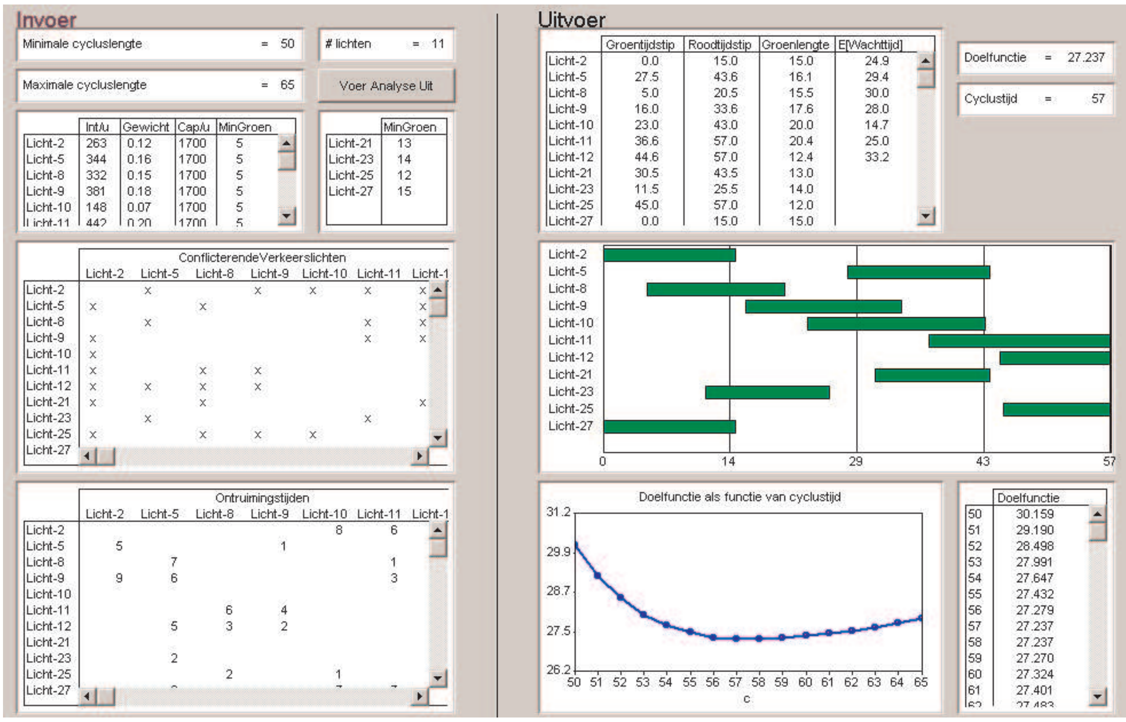


Figure 3.2: Print of AIMMS output screen

In table 3.4 the optimal signal scheme with a cycle time of 57 seconds is given.

i	b_i(s)	e_i(s)	g_i(s)	i	b_i(s)	e_i(s)	g_i(s)
2	0.0	15.0	15.0	25	45.0	57.0	12.0
5	27.5	43.6	16.1	26	45.0	57.0	12.0
8	5.0	20.5	15.5	27	0.0	15.0	15.0
9	16.0	33.6	17.6	28	53.6	15.0	18.4
10	23.0	43.0	20.0	31	30.5	43.6	13.1
11	36.6	57.0	20.4	32	30.5	43.6	13.1
12	44.6	57.0	12.4	33	11.5	20.5	9.0
71	36.1	4.0	24.9	34	11.5	32.1	20.6
21	30.5	43.5	13.0	35	45.0	57.0	12.0
22	30.5	43.5	13.0	36	47.0	57.0	10.0
23	11.5	25.5	14.0	37	0.0	15.0	15.0
24	11.5	36.1	24.6	38	53.6	15.0	18.4

Table 3.4: Optimal signal scheme

3.4 Simulation program

To test whether the developed solution method works well, a simulation program has been written. It simulates a fixed time control of an isolated single intersection. In this section the structure of this simulation program is explained.

3.4.1 Input

The input needed to simulate the system is summarized below:

$$
\begin{aligned}
I &= \text{Set of traffic signals (index } i\text{)} \\
c &= \text{Cycle time} \\
\lambda_i &= \text{Arrival rate of vehicles at traffic signal } i \\
\mu_i &= \text{Departure rate of vehicles at traffic signal } i \\
b_i &= \text{Time (mod } c\text{) when signal } i \text{ turns green (range } [0, c)\text{)} \\
e_i &= \text{Time (mod } c\text{) when signal } i \text{ turns red (range } [0, c)\text{)}
\end{aligned}
$$

The vehicles at signal i arrive according to a Poisson process with parameter λ_i. The departure process consists of deterministic departures. Each $1/\mu_i$ seconds a vehicle leaves approach i. The b_i and e_i are chosen in such a way that incompatible approaches don't have right of way simultaneously.

3.4.2 Discrete-event simulation

We have written a discrete-event simulation with an event scheduling approach. The event scheduling approach concentrates on the events and how they affect the system. The three events that can be distinguished are:

1. Arrival of a vehicle
2. Beginning of the effective green-time
3. End of the effective green-time

The departure of a vehicle can be seen as event four. But to obtain a value for the delay of a vehicle, the order in which vehicles have arrived needs to be stored. We didn't make use of this departure-event, but kept track of the amount of 'work', c.q. the time to clear the vehicles, waiting in the queue. Out of the amount of departure time waiting at the signal and the beginning and end of the effective green-times, the departure time and subsequently the delay can immediately be calculated when a vehicle arrives.

When for example, two approaches belong to the same traffic signal, we doubled the departure rate of this signal. This is not valid when a vehicle arrives and there is no waiting queue. In that case the departure rate is equal to the original departure rate instead of the doubled one.

We keep track of the time points at which the next events of the different types occur. To record these time points we make use of a binary search tree. In a tree, the time points are not ordered in a straight line, like earliest event first, and so on. Instead, the starting time point, called the root, is linked to two other nodes, called its children, and those nodes in turn are linked to other children, and so on. Formally, a tree is either empty, or a root, which is connected to one or more other trees, called the subtrees of the root. The order of all time points in this tree is important. Formally, in a binary search tree the following holds:

- All the time points in the left subtree take place earlier than the time point of the root
- All the time points in the right subtree take place later than the time point of the root
- The left and right subtrees are also binary search trees

We can conclude, that the event that takes place first, is the leftmost node in the tree. We use a binary search tree in order to minimize the distance we have to go to reach any given element. Searching for an element in a binary tree containing n nodes is an $O(\log n)$ process and building the tree in the first place is an $O(n\log n)$ process, if the tree is reasonably well balanced.

The simulation then consists of finding the smallest time point in this tree, setting the current time to this event time and executing the corresponding activities. Here we will describe how these events affect the system and what activities are carried out.

1. Arrival of a vehicle

With an arrival, the total amount of vehicles waiting at that approach is increased and a new arrival at the same approach is simulated.

2. Beginning of the effective green-time

When a traffic signal turns green, the state of this signal changes. Vehicles at the approach have right of way until the end of the effective green-time. A new event, the beginning of the effective green-time in the next cycle, is generated and added to the binary search tree.

3. End of the effective green-time

When a traffic signal turns red, the state of this signal changes. A new event, the end of the effective green time, so the beginning of the effective red-time, is generated and added to the binary search tree. When the signal turns red and a vehicle is not completely driven off, two different scenarios can be followed. We have chosen for preemptive resume, that means, the drive-off time of the vehicle is preempted, but is continued at the beginning of the next effective green-time.

After each event the total number of waiting vehicles at the approaches is updated. As stopping criterium for this simulation the runlength is taken. When the current simulation time exceeds this runlength the simulation is ended.

3.4.3 Overview

The simulation described in the previous subsection is outlined below.

```
Simulation time:= 0;
Initialize system and statistics;
Initialize event list;
WHILE simulation time<runlength DO
        Determine next event type in binary search tree;
        Remove first event;
        Update simulation time;
        Update system state and statistics;
        Generate and add future events;
ENDWHILE;
Compute and print statistics.
```

3.5 Analysis versus simulation

In this section three signal schemes for the intersection are simulated with the program, described in the previous section. The objective to be minimized is equal to the weighted sum of the average delays. First, the current situation, with a cycle time of 90 seconds is investigated. Because the investigated intersection is part of an arterial in which the intersections are phased, it is probably desired that the current cycle length is preserved. Therefore, in the second case the newly found signal scheme with a chosen cycle time of 90 seconds is investigated as well. Finally, the optimal signal scheme with a cycle time of 57 seconds is tested. For each case the average delay is calculated. From this, the objective given by the weighted average delay immediately follows. In the second and third table, the approximation of the average delay based on Webster's formula without correction term, is given as well.

Besides this, the degree of saturation (ρ_i^*) is calculated. This degree indicates the fraction of the maximum capacity that is utilized and can be computed as follows:

$$\rho_i^* = \frac{\lambda_i c}{\mu_i g_i}$$

For reliability of the results and small confidence intervals, the system is simulated for 24 hours = 86400 seconds and with 100 repetitions. Then a 95%-confidence interval for the average delay is constructed.

3.5.1 Current situation - Cycle time 90 seconds

In table 3.5 the simulation results as well as the input and saturation degree of the current situation are presented.

i	λ_i	μ_i	g_i(s)	ρ_i^*	d_i(s)
2	0.0731	0.4722	16.0	0.870	64.58(± 1.183)
5	0.0956	0.4722	18.0	1.012	862.73(± 70.470)
8	0.0922	0.4722	31.5	0.558	26.62(± 0.059)
9	0.1058	0.4722	21.0	0.961	131.73(± 8.100)
10	0.0411	0.4722	24.0	0.326	28.86(± 0.074)
11	0.1228	0.4722	36.5	0.641	24.88(± 0.058)
12	0.0717	0.4722	12.0	1.138	6048.36(± 133.163)
Objective: 898.77 seconds					

Table 3.5: Output of current situation

3.5.2 New signal scheme - Cycle time 90 seconds

In table 3.6 the simulation results of the new signal scheme with a cycle length of 90 seconds are presented.

i	λ_i	μ_i	g_i(s)	ρ_i^*	d_i(s)	d_i^{app}(s)
2	0.0731	0.4722	21.0	0.663	36.19(± 0.120)	41.17
5	0.0956	0.4722	28.2	0.646	30.16(± 0.072)	33.59
8	0.0922	0.4722	27.3	0.644	30.90(± 0.073)	34.25
9	0.1058	0.4722	29.9	0.675	29.72(± 0.076)	33.37
10	0.0411	0.4722	47.0	0.167	13.60(± 0.055)	12.06
11	0.1228	0.4722	35.1	0.667	26.27(± 0.061)	28.81
12	0.0717	0.4722	21.4	0.638	34.73(± 0.092)	39.41
Objective: 29.55 seconds						

Table 3.6: Output of new situation with $c = 90$ seconds

3.5.3 Optimal signal scheme - Cycle time 57 seconds

In table 3.7 the simulation results of the optimal signal scheme with a cycle length of 57 seconds are presented.

i	λ_i	μ_i	g_i(s)	ρ_i^*	d_i(s)	d_i^{app}(s)
2	0.0731	0.4722	15.0	0.588	21.83(± 0.061)	24.92
5	0.0956	0.4722	16.1	0.716	24.62(± 0.107)	29.45
8	0.0922	0.4722	15.5	0.718	25.25(± 0.119)	30.01
9	0.1058	0.4722	17.6	0.824	23.57(± 0.115)	28.03
10	0.0411	0.4722	20.0	0.248	15.53(± 0.049)	14.67
11	0.1228	0.4722	20.4	0.726	21.34(± 0.087)	25.01
12	0.0717	0.4722	12.4	0.698	28.09(± 0.131)	33.25
Objective: 23.32 seconds						

Table 3.7: Output of optimal situation with $c = 57$ seconds

3.5.4 Discussion of results

From the simulation results in the previous subsection we conclude, that the newly found signal scheme with a cycle time of 90 seconds will lead to an enormous improvement of the objective function, compared to the current situation. With the estimates of the traffic intensity used in this chapter, the degree of saturation is in some cases even more than 1. This leads to unacceptable delays. In these cases the queue length will increase constantly during rush-hour. When the rush-hour is finished and the traffic intensities decrease, then the problem will be solved. That is the reason, why it is difficult to compare the new situations with the current situation in a quantitative way. Because, the longer the runlength of the simulation is, the larger the average delay of these over-saturated signals will be. As a result, the larger the objective will be.

The objective will even more improve, when other cycle times are permitted. The optimal cycle time is 57 seconds. In this case an improvement of the objective of 6.23 seconds (= 21.1%) in comparison with the signal scheme with a cycle length of 90 seconds is achieved.

In table 3.6 and 3.7 the approximations of the average delay, based on Webster's formula without correction term are given. These approximations differ sometimes quite a lot from the simulation results. The reason therefore is, that we didn't make use of Webster's formula with correction term. In chapter 4 more information about this formula and better alternatives is given.

3.6 Sensitivity analysis

In this section, we do a sensitivity analysis of the results presented in the previous section. Out of this analysis, we would like to conclude whether the good results are maintained, when the occupation rate of the vehicles at the approaches is varied. This is done, because the signal schemes found in the previous section are based on estimations of the arrival rate and departure rate. Therefore, the effects of an increase and decrease of the arrival rate on the results in case the signal schemes are given as in table 3.6 and 3.7 are investigated. An increase and decrease of the departure rate will give almost the same results. We cannot only see the effect of possible wrong estimations of the arrival and departure rates, but we can also see how these signal schemes will perform in the future, when the amount of traffic will increase.

In the following two tables, the results of this analysis are presented. We have simulated cases for which the arrival rates are decreased with 5% and 10% and increased with 5%, 10%, 20%, 30% and 50%. In these tables the average delay and the objective value are given (in seconds) for the seven different cases as well as for the original case (= 0%).

i	λ_i	μ_i	$g_i(s)$	-10%	-5%	0%	$+5\%$	$+10\%$	$+20\%$	$+30\%$	$+50\%$
2	0.0731	0.4722	21.0	34.35	35.19	36.19	37.34	38.99	43.58	53.58	>300
5	0.0956	0.4722	28.2	28.98	29.48	30.16	30.97	31.94	34.88	40.52	109.66
8	0.0922	0.4722	27.3	29.61	30.23	30.90	31.76	32.82	35.74	40.60	123.39
9	0.1058	0.4722	29.9	28.46	28.93	29.72	30.49	31.90	35.36	44.15	>300
10	0.0411	0.4722	47.0	13.59	13.56	13.60	13.70	13.70	13.82	14.01	14.21
11	0.1228	0.4722	35.1	24.96	25.59	26.27	27.17	28.01	31.33	37.79	>300
12	0.0717	0.4722	21.4	33.27	33.83	34.73	35.52	36.64	40.02	45.94	124.31
Objective:				28.28	28.83	29.55	30.38	31.44	34.60	41.03	>300

Table 3.8: Sensitivity analysis of new situation - $c = 90$ seconds

i	λ_i	μ_i	$g_i(s)$	-10%	-5%	0%	$+5\%$	$+10\%$	$+20\%$	$+30\%$	$+50\%$
2	0.0731	0.4722	15.0	21.04	21.38	21.83	22.45	23.04	24.86	27.87	43.60
5	0.0956	0.4722	16.1	22.33	23.55	24.62	26.17	28.57	35.94	66.83	>1000
8	0.0922	0.4722	15.5	23.02	23.85	25.25	26.99	29.67	38.48	68.40	>1000
9	0.1058	0.4722	17.6	21.43	22.39	23.57	25.45	27.29	36.64	67.25	>1000
10	0.0411	0.4722	20.0	15.33	15.47	15.53	15.57	15.65	15.82	15.95	16.35
11	0.1228	0.4722	20.4	19.45	20.29	21.34	22.94	25.39	33.12	57.32	>1000
12	0.0717	0.4722	12.4	25.61	26.75	28.09	30.02	32.54	40.21	64.51	>1000
Objective:				21.44	22.29	23.32	24.79	26.78	33.66	56.73	>300

Table 3.9: Sensitivity analysis of optimal situation - $c = 57$ seconds

In both cases an increase of the arrival rate with 50% already yields an unstable situation. The higher the occupation rate, the larger the cycle time must be. So in cases where the occupation rate is increased with 50%, even a larger cycle time than 90 seconds has to be taken. In figure 3.3 the objective values of the two different adjustments are plotted as a function of the percentage of the arrival rate. If the arrival rate is increasing, the objective value for c is 57 seconds is indeed increasing much faster, than the objective value for c is 90 seconds. With an increase of 20% the objective values of both cases are practically the same.

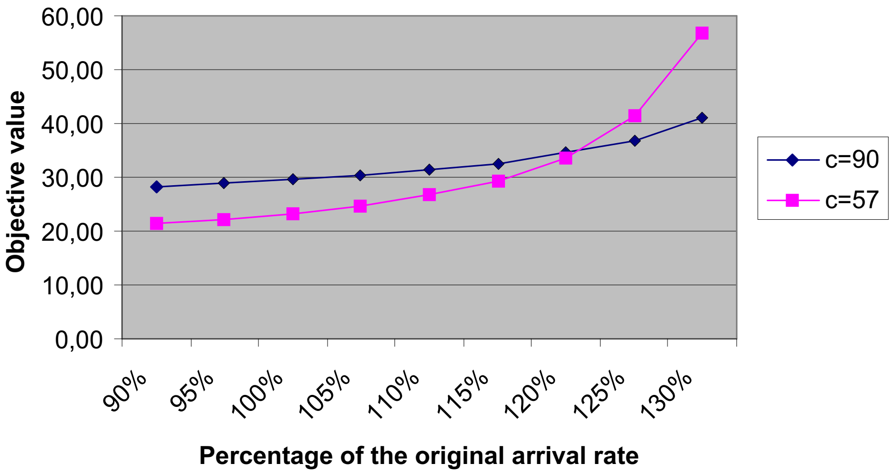


Figure 3.3: The objective value plotted as a function of the percentage of the original arrival rate

CHAPTER 4

APPROXIMATION OF THE AVERAGE DELAY

In 1958 Webster [26] developed an approximating formula for the average delay of vehicles in a fixed-time control as function of the effective green-time, the cycle time, the arrival rate and the departure rate. There has been a broad effort to obtain good approximations for the average delay of vehicles, but until now, researchers didn't succeed. In this chapter we try to find a more accurate approximation of this average delay.

The outline of this chapter is as follows: In the first section a short model description is given. A summary about the literature already published about this subject is presented in section 4.2. In section 4.3, 4.4 and 4.5 some information is given about Webster's, Miller's and Newell's formula, which are most commonly used in mathematical models. The basic idea of our new approximation formula is worked out in section 4.6. In section 4.7 an improvement of the original formula is proposed and a correction term is determined, which is added to the original approximation formula, found in section 4.6. A second improvement of the original approximation formula is outlined in section 4.8. In section 4.9 the results of the new approximation formula are given and compared with the formula of Webster. Finally, in the last section conclusions are drawn and recommendations are made.

4.1 Model description

In this chapter we try to find an approximation formula for the average delay of vehicles in a fixed time control. This formula depends on four parameters: the length of the effective green-time, the cycle time, the arrival rate and departure rate.

In all approximation formulas discussed in this chapter a number of assumptions are made. Vehicles arrive at approach i according to a Poisson process with rate λ_i. This assumption is valid because an isolated intersection is considered, where the vehicles will not arrive in platoons. The arrival rate is constant, that means that no periods of long over- or underload can occur. Vehicles arrive during the whole cycle of length c, but can only drive off during the effective green-time g_i. The effective green-time as well as the cycle time is fixed. The service time (drive-off time) of vehicles on approach i during the effective green-time is constant, with mean $1/\mu_i$. In practice, the drive off time is not constant, but depending on the acceleration and size of the vehicle.

When the signal turns red and the service is not completely finished, two different scenarios can be followed. We have chosen for preemptive resume, that means, the service of the vehicle is preempted, but is continued at the beginning of the next effective green-time.

4.2 Overview of literature

An exact expression for the average delay at fixed-time signals was first derived by Beckman et al. [2] in 1956 with the assumption of a binomial arrival process and deterministic service. But this restrictive assumption of a binomial arrival process reduced the practical usefulness. The difficulty in obtaining exact expressions for this delay, was the reason for a broad effort for signal delay estimation using approximation models and bounds.

The first widely used approximation formula was developed by Webster [26] from a combination of theoretical and numerical simulation approaches. After that, Miller [18] obtained an approximate formula under Poisson arrivals and fixed service time during the effective green-time. Newell [19] aimed at developing an approximation formula for general arrival and departure distributions. He derived a very complex expression for the expected number of waiting vehicles at the beginning of the effective red-time. Newell's expression appears to be more accurate than Miller's. On the other hand Miller's formula yields in some cases better results than Webster's formula.

In the next three sections we will therefore discuss Webster's, Miller's and Newell's approximation formulas.

4.3 Webster's formula

It was found by Webster that the average delay of vehicles at approach i can be approximated by:

$$E[D_i] = \frac{(c-g_i)^2}{2c(1-\rho_i)} + \frac{\rho_i c^2}{2g_i(\mu_i g_i - \lambda_i c)} - 0.65\left(\frac{c}{\lambda_i^2}\right)^{1/3}\left(\frac{\lambda_i c}{\mu_i g_i}\right)^{2+5g_i/c} \tag{4.1}$$

where $E[D_i]$ = average delay per vehicle on approach i
c = cycle time
g_i = effective green time of traffic signal at approach i
λ_i = arrival rate of vehicles at approach i (the interarrival time is exponential)
μ_i = departure rate of vehicles at approach i (the departure time is deterministic)
ρ_i = λ_i/μ_i

and $E[D_i]$, c and g_i are in seconds and λ_i and μ_i are in vehicles per second.

The formula of Webster is based on an M/D/1 system, where the interarrival times and service times are respectively exponentially and deterministically distributed and where the server is periodically on and off alternatingly. The expression for the delay was not derived entirely theoretically. The first two terms have a theoretical meaning, but the last term is purely empirical. The first term of equation (4.1) is the expression for the delay when the traffic can be considered to be arriving deterministically at a uniform rate as a fluid. Although the agreement between computed delays and those derived from this term is fairly good at low arrival rates, it is not so at higher rates. The second term of equation (4.1) makes some allowance for the random nature of the arrivals. It is the average delay for an M/D/1 system, when the service times are stretched with a factor c/g_i. Then we assume that the traffic signal is always green, and the vehicles have a service time with mean $\frac{c}{\mu_i g_i}$. The third term based on simulation results, the so-called correction term, corrects the first two terms. The value of this term is in the range 5 to 15 percent of the original formula of Webster. In figure 4.1 the mean waiting time $E[D_i]$ according to Webster's formula is plotted for certain g_i.

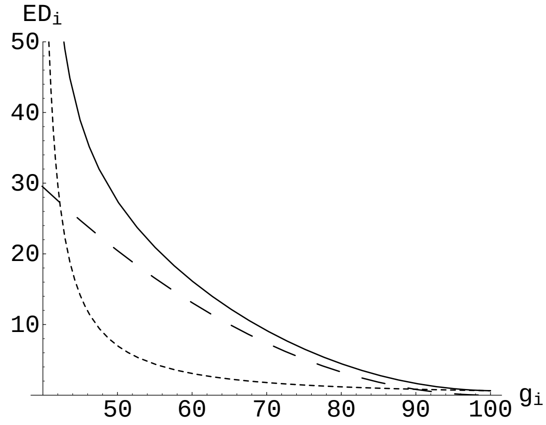


Figure 4.1: Webster's formula of the average delay function as function of the effective green-time g_i for $\lambda_i = 0.19$, $\mu_i = 0.5$ and $c = 100$; the dotted line is the first term of equation (4.1), the dashed line is the second term of equation (4.1) and the solid line is the sum of these two terms.

4.4 Miller's formula

It was found by Miller that the average delay of vehicles at approach i can be approximated by:

$$E[D_i] = \frac{(c-g_i)}{2c(1-\rho_i)}\left((c-g_i)+\frac{2E[X_i^{BR}]}{\lambda_i}+\frac{1}{\mu_i}\left(1+\frac{1}{1-\rho_i}\right)\right) \quad (4.2)$$

where $E[X_i^{BR}]$ = expected number of waiting vehicles at the beginning of the effective red-time

and $E[D_i]$, c and g_i are in seconds and λ_i and μ_i are in vehicles per second.
An expression for $E[X_i^{BR}]$ is given by:

$$E[X_i^{BR}] = \frac{\exp\left[-1.33\sqrt{\mu_i g_i (1-\rho_i^*)/\rho_i^*}\right]}{2(1-\rho_i^*)}$$

where ρ_i^* = degree of saturation given by $(\lambda_i c)/(\mu_i g_i)$

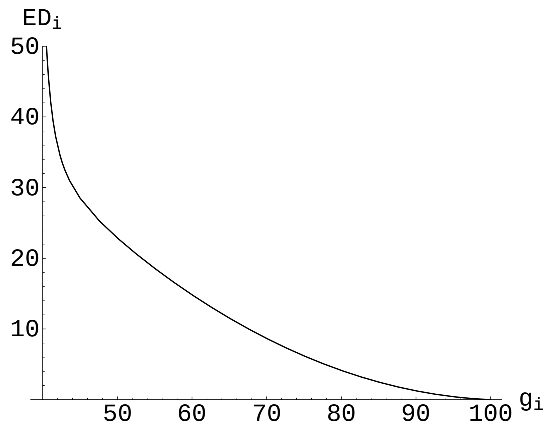


Figure 4.2: Miller's formula of the average delay function as function of the effective green-time g_i for $\lambda_i = 0.19$, $\mu_i = 0.5$ and $c = 100$

4.5 Newell's formula

Newell assumes the same average delay formula as Miller, given in equation (4.2), but gives another approximation for $E[X_i^{BR}]$:

$$E[X_i^{BR}] \quad = \quad \frac{\lambda_i c(1-\rho_i^*)}{\pi} \int_0^{\pi/2} \frac{\tan^2\theta}{-1+\exp[\mu_i g_i (1-\rho_i^*)/(2\cos^2\theta)]} d\theta$$

In figure 4.3 the mean waiting time $E[W_i]$ according to Newell's formula is plotted for certain g_i.

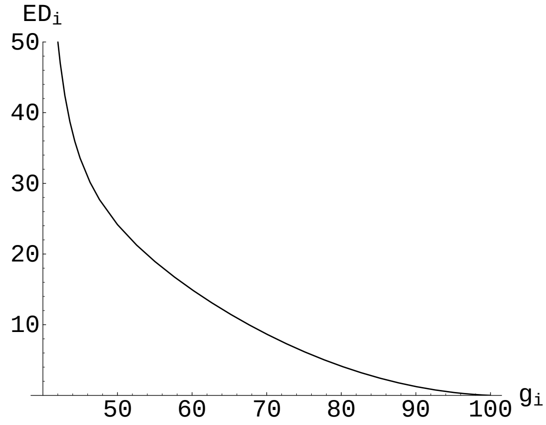


Figure 4.3: Newell's formula of the average delay function as function of the effective green-time g_i for $\lambda_i = 0.19$, $\mu_i = 0.5$ and $c = 100$

Because the expression of Newell for $E[X_i^{BR}]$ is difficult, the formulas of Webster and Miller are used more often nowadays than Newell's approximation formula. Therefore we will compare the results of the new found formula in section 4.9 only with results of Webster and Miller.

4.6 New approximation formula

In this section, the newly developed formula will be derived.
Let the number of waiting vehicles at approach i be X_i. When an expression for the average number of waiting vehicles can be found, the average delay can easily be found with Little's Law [1].

4.6.1 Number of waiting vehicles

During the whole cycle vehicles arrive, but only during the effective green-time they can leave the queue with a constant rate. During the effective green-time the number of waiting vehicles will gradually decrease, as long as there are waiting vehicles.
A discrete-event simulation program has been written to see how the average number of waiting vehicles decreases during the effective green-time. The only two events of this simulation program are:

1. Arrival of a vehicle
2. Measurement of number of waiting vehicles

When a vehicle arrives the total number of waiting vehicles increases with one and a new arriving event will be generated. At regular points of time the total number of vehicles is measured. The events of these measurements are generated at the beginning of the simulation. The system will be simulated until the total simulation time is exceeded. The reliability of the results can be improved by increasing the total number of simulations. In figure 4.4 the results of an example are graphically represented.

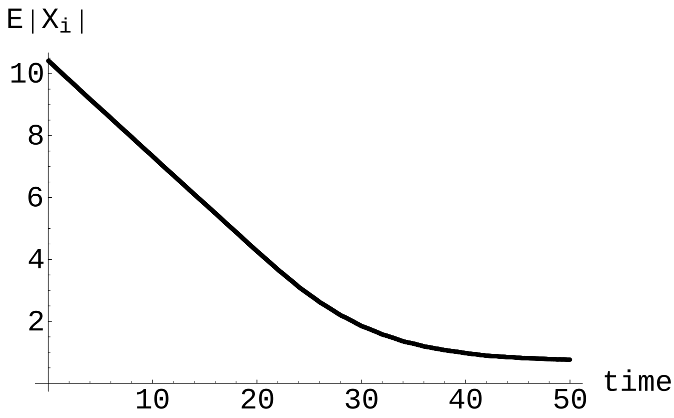


Figure 4.4: The mean number of customers as a function of time (time=0 is the beginning of the effective green-time), for $\lambda_i = 0.19$, $\mu_i = 0.5$, $c = 100$ and $g_i = 50$

4.6.2 Approximation

The curve of figure 4.4 can be approximated by two straight lines. When the signal turns green, the average number of waiting vehicles will decrease from a start-level to another end-level with $\mu_i - \lambda_i$ per second as long as there are waiting vehicles. As can be seen in figure 4.4 the average number of waiting vehicles ($E[X_i]$) is stable after some time, corresponding to the end-level. This stable level l_i, is equal to the expected number of customers in an $M/D/1$ system given by:

$$l_i = \rho_i + \frac{\rho_i^2}{2(1-\rho_i)}$$

When the traffic signal turns red, the total number of waiting vehicles will gradually increase (fluid approach), with an average speed of λ_i vehicles per second until the signal turns green again. At the same time, this reached level is the start-level again. The increase in the average number of waiting vehicles during the effective red-time is equal to:

$$(c - g_i)\lambda_i$$

and the new start-level is then given by:

$$l_i + (c - g_i)\lambda_i$$

The cyclic behavior of the average number of waiting vehicles is depicted in figure 4.5.

The intersection of the two dotted lines is called the relaxation time and is indicated with r_i. This relaxation time is given by:

$$r_i = \frac{(c - g_i)\lambda_i}{\mu_i - \lambda_i}$$

The surface under the diagram from 0 to c seconds, approximated by the straight lines as given in figure 4.5 can be easily computed:

$$\text{Surface} = l_i c + \frac{1}{2} r_i (c - g_i)\lambda_i + \frac{1}{2}(c - g_i)^2 \lambda_i$$

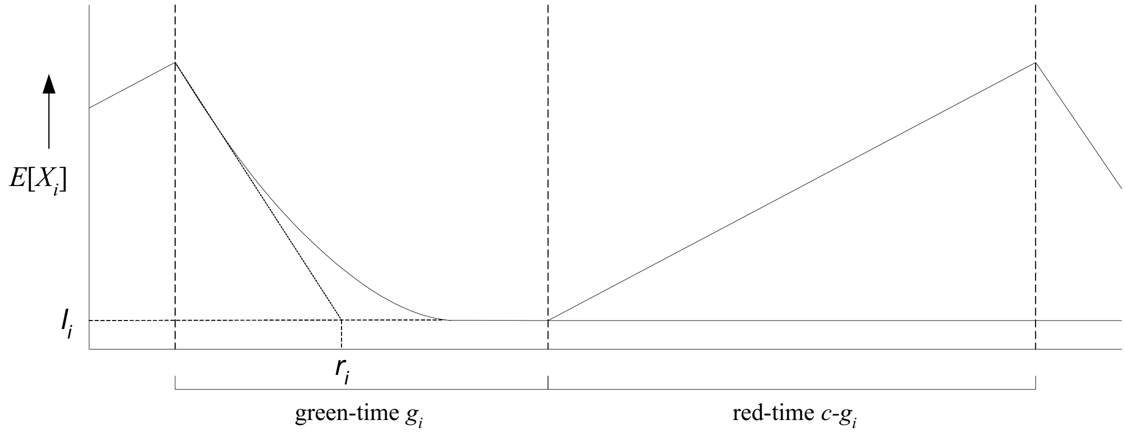


Figure 4.5: Cyclic behavior of the average number of waiting vehicles

The average number of waiting vehicles at approach i is now given by:

$$\begin{aligned} E[X_i] &= \frac{\text{Surface}}{c} \\ &= l_i + \frac{r_i(c-g_i)\lambda_i}{2c} + \frac{(c-g_i)^2\lambda_i}{2c} \\ &= l_i + \frac{(c-g_i)^2\lambda_i^2}{2c(\mu_i-\lambda_i)} + \frac{(c-g_i)^2\lambda_i}{2c} \end{aligned}$$

With Little ($E[X_i] = \lambda_i E[D_i]$), the average delay is now given by:

$$E[D_i] = \frac{l_i}{\lambda_i} + \frac{(c-g_i)^2\lambda_i}{2c(\mu_i-\lambda_i)} + \frac{(c-g_i)^2}{2c}$$

Rewriting yields:

$$E[D_i] = \frac{l_i}{\lambda_i} + \frac{(c-g_i)^2}{2c(1-\rho_i)} \tag{4.3}$$

where the second term of formula (4.3) corresponds to the first term of formula (4.1).

4.7 First improvement of approximation formula

In this section we determine a correction term to improve the approximation formula developed in the previous section. First we mention three important things:

Remark 1

The curve of the average number of waiting vehicles during the effective green-time is approximated by two straight lines. This yields an underestimation of the average number of waiting vehicles resulting in an underestimation of the average delay. When $\mu_i >> \lambda_i$, the approximation of the curve by two straight lines is justified more than in other cases. And as a result the approximation in equation (4.3) will be more accurate than in other cases.

Remark 2

When the system is close to unstability (the degree of saturation is almost 1) the approximation is very inaccurate. For stability of the number of vehicles at approach i, we require that:

$$\lambda_i c \quad < \quad \mu_i g_i$$

Hence if $\mu_i g_i - \lambda_i c$ is close to 0 the approximation is bad. The fluctuation in arrivals is the reason for this. The effective green-time is just long enough, for all vehicles to drive off. When a more than average number of arrivals has taken place, these extra vehicles cannot drive off, during the next effective green-time and will affect the average number of waiting vehicles. In the original formula we do not take into account the random nature and fluctuation of the arrivals of vehicles.

Remark 3

When $g_i \approx c$, the correction term has to be small, because in almost all cases where $g_i \approx c$, the degree of saturation is relatively small. The original formula yields already good results in this case. On the other hand, when g_i is much smaller than c, it will be more likely, that the queue at the end of the effective green-time is not empty because of stochastic reasons (more arrivals during a cycle than the average amount).

In all cases the original approximation formula yields an underestimation of the actual delay. Therefore we want to add a correction term to improve the approximation formula. The correction term we found is based on the three remarks above, where we also have to take in mind, that the unit of this correction term has to be 'seconds'.

As educated guess for a correction term we try:

$$\frac{(c - g_i)\rho_i}{\mu_i g_i - \lambda_i c} \tag{4.4}$$

- When $c - g_i$ is large the correction term has to be large.
- When $\mu_i g_i - \lambda_i c$ is small the correction term has to be large.
- When ρ_i is large the correction term has to be large.

Combining the correction term above and the original approximation formula as in (4.3) yields the following improved approximation:

$$E[D_i] \quad = \quad \frac{l_i}{\lambda_i} + \frac{(c - g_i)^2}{2c(1 - \rho_i)} + \frac{(c - g_i)\rho_i}{\mu_i g_i - \lambda_i c} \tag{4.5}$$

In figure 4.6 we have plotted the absolute difference (the difference between the average delay of the simulation results and the approximation obtained by equation (4.3)) against the correction term in equation (4.4). Indeed a linear relation can be seen between this difference and the correction term. For (almost) all values of the absolute difference the correction term will be too small, as can be seen in this figure.

4.8 Second improvement of approximation formula

We can see the model described in this section as an M/D/1 model with vacations. We define a server vacation as an effective red-time period, as well as periods during the effective green-time when no vehicles are waiting at the queue. From Fuhrmann & Cooper [11] we know:

$$X \quad \overset{d}{=} \quad X^{M/D/1} + X^I \tag{4.6}$$

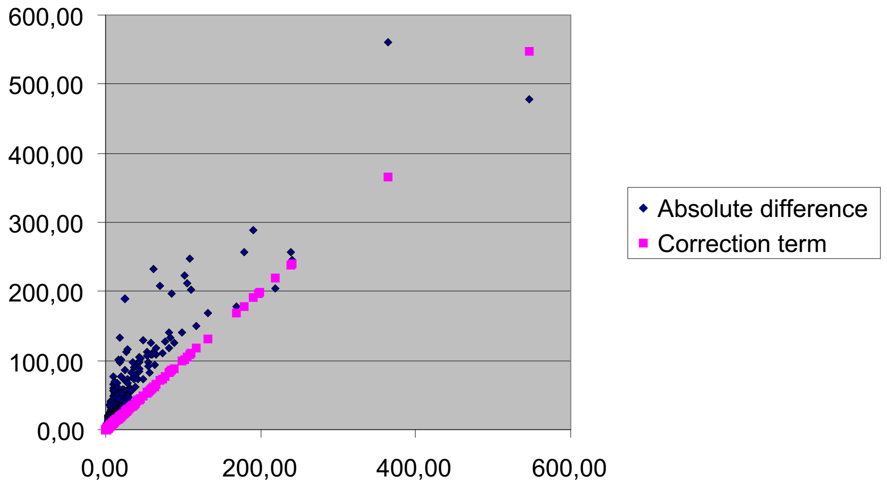


Figure 4.6: Scatterplot of the absolute difference against the correction term given in equation (4.4); on the x-axis the correction term (in seconds); on the y-axis the absolute difference (in seconds)

where X^I is the number of vehicles waiting on an arbitrary vacation period. Furthermore $X^{M/D/1}$ and X^I are mutually independent.
So using equation (4.6), the following equation for the expected number of vehicles at approach i can be found:

$$E[X_i] = \rho_i + \frac{\rho_i^2}{2(1-\rho_i)} + E[X_i^I] \tag{4.7}$$

As said earlier, we distinguish two sorts of server vacations. Let Z_i be the total idle time during the effective green-time. Out of the following balance equation Z_i can be determined easily:

$$\rho_i c = g_i - Z_i \tag{4.8}$$

So to determine $E[X_i^I]$ we distinguish two cases:

$$E[X_i^I] = \frac{Z_i}{Z_i + (c-g_i)} \cdot 0 + \frac{c-g_i}{Z_i + (c-g_i)} \left(E[X_i^{BR}] + \frac{\lambda_i (c-g_i)}{2} \right)$$

with $E[X_i^{BR}]$ the number of waiting vehicles at the beginning of an arbitrary effective red-time. So using the balance equation (4.8) the following formula results:

$$E[X_i^I] = \frac{c-g_i}{c(1-\rho_i)} \left(E[X_i^{BR}] + \frac{\lambda_i (c-g_i)}{2} \right) \tag{4.9}$$

Combining equation (4.7) and (4.9) the following *exact* formula can be obtained:

$$E[X_i] = \rho_i + \frac{\rho_i^2}{2(1-\rho_i)} + \frac{\lambda_i (c-g_i)^2}{2c(1-\rho_i)} + \frac{c-g_i}{c(1-\rho_i)} E[X_i^{BR}] \tag{4.10}$$

It should be observed that the first three terms of this last equation correspond to equation (4.3). The last term, is the so called correction term. An approximation of this term is determined in the rest of this section. The average delay formula is then given by:

$$E[D_i] = \frac{1}{\mu_i} + \frac{\rho_i}{2\mu_i(1-\rho_i)} + \frac{(c-g_i)^2}{2c(1-\rho_i)} + \frac{c-g_i}{\lambda_i c(1-\rho_i)} E[X_i^{BR}] \tag{4.11}$$

Remark

Note that equation (4.11) can be directly derived with PASTA and Little's law [1]. An arriving vehicle finds on average $E[L_i^q]$ vehicles in the queue and each of them has a deterministic service time with mean $1/\mu_i$. A vehicle can arrive during the service of another vehicle. By PASTA we know that the probability a vehicle finds the server busy is equal to ρ_i. Vehicles can also arrive during the effective red-time of the signal and therefore have to wait a residual effective red-time. Again by PASTA we know that the probability a vehicle arrives in the effective red-time is equal to the fraction of time the signal is red during a cycle. This fraction is given by $(c - g_i)/c$. It is possible that an arriving vehicle cannot be served in that same effective green-time (when it arrives during the effective green-time) or in the next effective green-time (when it arrive during the effective red-time). For the average number of effective red-times a vehicle has to wait an expression can be found. With the same argument as for Little the following relation holds: the number of vehicles waiting at the beginning of the effective red-time is equal to the average number of arrivals during a cycle times the average number effective red-times a vehicle has to wait before being served.

So an expression for the average delay can be formulated based on the four terms described above. This expression depends on the average number of vehicles in the queue and again with Little we can derive the same expression for the average delay as in equation (4.11).

Now we only have to determine an approximation of $E[X_i^{BR}]$. Therefore we examine the behavior of $E[X_i^{BR}]$ in heavy and light traffic.

4.8.1 Heavy traffic approximation

In heavy traffic we assume that the effective green-time is on average just long enough to handle all the traffic and let it drive off.

Define $X_{i,n}^{BR}$ as the number of waiting vehicles at the beginning of the n'th effective red-time. Then the following equation holds:

$$X_{i,n+1}^{BR} = X_{i,n}^{BR} + N(c) - \mu_i g_i \tag{4.12}$$

with $N(t)$ a Poisson process with rate λ_i, indicating the number of arrivals during interval $[0, t]$. For the first and second moment of the Poisson Process we know:

$$\begin{aligned} E[N(t)] &= \lambda_i t \\ E[N^2(t)] &= (\lambda_i t)^2 + \lambda_i t \end{aligned}$$

Computing the second moment of $X_{i,n+1}^{BR}$ in equation (4.12) yields the following:

$$E[(X_{i,n+1}^{BR})^2] = E[(X_{i,n,}^{BR})^2] + 2E[X_{i,n}^{BR}]E[N[c] - \mu_i g_i] + E[N(c) - \mu_i g_i]^2$$

Now let n go to infinity. Then $E[(X_{i,n+1}^{BR})^2]$ is equal to $E[(X_{i,n}^{BR})^2]$. So in the formula above these to terms cancel each other when n goes to infinity.

Define $E[X_i^{BR}]$ as $E[X_{i,n}^{BR}]$ with $n \to \infty$. Then we can determine the following expression for $E[X_i^{BR}]$:

$$\begin{aligned} E[X_i^{BR}] &= \frac{1}{2(\mu_i g_i - \lambda_i c)} \left((\lambda_i c)^2 + \lambda_i c + (\mu_i g_i)^2 - 2\lambda_i c \mu_i g_i \right) \\ &= \frac{1}{2(\mu_i g_i - \lambda_i c)} \left(\lambda_i c + (\lambda_i c - \mu_i g_i)^2 \right) \\ &= \frac{\lambda_i c}{2(\mu_i g_i - \lambda_i c)} - \frac{1}{2}(\lambda_i c - \mu_i g_i) \end{aligned} \tag{4.13}$$

In heavy traffic the last term in equation (4.13) can be neglected. So the heavy traffic approximation for the number of vehicles at the beginning of an arbitrary effective red-time is given by:

$$E[X_i^{BR}] \approx \frac{\lambda_i c}{2(\mu_i g_i - \lambda_i c)}$$

4.8.2 Light traffic approximation

In light traffic we assume that the effective green-time is much too long to let the vehicles drive off. After a fraction of the effective green-time the small number of waiting vehicles has driven off. So the number of vehicles at the end of the effective green-time and the beginning of the effective red-time is given by the average number of vehicles in an M/D/1 system.

So the light traffic approximation for the number of vehicles at the beginning of an arbitrary effective red-time is given by:

$$E[X_i^{BR}] \quad \approx \quad l_i = \rho_i + \frac{\rho_i^2}{2(1-\rho_i)}$$

4.8.3 Interpolation

To find a good approximation of the average number of waiting vehicles at the beginning of an arbitrary effective red-time in heavy traffic as well as in light traffic, we can interpolate the results of the heavy and light traffic approximation. In case of heavy traffic the degree of saturation ($\rho_i^* := (\lambda_i c)/(\mu_i g_i)$) will approach 1. On the other hand, in case of light traffic this same variable is almost 0.

So first we approximate $E[X_i^{BR}]$ with ρ_i^* times the heavy traffic approximation plus $1-\rho_i^*$ times the light traffic approximation:

$$\rho_i^* \frac{\lambda_i c}{2(\mu_i g_i - \lambda_i c)} + (1-\rho_i^*) l_i$$

From the results in research not presented here, we have seen that the approximation for $E[X_i^{BR}]$, as given in the equation above is too large. Probably the heavy traffic approximation weighs too much. We can correct this a little bit, by taking the light traffic approximation for $E[X_i^{BR}]$ to be 0 instead of l_i.

So then we approximate $E[X_i^{BR}]$ with ρ_i^* times the heavy traffic approximation:

$$\rho_i^* \frac{\lambda_i c}{2(\mu_i g_i - \lambda_i c)}$$

From this, the correction term (equal to the last term in equation (4.10)) can be easily computed:

$$\rho_i^* \frac{c - g_i}{2(1-\rho_i)(\mu_i g_i - \lambda_i c)} \tag{4.14}$$

In figure 4.7 we have plotted the absolute difference (the difference between the simulation results and the approximation obtained by equation (4.3)) against this last correction term in equation (4.14).

But maybe other weight factors of ρ_i^* yield better results then this first correction term. We examined other correction terms of the form:

$$(\rho_i^*)^a \frac{c - g_i}{2(1-\rho_i)(\mu_i g_i - \lambda_i c)} \tag{4.15}$$

with a a positive integer.

We have tried various values of a and found the best results for $a = 4$.

In figure 4.8 we have plotted the absolute difference against this last correction term in equation (4.15). It should be observed that the correction term in figure 4.8 better corresponds to the absolute difference as the correction term in figure 4.7.

So our new and final approximation formula is given by:

$$E[D_i] \quad = \quad \frac{l_i}{\lambda_i} + \frac{(c-g_i)^2}{2c(1-\rho_i)} + (\rho_i^*)^4 \frac{c - g_i}{2(1-\rho_i)(\mu_i g_i - \lambda_i c)} \tag{4.16}$$

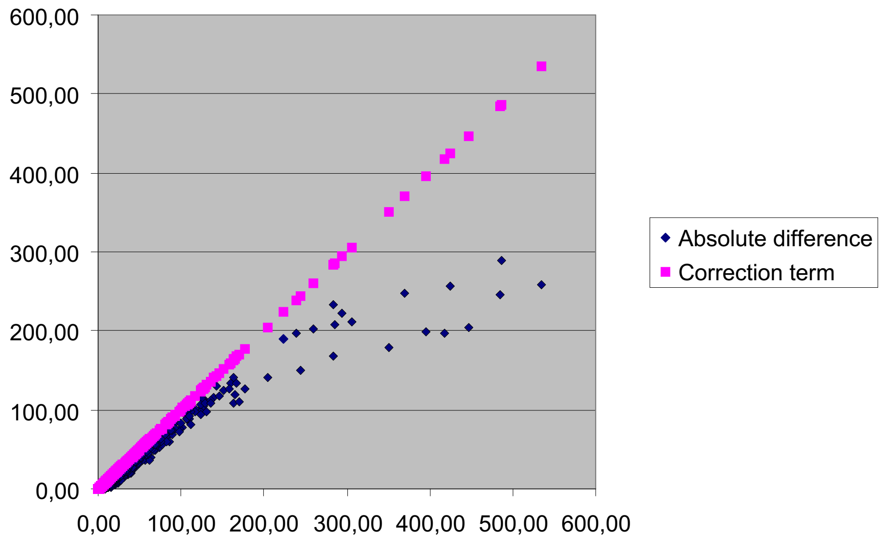


Figure 4.7: Scatterplot of the absolute difference against the correction term given in equation (4.14); on the x-axis the correction term (in seconds); on the y-axis the absolute difference (in seconds)

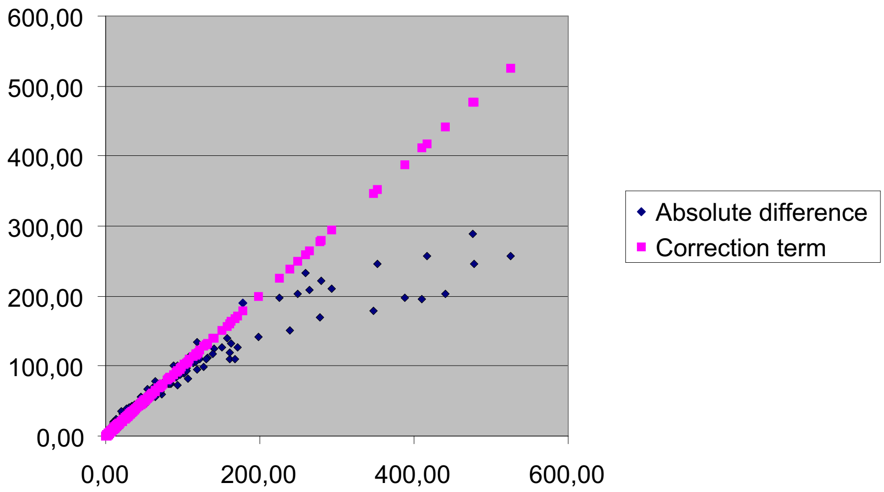


Figure 4.8: Scatterplot of the absolute difference against the correction term given in equation (4.15), with $a = 4$; on the x-axis the correction term (in seconds); on the y-axis the absolute difference (in seconds)

4.9 Results

In this section, the accuracy of the three approximation formulas (the original formula, the first improved formula and second improved formula) is tested by simulation. The simulation program that is used, will be explained in section 3.4. We have simulated 3000 cases, where we have taken the degree of saturation ρ_i^* (given by $\frac{\lambda_i c}{\mu_i g_i}$) randomly between 0 and 1. The values of the other variables are determined in the following order:

- Cycle c: The cycle length (integer value) is randomly chosen between 60 and 140 seconds.
- Departure rate μ_i: The departure rate is randomly chosen between 0.44 and 0.66.

- Effective green-time g_i: The length of the effective green-time is randomly chosen between 5 and $c - 10$ seconds.
- Arrival rate λ_i: The arrival rate is then given by $\rho_i^* \mu_i g_i / c$.

To gain reliable simulation results, for each case the runlength was chosen 24 hours and was repeated 100 times.

For each of the 3000 cases, the width of the 95%-confidence interval will be different. Nevertheless, to give an indication for the accuracy of the results, we will give the simulation results (with 95%-confidence interval) for a light traffic, medium traffic and high traffic case in the following table. In the heavy traffic case, the 95%-confidence interval is very large, so in this cases it is difficult to compare the simulation results with the approximation.

	λ_i	μ_i	c	g_i	ρ_i^*	$E[D_i]$
light	0.027	0.500	100	45	0.123	18.13(± 0.075)
medium	0.194	0.500	100	45	0.864	34.08(± 0.225)
heavy	0.222	0.500	100	45	0.988	156.50(± 12.538)

Table 4.1: Simulation results for a light traffic, medium traffic and heavy traffic case

4.9.1 Accuracy of the approximation

Now that we have found two approximation formulas (given in equation (4.5) and (4.16)) for the average delay of vehicles at a fixed-control intersection, we would like to investigate the accuracy of the original and improved approximations. To investigate this, we use two criteria. The first criterium, is the absolute difference (in seconds) and is given by the absolute value of the difference between the average delay determined by simulation and by approximation. But we will not only investigate the absolute difference, but also the difference with respect to the average delay, the so called difference in terms of percentage (in percents). This difference in terms of percentage is defined by the absolute difference divided by the average delay determined by simulation times 100.

The absolute difference is on average 6.7 (sec) for the original formula (without correction term). For the first improvement and the second improvement this absolute difference is 3.2 (sec) and 2.6 (sec) respectively. So the improved approximation formulas cause a reduction of the absolute difference of 52.0% and 61.1%. The difference in terms of percentage is on average 10.2% for the original approximation. For both improved formulas as given in equation (4.5) and (4.16) this difference in terms of percentage is 4.9% and 2.4%.

The improvement found in section 4.7 yields reasonably good results in cases where the saturation degree is high and the effective green-time is large in comparison with the cycle time. On the other hand, when the saturation degree is high and the effective green-time is small compared with the cycle time, the results of this formula are bad. But when the saturation degree is small, the results are very accurate. This formula yields (even with the added correction term) in most cases still an underestimation of the actual delay.

The improvement found in section 4.8 yields very bad results when the saturation degree approaches 1. In these cases the approximation formula gives an overestimation of the actual delay. When the saturation degree is not that high the approximation is very accurate.

For the original formula as well as for both improved approximation formulas the histogram of the difference in terms of percentage (with class interval 1) is given in figure 4.9. The first three bars represent the number of cases for which the difference in terms of percentage lies between 0% and 1%. The second three bars represent the number of cases for which the difference in terms of percentage lies between 1% and 2%, and so on.

From this histogram we conclude that the second improved formula yields the best results. For 25.2% of all 3000 cases, the difference in terms of percentage is more than 10% for the original formula versus

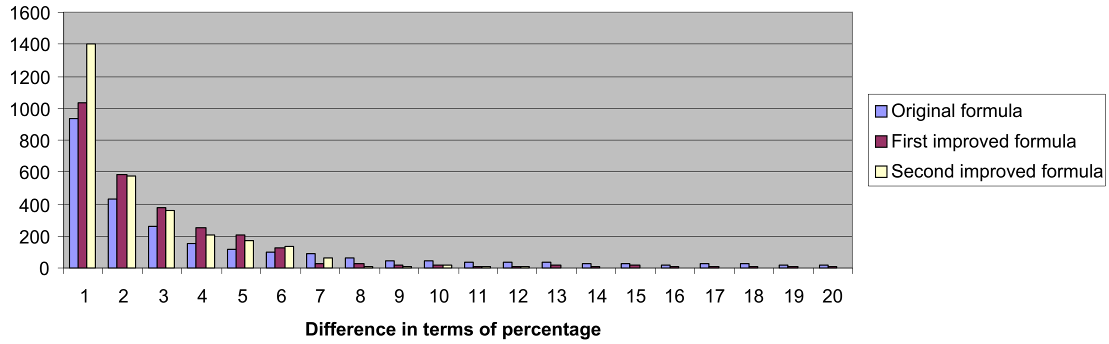


Figure 4.9: Histogram (based upon 3000 cases) of the difference in terms of percentage of the original and improved approximation formulas

11.0% and 1.9% for the first and second improvement respectively. The percentage of all cases for which the difference in terms of percentage is less than 3% is 54.0% for the original formula. For the first and second improvement these percentages are respectively 66.4% and 77.8%.

4.9.2 New formula versus Webster and Miller

To test whether the new approximation formula is more accurate than the existing formula of Webster (with correction term) and Miller, we will compare the results of the three formulas (4.1), (4.2) and (4.16) with the results of simulation.

The average absolute difference of Webster's formula and Miller's formula is on average 3.3 (sec) and 4.0 (sec) respectively, compared with 2.6 (sec) for the newly developed approximation formula. In more than 78% of all cases our approximation formula yields better results than Webster's formula. This percentage is even better when we compare the newly developed formula with Miller and is given by 86%. The average difference in terms of percentage is given by 11.5%, 7.9% and 2.4% for Webster's formula, Miller's formula and the new approximation respectively.

In figure 4.10 the histogram of the difference in terms of percentage is given for the three approximation formulas (with class interval 1). The first three bars represent the number of cases for which the difference in terms of percentage lies between 0% and 1%. The second three bars represent the number of cases for which the difference in terms of percentage lies between 1% and 2%, and so on.

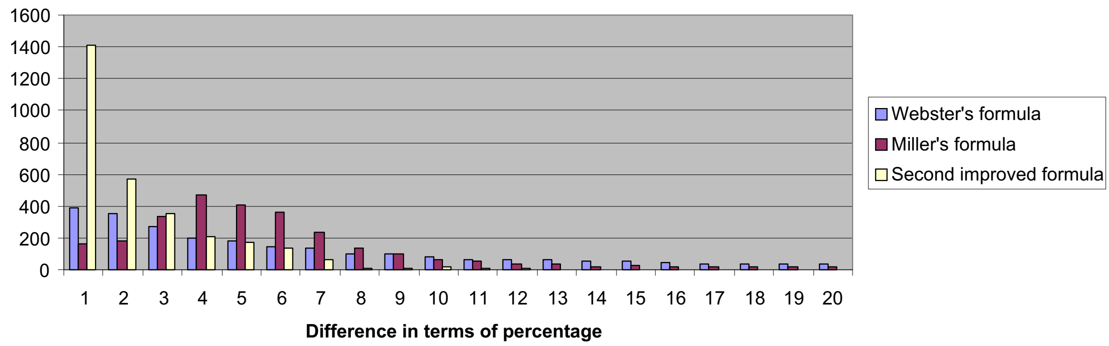


Figure 4.10: Histogram (based upon 3000 cases) of the difference in terms of percentage of the improved approximation formula, Webster's formula and Miller's formula

Overall the newly developed approximation formula yields much better results as can be concluded from the histogram. In 35.1% of all cases Webster's formula had a difference in terms of percentage of more than 10%. This percentage is 18.3% for Miller's formula, compared with 1.9% for the new approximation. So the number of cases that has big difference will be less. On the other hand our approximation yields more accurate results as well. In 33.8% and 22.6% of all 3000 cases Webster's and Miller's formula has a difference in terms of percentage that is less than 3%. For the new approximation formula (as was already given in the previous subsection) this percentage is given by 77.8%.

Especially in case the degree of saturation is almost 1, Webster's formula yields better results than Miller's formula and our new approximation formula. But in these cases the average delay is large, and the difference in percentage of Webster's formula will not deviate that much from the difference in percentage of the newly developed approximation. In case the degree of saturation is not that high ($< 90\%$), the results of our formula are very accurate in contrast with the results of Webster and Miller. Miller's formula is very accurate (compared with both other formulas) in cases where the expression $\mu_i g_i - \lambda_i c$ is large.

All statistics are summarized in table 4.2.

	Webster's formula	Miller's formula	New formula
Average absolute difference	3.3 sec	4.0 sec	2.6 sec
Difference in percentage	11.5%	7.9%	2.4%
Difference in percentage $> 10\%$	35.1%	18.3%	1.9%
Difference in percentage $< 3\%$	33.8%	22.6%	77.8%
Percentage new formula is better than webster	78.7%	86.2%	-

Table 4.2: Summary of statistics of Webster's and Miller's formula and the newly developed approximation formula

4.10 Conclusions and recommendations

From the results in section 4.9, we can conclude that the newly developed approximation formula yields better results than Webster's and Miller's approximation. When the saturation degree is high ($> 98\%$) Webster's formula will give better results than our new formula. When the expression $\mu_i g_i - \lambda_i c$ is large Miller's formula will give better results in most cases. On the other hand, when the saturation degree is not high ($< 90\%$), the new formula yields much better results than both other formulas. In practice there are no adjustments of traffic signals for which the saturation degree is higher than 90%, because in these cases it is most likely that there will form traffic jams. So the newly developed approximation will yield very good results in almost all practical cases.

Probably the newly developed formula can be improved. For example by trying to give a better estimate for $E[X_i^{BR}]$. But we prefer the simplicity of this formula to more accurate results.

Further research can be done on the behavior of approximation formulas of the average delay when the saturation degree is temporarily more than 100% (situation of overload). In these cases the queue will continuously grow. So the average delay will depend on the timelength of overload. On these so-called time-dependent delay models research has been done already, but this may still be improved.

In this chapter we have determined an approximation formula for an isolated intersection. The assumption that the vehicles arrive one by one according to a Poisson process is then validated. But for most intersections within a city this assumption doesn't hold, because the arrival process of vehicles is affected by other intersections. Then vehicles pass the signals in platoons that are separated by a time-interval of which the length is equal to the effective red-time. The platooning effect on average delay functions can be researched as well.

The formula presented in this chapter assumed fixed-time signal control. The introduction of fully-actuated control requires new delay formulas that are sensitive to this process. In subsection 1.4.3 this control is thoroughly described. Delays at these intersections depend on different aspects: maximum and minimum effective green-time and the gap time. Because nowadays most of the intersections use fully-actuated control, research on this topic will be of great importance.

CHAPTER 5

Analyzing exhaustive control

This chapter presents an approximation algorithm to find the effective green-times of traffic signals and average delays of vehicles when exhaustive control is used. With exhaustive service, the effective green-time of an approach lasts until no vehicles are present at that approach anymore. The approaches of one intersection are grouped, so that in each group the vehicles on the approaches have right of way simultaneously. The effective green-time of a group of approaches lasts until no vehicles are present at any of the approaches within that group. In this chapter an algorithm is determined to approximate these effective green-times.

The outline of this chapter is as follows: In the first section a short introduction of the problem and a model description is given. In section 5.2 the concept of the approximation algorithm is explained. The algorithm is based on moment fitting. On the busy period we will fit a distribution. In section 5.3 some background information about the busy period is given. On the busy period we will fit a distribution to approximate the time needed to empty the queues within the same group. The computation of these times is done in section 5.4. We also use moment fitting in section 5.5 to approximate the number of waiting vehicles when the signals turn green. In section 5.6 a first approximation algorithm, based on a one-moment fit is given. An improvement of the developed algorithm, based on a two-moment fit is given in section 5.7. When the first two moments of the effective green-time are known, we can approximate the average delay of vehicles in section 5.8. These approximations are compared with simulation results. Some information about the speed of convergence is given in section 5.9. In section 5.10, the discrete event simulation is described. The results of the approximation developed in this chapter are presented in section 5.11. Finally, in the last section conclusions are drawn and recommendations are made.

5.1 Model description

In this chapter we consider a simple intersection as given in figure 5.1.

The four approaches (2, 5, 8 and 11) are grouped in two pairs: Group G_1 consists of approach 2 and 8; Groups G_2 consists of approach 5 and 11. The groups are served in a cyclic order. In this simple problem, the order in which the groups have right of way will be $\{G_1, G_2, G_1, G_2, \ldots\}$. On more complex intersections, the way of grouping approaches is a problem by itself. The vehicles on the approaches in one group have right of way simultaneously. The effective green-time of approaches in that group, lasts until no vehicles are present at any of the approaches within that group anymore. Because of the exhaustive control, vehicles that arrive during the red period of a traffic signal will be served during the next green period. Vehicles that arrive during a green period, will be served during that same green period.

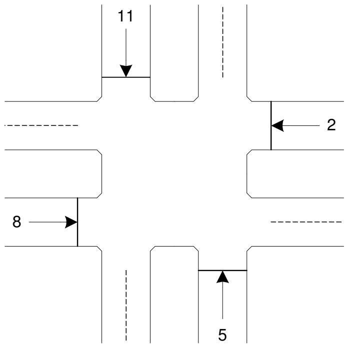


Figure 5.1: Example of a simple intersection

Vehicles on approach i are assumed to arrive according to a Poisson process with intensity λ_i. Because the departure process of vehicles at the approaches is not exactly known, we will investigate three different departure processes in this chapter, namely Exponential ($M/M/1$), Deterministic ($M/D/1$) and Erlang-4 ($M/E_4/1$). The method described in this chapter only uses the first two moments of the departure times, so different departure processes don't result in different methods. For all three different processes the first moment of the departure time is equal to $1/\mu_i$.
When the traffic signals of group G_1 turn red, for safety reasons, a deterministic clearance time $s_{G_1G_2}$ is needed before the traffic signals of group G_2 turn green, and vice versa. In this chapter, we assume (for simplicity) that the clearance times $s_{G_1G_2} = s_{G_2G_1} = 0$. So right after group G_1 was turned red, group G_2 is turned green.
We want to estimate the first and second moment of the effective green-times T_i of approach i. Note that, in this particular example, $T_2 = T_8$ and $T_5 = T_{11}$.

5.1.1 All empty situation

There is a special situation when the traffic signals of a group turn red, but no vehicles are waiting at both approaches of the other group. So, at that moment, no vehicles are waiting at each of the approaches of the intersection. In this case three possible control strategies can be followed:

- The signals of the main group, that means the group which contains the signal with the highest occupation rate, are turned green. When a vehicle at any of the signals of the other group arrives, before a vehicle at any of the signals of the main group arrives, the signals are switched. This strategy is called the *main stream green* strategy.
- All signals are turned red until at any of the approaches on the intersection a vehicle arrives. Then the corresponding signal and the signals in the same group are turned green. This strategy is called the *all red* strategy.
- The signals that have just turned green, stay green until at one of the approaches in that group a vehicle arrives. When no vehicles are present at any of the two approaches, the signal is turned red. This strategy is called the *stay green* strategy.

In this chapter, we have implemented the latter two strategies, but in future research the first strategy can be investigated as well.

5.1.2 Stability

For stability we require that the total number of arriving vehicles during one cycle is strictly less than the number of vehicles that can drive off during the effective green-time. When $s_{G_1G_2} = s_{G_2G_1} = 0$ we state the following theorem (without proof):

Theorem 1. *The system is stable, if and only if the following condition holds:*

$$max(\rho_2, \rho_8) + max(\rho_5, \rho_{11}) \quad < \quad 1 \tag{5.1}$$

where $\rho_i = \lambda_i/\mu_i$.

5.2 Stochastic process

A well known characteristic of exhaustive service is the fact that when a signal turns red, no vehicles are waiting at the approaches anymore. So the number of vehicles waiting at the approach when the signal turns green again, is equal to the number of vehicles that arrived during the effective red-time of that approach. The effective green-time of group G_1 corresponds to the effective red-time of group G_2, and vice versa. So the number of vehicles waiting at the approaches of group G_2, when the signals of this group turn green, is equal to the number of vehicles that arrived at the approaches during the effective green-time of group G_1. When the signals of group G_1 turn green, vehicles leave the approaches 2 and 8. We want to determine the first time when both approaches are empty, that means, when no vehicles are waiting at the approaches to leave. Therefore we will model the process of the number of waiting vehicles at each approach as a continuous time stochastic process. Define the process as $\{X_i(t), t \geq 0\}$ by $X_i(t)$ = the number of waiting vehicles at approach i at time t, $t \geq 0$.

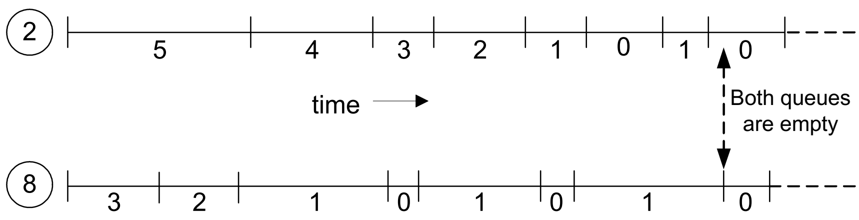


Figure 5.2: Example of state diagram of group G_1

For group G_1, the stochastic process $\{X(t) = (X_2(t), X_8(t)), t \geq 0\}$ has an infinite state space. The same holds for the stochastic process of group G_2, $\{X(t) = (X_5(t), X_{11}(t)), t \geq 0\}$. In figure 5.2 we have given the state diagram of the process $X(t) = (X_2(t), X_8(t))$. The numbers just below the horizontal lines indicate the number of waiting vehicles at different points of time.

5.3 Busy period

We define the first passage time into state $\vec{j}$ when started in state i as:

$$T_{i\vec{j}} \quad = \quad \min\{t \geq 0 : X(t) = \vec{j} | X(0) = i\}$$

where $\vec{j}$ is the target state and $t = 0$ is the beginning of the effective green-time. At $t = 0$ the first departure starts. We want to determine the expected value of this random variable. Let:

$$M_{i\vec{j}}(n) \quad = \quad E[T^n_{i\vec{j}}]$$

be the nth moment of the first passage time. So the expected value of $T_{i\vec{j}}$ is given by:

$$M_{i\vec{j}}(1) \quad = \quad E[T_{i\vec{j}}]$$

When for example, the traffic signals of group G_1 turn green, the process is in state $i = (n_2, n_8)$. When no vehicles are present at the two approaches in group G_1, we have reached the target state $\vec{j} = (0, 0)$ and the signals turn red. So the expected effective green-time is given by:

$$M_{(n_2,n_8)(0,0)}(1) \quad = \quad E[min\{t \geq 0 : X(t) = (0, 0)\} | X(0) = (n_2, n_8)]$$

To approximate the first two moments of the effective green-time we make use of the busy period. Instead of taking the number of waiting vehicles as different states, we introduce another stochastic process $\{Q_i(t), t \geq 0\}$ by $Q_i(t) =$the number of busy periods at approach i at time t, $t \geq 0$. Each vehicle waiting to leave the intersection when the signals have just turned green represents a busy-period. This busy period starts, when the vehicle starts leaving the intersection. During this departure time, vehicles can arrive at the intersection. Instead of joining the normal queue, they form a separate queue. When the vehicle has left the intersection, vehicles in the separate queue have right of way over vehicles in the normal queue. When during the departures of (possible) vehicles in the separate queue, other vehicles arrive they also join the separate queue. The busy period ends when all vehicles of the separate queue have left. Then the next of the vehicles in the normal queue starts service and represents again a busy period, and so on. The number of busy periods is decreasing until no busy periods are present anymore. When the number of busy periods in the other queue is also zero, we can turn the signals from green to red. Otherwise, we wait an *idle period*, before a vehicle arrives. This vehicle brings along one busy period. So once $Q_i(t)$ has reached the state 0, it will alternate between the states 0 and 1.

It is difficult to determine the distribution of the busy period exactly. On the mean and the variance of the busy period, we can fit a distribution. Therefore we try to fit different distributions on the first and on the first two moments. On the first moment we will fit an exponential distribution. Later, we will use a Coxian-2 distribution for a two-moment fit. Therefore, we first have to calculate the first and second moment of the busy and idle period.

Kleinrock [16, Section 5.8] derived the distribution for the length of the busy period for the $M/G/1$ queue. The departure times are distributed with distribution function $F_B(\cdot)$. The first and second moment of the busy period are given by:

$$E[BP_i] = \frac{E[B]}{(1-\rho_i)} \tag{5.2}$$

$$E[BP_i^2] = \frac{E[B^2]}{(1-\rho_i)^3} \tag{5.3}$$

From (5.2) and (5.3), we can easily derive the first and second moment of the busy period of approach i for the different systems.

$M/M/1$

$$E[BP_i] = \frac{1}{\mu_i(1-\rho_i)}$$

$$E[BP_i^2] = \frac{2}{\mu_i^2(1-\rho_i)^3}$$

$M/D/1$

$$E[BP_i] = \frac{1}{\mu_i(1-\rho_i)}$$

$$E[BP_i^2] = \frac{1}{\mu_i^2(1-\rho_i)^3}$$

$M/E_4/1$

$$E[BP_i] = \frac{1}{\mu_i(1-\rho_i)}$$

$$E[BP_i^2] = \frac{5}{4\mu_i^2(1-\rho_i)^3}$$

When one of the approaches within a group reaches state 0, then the system passes through alternating cycles of a busy period (BP), idle period (IP), busy period, idle period, and so on, until both approaches of the same group are in state 0 at the same time. The first and second moment of the idle period of approach i are independent of the departure process of the vehicles and given by:

$$E[IP_i] = \frac{1}{\lambda_i}$$
$$E[IP_i^2] = \frac{2}{\lambda_i^2}$$

5.4 Distribution of the busy period

To compute the expected time (and the second moment of this time) from state $(Q_2(t), Q_8(t)) = (n_2, n_8)$ into state $(Q_2(t), Q_8(t)) = (0, 0)$ we will fit a distribution on the busy period to obtain an approximating distribution. First we will fit an exponential distribution on the first moment of this busy period. Subsequently we will use the Coxian-2 distribution for a two-moment fit on the busy period.

5.4.1 One moment fit

For each state we can fit an exponential distribution on the mean time, spent in that particular state. Then the process $\{Q(t) = (Q_2(t), Q_8(t)), t \geq 0\}$ becomes a Markov process. In figure 5.3, the rate diagram of the stochastic process $\{Q(t) = (Q_2(t), Q_8(t)), t \geq 0\}$ is given. So for approach i holds: $\beta_i = \frac{1}{E[BP_i]}$.

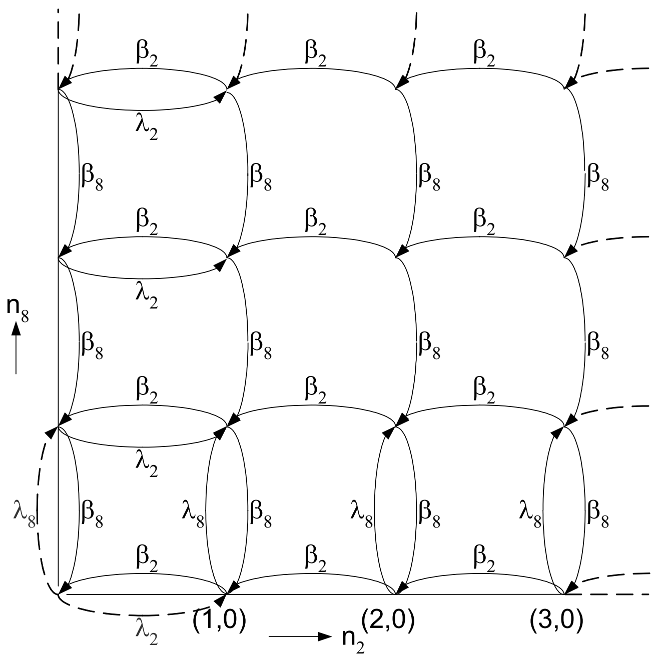


Figure 5.3: Transition rate diagram of the stochastic process $\{Q(t) = (Q_2(t), Q_8(t)), t \geq 0\}$

For each state (n_2, n_8) the expected time until state $(0, 0)$ is reached $(= M_{(n_2,n_8)(0,0)}(1))$ can be calculated by conditioning on the first transition, resulting in the following equations:

$$\begin{aligned}
M_{(n_2,n_8)(0,0)}(1) &= \frac{1}{\beta_2+\beta_8} + \frac{\beta_2}{\beta_2+\beta_8} M_{(n_2-1,n_8)(0,0)}(1) + \frac{\beta_8}{\beta_2+\beta_8} M_{(n_2,n_8-1)(0,0)}(1) \\
M_{(0,n_8)(0,0)}(1) &= \frac{1}{\lambda_2+\beta_8} + \frac{\lambda_2}{\lambda_2+\beta_8} M_{(1,n_8)(0,0)}(1) + \frac{\beta_8}{\lambda_2+\beta_8} M_{(0,n_8-1)(0,0)}(1) \\
M_{(n_2,0)(0,0)}(1) &= \frac{1}{\lambda_8+\beta_2} + \frac{\lambda_8}{\lambda_8+\beta_2} M_{(n_2,1)(0,0)}(1) + \frac{\beta_2}{\lambda_8+\beta_2} M_{(n_2-1,0)(0,0)}(1) \\
M_{(0,1)(0,0)}(1) &= \frac{1}{\lambda_2+\beta_8} + \frac{\lambda_2}{\lambda_2+\beta_8} M_{(1,1)(0,0)}(1) + \frac{\beta_8}{\lambda_2+\beta_8} M_{(0,0)(0,0)}(1) \\
M_{(1,0)(0,0)}(1) &= \frac{1}{\lambda_8+\beta_2} + \frac{\lambda_8}{\lambda_8+\beta_2} M_{(1,1)(0,0)}(1) + \frac{\beta_2}{\lambda_8+\beta_2} M_{(0,0)(0,0)}(1) \\
M_{(1,1)(0,0)}(1) &= \frac{1}{\beta_2+\beta_8} + \frac{\beta_2}{\beta_2+\beta_8} M_{(0,1)(0,0)}(1) + \frac{\beta_8}{\beta_2+\beta_8} M_{(1,0)(0,0)}(1) \\
M_{(0,0)(0,0)}(1) &= 0
\end{aligned}$$

Each of the equations above consists of three parts. For example the first equation is built up as follows: The first part $1/(\beta_2+\beta_8)$ indicates the amount of time you have to wait on average before transition to another state. With probability $\beta_2/(\beta_2+\beta_8)$ and $\beta_8/(\beta_2+\beta_8)$ a transition from state $(Q_2(t), Q_8(t)) = (n_2, n_8)$ to respectively state $(Q_2(t), Q_8(t)) = (n_2-1, n_8)$ and $(Q_2(t), Q_8(t)) = (n_2, n_8-1)$ takes place.

This equation system can be solved recursively. First the values of $M_{(0,1)(0,0)}(1)$, $M_{(1,0)(0,0)}(1)$, $M_{(1,1)(0,0)}(1)$ and $M_{(0,0)(0,0)}(1)$ (the bottom left corner of figure 5.3) are determined out of the last four equations. When these values are known, the edges $M_{(0,n_8)(0,0)}(1)$, $M_{(1,n_8)(0,0)}(1)$, $M_{(n_2,0)(0,0)}(1)$ and $M_{(n_2,1)(0,0)}(1)$ can be solved recursively, using the second and third equation. Finally, the rest of the values $M_{(n_2,n_8)(0,0)}(1)$ can be determined, using the first equation.

5.4.2 Two-moment fit

In the previous subsection, we approximated the length of a busy period by an exponential distribution. To obtain a more accurate approximation, we can use a Coxian-2 distribution for a two-moment fit, in case $c^2_{BP_i} \geq 0.5$. In section 5.3 the expressions for the first two moments of the busy period are given for the three different departure processes. The squared coefficient of variation of the busy period BP is then given by:

$M/M/1$

$$c^2_{BP_i} = \frac{1+\rho_i}{1-\rho_i}$$

$M/D/1$

$$c^2_{BP_i} = \frac{\rho_i}{1-\rho_i}$$

$M/E_4/1$

$$c^2_{BP_i} = \frac{1/4+\rho_i}{1-\rho_i}$$

For the $M/M/1$ this expression is greater than 1, so strictly greater than 0.5 for all values $0 \leq \rho_i < 1$. For the $M/D/1$ and $M/E_4/1$, we have to be careful, because the squared coefficient of variation of the busy period is not always greater than 0.5. For $M/D/1$ and $M/E_4/1$ the criteria $\rho_i > 1/3$ and $\rho_i > 1/6$ must hold.

For the parameters of the Coxian-2 distribution (explained in figure 5.4) the following is suggested by Adan and Resing [1]:

$$\begin{aligned}\mu_{ia} &= 2/E[BP_i]\\ p_{ia} &= 0.5/c^2_{BP_i}\\ \mu_{ib} &= \mu_{ia}p_{ia}\end{aligned}$$

The Coxian-2 distribution consists of one exponential phase with parameter μ_{ia} with probability $(1-p_{ia})$. It consists of two exponential phases with parameter μ_{ia} and μ_{ib} respectively with probability p_{ia}. So besides the number of busy periods waiting at each approach we now also have to keep track of the phase (a or b) of the Coxian-2 distribution. In figure 5.4 the phase diagram for the Coxian-2 distribution is given.

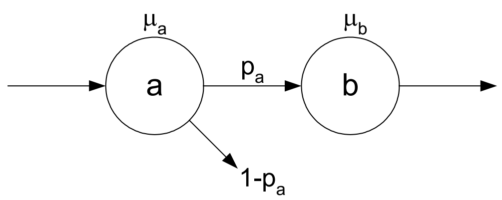


Figure 5.4: Phase diagram for the Coxian-2 distribution

Define now the continuous-time stochastic process $\{Q_i(t), t \geq 0\}$ by $Q_i(t)$= the number of busy periods at approach i at time t, $t \geq 0$ and $\{R_i(t), t \geq 0\}$ by $R_i(t) \in \{a, b\}$= the phase of the Coxian-2 distribution (see figure 5.4). For group G_1, the stochastic process $\{Q(t) = (Q_2(t), Q_8(t)), t \geq 0\}$ has an infinite state space. The same holds for the stochastic process of group G_2, $\{Q(t) = (Q_5(t), Q_{11}(t)), t \geq 0\}$. The stochastic processes $\{R(t) = (R_2(t), R_8(t)), t \geq 0\}$ and $\{R(t) = (R_5(t), R_{11}(t)), t \geq 0\}$ have a finite state space. As a result, the stochastic process $\{S(t) = (Q(t), R(t)), t \geq 0\}$ has an infinite state space as well. The process $\{S(t)\}$ is a Markov process.

Actually, each point in figure 5.3 can be replaced by four points (a, a), (a, b), (b, a), (b, b), representing the four different states of a two-dimensional Coxian-2 distribution. A simplified rate diagram is given in figure 5.5. On the edges (when $n_2 = 0$ or $n_8 = 0$) the rate diagram is more difficult, because an arrival rate λ is involved then.

When the traffic signals of group G_1 turn green, the Markov process is in state $((n_2, n_8), (a, a))$. When no vehicles are present at the two approaches in group G_1, we have reached the target state $\vec{j} := (0, 0)$ and the signals turn red. Note that $(0, 0)$ is a short-hand notation of $((0, 0), (a, a))$.

So the expected effective green-time is given by:

$$M_{((n_2,n_8),(a,a))(0,0)}(1) \quad = \quad E[min\{t \geq 0 : S(t) = (0, 0)\}|S(0) = ((n_2, n_8), (a, a))]$$

In the same way as in the previous section, for each state $((n_2, n_8), (a, a))$, $((n_2, n_8), (a, b))$, $((n_2, n_8), (b, a))$ and $((n_2, n_8), (b, b))$ the expected time until state $(0, 0)$ is reached can be solved iteratively, by formulating a number of equations like the equations in subsection 5.4.1. For example:

$$\begin{aligned}M_{((1,1),(a,a))(0,0)}(1) \quad = \quad & \frac{1}{\mu_{2a}+\mu_{8a}} + \frac{p_2\mu_{2a}}{\mu_{2a}+\mu_{8a}}M_{((1,1),(b,a))(0,0)}(1) + \\ & \frac{(1-p_2)\mu_{2a}}{\mu_{2a}+\mu_{8a}}M_{((0,1),(a,a))(0,0)}(1) + \frac{p_8\mu_{8a}}{\mu_{2a}+\mu_{8a}}M_{((1,1),(a,b))(0,0)}(1) + \\ & \frac{(1-p_8)\mu_{8a}}{\mu_{2a}+\mu_{8a}}M_{((1,0),(a,a))(0,0)}(1)\end{aligned}$$

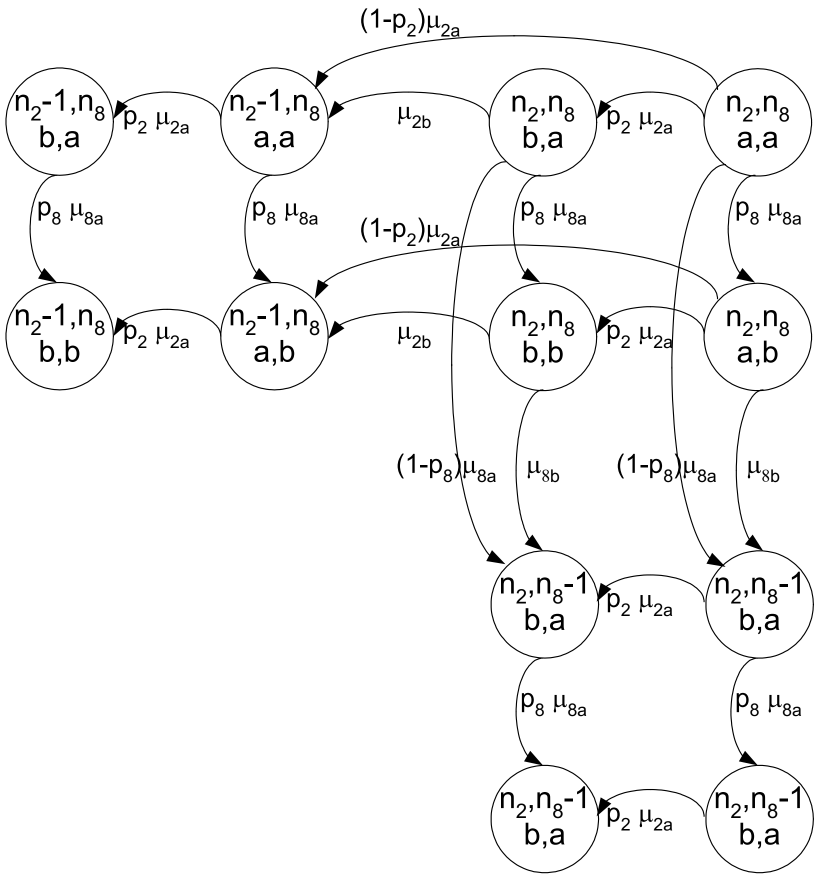


Figure 5.5: Part of the rate diagram of the stochastic process $\{S(t) = ((Q_2(t), Q_8(t)), (R_2(t), R_8(t))), t \geq 0\}$

This can be rewritten as:

$$
\begin{aligned}
M_{((1,1),(a,a))(0,0)}(1) \quad = \quad & \frac{p_2\mu_{2a}}{\mu_{2a}+\mu_{8a}}\left(\frac{1}{\mu_{2a}+\mu_{8a}} + M_{((1,1),(b,a))(0,0)}(1)\right) + \\
& \frac{(1-p_2)\mu_{2a}}{\mu_{2a}+\mu_{8a}}\left(\frac{1}{\mu_{2a}+\mu_{8a}} + M_{((0,1),(a,a))(0,0)}(1)\right) + \\
& \frac{p_8\mu_{8a}}{\mu_{2a}+\mu_{8a}}\left(\frac{1}{\mu_{2a}+\mu_{8a}} + M_{((1,1),(a,b))(0,0)}(1)\right) + \\
& \frac{(1-p_8)\mu_{8a}}{\mu_{2a}+\mu_{8a}}\left(\frac{1}{\mu_{2a}+\mu_{8a}} + M_{((1,0),(a,a))(0,0)}(1)\right)
\end{aligned}
$$

The second moment $M_{((n_2,n_8),(a,a))(0,0)}(2)$ can easily be computed. The second moment is given by the sum of the four probabilities multiplied with the second moment of the terms between brackets. So it is then given by:

$$
\begin{aligned}
M_{((1,1),(a,a))(0,0)}(2) \;=\; & \frac{p_2\mu_{2a}}{\mu_{2a}+\mu_{8a}}\left(\frac{2}{(\mu_{2a}+\mu_{8a})^2}+M_{((1,1),(b,a))(0,0)}(2)+\frac{2M_{((1,1),(b,a))(0,0)}(1)}{\mu_{2a}+\mu_{8a}}\right)+\\
& \frac{(1-p_2)\mu_{2a}}{\mu_{2a}+\mu_{8a}}\left(\frac{2}{(\mu_{2a}+\mu_{8a})^2}+M_{((0,1),(a,a))(0,0)}(2)+\frac{2M_{((0,1),(a,a))(0,0)}(1)}{\mu_{2a}+\mu_{8a}}\right)+\\
& \frac{p_8\mu_{8a}}{\mu_{2a}+\mu_{8a}}\left(\frac{2}{(\mu_{2a}+\mu_{8a})^2}+M_{((1,1),(a,b))(0,0)}(2)+\frac{2M_{((1,1),(a,b))(0,0)}(1)}{\mu_{2a}+\mu_{8a}}\right)+\\
& \frac{(1-p_8)\mu_{8a}}{\mu_{2a}+\mu_{8a}}\left(\frac{2}{(\mu_{2a}+\mu_{8a})^2}+M_{((1,0),(a,a))(0,0)}(2)+\frac{2M_{((1,0),(a,a))(0,0)}(1)}{\mu_{2a}+\mu_{8a}}\right)
\end{aligned}
$$

5.5 Distribution of the number of arrivals

The next step is to find an approximation of the number of vehicles that arrived during the red period. Especially, we want to determine the joint probability mass function ($p_{i,j}(n_i, n_j)$) of the number of arrivals at the approaches in one group. To obtain an approximate distribution, we will fit a distribution on the first two moments of the effective red-time. On the first moment, we can fit an exponential distribution. We have seen by simulation that in almost all cases the squared coefficient of variation ($c^2_{T_i}$) is greater than 0.5 and in some cases even greater than 1. Therefore we will use a Coxian-2 (in case $0.5 < c^2_{T_i} \leq 1$) or a hyperexponential distribution (in case $c^2_{T_i} > 1$) for a two-moment fit. When the squared coefficient of variation is less than 0.5 one has to fit an $E_{k-1,k}$ distribution (a mix of an E_{k-1} and E_k distribution) with a certain set of parameters. Because only in rare cases (when the arrival rates of vehicles are quite similar and small) $c^2_{T_i}$ is smaller than 0.5, we have not implemented the two-moment fit of a mix of Erlang distributions in this chapter.

5.5.1 One moment fit

To find an approximation of the probability mass function $p_{2,8}(n_2, n_8)$ of arriving vehicles at group G_1 during the effective green-time of group G_2, we will fit an exponential distribution on the first moment of the effective green-time of group G_2. The parameter of this distribution is given by $\mu := 1/E[T_5|T_5 > 0] = 1/E[T_{11}|T_{11} > 0]$. The probability mass function $p_{2,8}(n_2, n_8)$ can now easily be derived:

$$
p_{2,8}(n_2,n_8) \;=\; \left(\frac{\lambda_2}{\lambda_2+\lambda_8+\mu}\right)^{n_2}\left(\frac{\lambda_8}{\lambda_2+\lambda_8+\mu}\right)^{n_8}\left(\frac{\mu}{\lambda_2+\lambda_8+\mu}\right)\binom{n_2+n_8}{n_2}, \qquad n_2,n_8\geq 0
$$

An explanation of the formula above is given by the fact that n_2 and n_8 arrivals have taken place at approaches 2 and 8 respectively, before the effective green-time ends. These arrivals can take place in $\binom{n_2+n_8}{n_2}$ different orders.

The probability mass function can also be computed recursively to speed up the algorithm. The recursiveness is obtained by conditioning on the first event that will take place.

$$
\begin{aligned}
p_{2,8}(0,0) &= \frac{\mu}{\lambda_2+\lambda_8+\mu}\\
p_{2,8}(n_2,0) &= \frac{\lambda_2}{\lambda_2+\lambda_8+\mu}p_{2,8}(n_2-1,0), \qquad n_2\geq 0\\
p_{2,8}(0,n_8) &= \frac{\lambda_8}{\lambda_2+\lambda_8+\mu}p_{2,8}(0,n_8-1), \qquad n_8\geq 0\\
p_{2,8}(n_2,n_8) &= \frac{\lambda_2}{\lambda_2+\lambda_8+\mu}p_{2,8}(n_2-1,n_8)+\frac{\lambda_8}{\lambda_2+\lambda_8+\mu}p_{2,8}(n_2,n_8-1), \qquad n_2,n_8\geq 0
\end{aligned}
$$

5.5.2 Two-moment fit: Coxian-2

We have seen by simulation, that in almost all cases $c^2_{T_i|T_i>0} \geq 0.5$. So we can use a Coxian-2 distribution for a two-moment fit. In figure 5.4 the phase diagram for the Coxian-2 distribution is given.
The following is suggested by Adan and Resing [1]:

$$\begin{aligned}
\mu_a &= \frac{2}{E[T_5|T_5>0]} \\
p_a &= \frac{1}{2c^2_{T_5|T_5>0}} \\
\mu_b &= \mu_a p_a
\end{aligned}$$

In this chapter we make use of the set of parameters given by Adan and Resing.
Now (by conditioning), the probability mass function $p_{2,8}(n_2, n_8)$ can easily be derived. With probability $(1 - p_a)$ it comes to an end after the first phase a. This results in the same probability mass functions as the one moment fit. With probability p_a it goes through, up to the second phase b.
Where in the first phase a_2, with $0 \leq a_2 \leq x_2$ and a_8, with $0 \leq a_8 \leq x_8$ vehicles arrived at approach 2 and 8 respectively, in the second phase $x_2 - a_2$ and $x_8 - a_8$ vehicles have to arrive. This results in a double summation, as can be seen below:

$$\begin{aligned}
p_{2,8}(n_2, n_8) = {} & (1-p_a)\left(\frac{\lambda_2}{\lambda_2+\lambda_8+\mu_a}\right)^{n_2}\left(\frac{\lambda_8}{\lambda_2+\lambda_8+\mu_a}\right)^{n_8}\left(\frac{\mu_a}{\lambda_2+\lambda_8+\mu_a}\right)\binom{n_2+n_8}{n_2} + \\
& p_a \sum_{a_2=0}^{n_2}\sum_{a_8=0}^{n_8}\left(\frac{\lambda_2}{\lambda_2+\lambda_8+\mu_a}\right)^{a_2}\left(\frac{\lambda_8}{\lambda_2+\lambda_8+\mu_a}\right)^{a_8}\left(\frac{\mu_a}{\lambda_2+\lambda_8+\mu_a}\right)\binom{a_2+a_8}{a_2} \\
& \left(\frac{\lambda_2}{\lambda_2+\lambda_8+\mu_b}\right)^{n_2-a_2}\left(\frac{\lambda_8}{\lambda_2+\lambda_8+\mu_b}\right)^{n_8-a_8}\left(\frac{\mu_b}{\lambda_2+\lambda_8+\mu_b}\right)\binom{n_2-a_2+n_8-a_8}{n_2-a_2}
\end{aligned} \tag{5.4}$$

Because of the double summation in the formula above, the computation time increases enormously, when n_2 and n_8 increase. We can solve the $p_{2,8}(n_2, n_8)$ recursively, which takes a small fraction of the time to compute these probabilities compared to formula (5.4). Therefore we will first introduce the probability mass function $p_{2,8}(n_2, n_8, i)$ $(i \in \{a, b\})$ of the number of arriving vehicles during phase a and phase a and b together respectively..
The recursiveness is obtained by conditioning on the first event that will take place.

$$\begin{aligned}
p_{2,8}(0,0,a) &= \frac{\mu_a}{\lambda_2+\lambda_8+\mu_a} \\
p_{2,8}(0,0,b) &= \frac{\mu_a}{\lambda_2+\lambda_8+\mu_a}\frac{\mu_b}{\lambda_2+\lambda_8+\mu_b} \\
p_{2,8}(n_2,0,a) &= \frac{\lambda_2}{\lambda_2+\lambda_8+\mu_a}p_{2,8}(n_2-1,0,a) \\
p_{2,8}(n_2,0,b) &= \frac{\lambda_2}{\lambda_2+\lambda_8+\mu_b}p_{2,8}(n_2-1,0,b) + \frac{\mu_b}{\lambda_2+\lambda_8+\mu_b}p_{2,8}(n_2,0,a) \\
p_{2,8}(0,n_8,a) &= \frac{\lambda_8}{\lambda_2+\lambda_8+\mu_a}p_{2,8}(0,n_8-1,a) \\
p_{2,8}(0,n_8,b) &= \frac{\lambda_8}{\lambda_2+\lambda_8+\mu_b}p_{2,8}(0,n_8-1,b) + \frac{\mu_2}{\lambda_2+\lambda_8+\mu_b}p_{2,8}(0,n_8,a) \\
p_{2,8}(n_2,n_8,a) &= \frac{\lambda_2}{\lambda_2+\lambda_8+\mu_a}p_{2,8}(n_2-1,n_8,a) + \frac{\lambda_8}{\lambda_2+\lambda_8+\mu_a}p_{2,8}(n_2,n_8-1,a) \\
p_{2,8}(n_2,n_8,b) &= \frac{\lambda_2}{\lambda_2+\lambda_8+\mu_b}p_{2,8}(n_2-1,n_8,b) + \frac{\lambda_8}{\lambda_2+\lambda_8+\mu_b}p_{2,8}(n_2,n_8-1,b) + \\
&\quad \frac{\mu_b}{\lambda_2+\lambda_8+\mu_b}p_{2,8}(n_2,n_8,a)
\end{aligned}$$

and $p_{2,8}(n_2, n_8)$ follows directly:

$$p_{2,8}(n_2, n_8) = (1 - p_a)p_{2,8}(n_2, n_8, a) + (p_a)p_{2,8}(n_2, n_8, b)$$

For example $p_{2,8}(n_2, n_8, b)$ is determined as follows: the probability that during phase a and b together n_2 and n_8 vehicles have arrived is split up in three different parts:

- The probability that during phase a and b together $n_2 - 1$ and n_8 vehicles have arrived and before phase b ends, a vehicle at approach 2 arrives.
- The probability that during phase a and b together n_2 and $n_8 - 1$ vehicles have arrived and before phase b ends, a vehicle at approach 8 arrives.
- The probability that during phase a n_2 and n_8 vehicles have arrived and phase b ends, before an arrival has taken place at approach 2 or 8.

5.5.3 Two-moment fit: Hyperexponential

We have seen by simulation, that in some cases even $c^2_{T_i|T_i>0} \geq 1$. In these cases we can also use a hyperexponential (H_2) distribution for a two-moment fit instead of a Coxian-2 distribution. A random variable is hyperexponentially distributed, if it is distributed with probability p_a (respectively p_b) as an exponential variable with mean $1/\mu_a$ (respectively $1/\mu_b$). In figure 5.6 the phase diagram for the hyperexponential distribution is given.

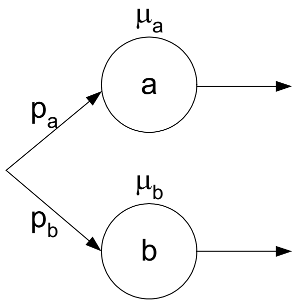


Figure 5.6: Phase diagram for the hyperexponential H_2 distribution

The following is suggested by Tijms [25, Appendix B]:

$$\begin{aligned} p_a &= 1/2\left(1 + \sqrt{\frac{c^2_{T_5|T_5>0} - 1}{c^2_{T_5|T_5>0} + 1}}\right) \\ p_b &= 1 - p_a \\ \mu_a &= \frac{2p_a}{E[T_5|T_5 > 0]} \\ \mu_b &= \frac{2p_b}{E[T_5|T_5 > 0]} \end{aligned}$$

So to derive a closed formula for the probability mass function $p_{2,8}(n_2, n_8)$, we make use of the results derived in subsubsection 5.5.1. The probability mass function $p_{2,8}(n_2, n_8)$ is given by:

$$\begin{aligned} p_{2,8}(n_2, n_8) = \; & p_a\left(\frac{\lambda_2}{\lambda_2+\lambda_8+\mu_a}\right)^{n_2}\left(\frac{\lambda_8}{\lambda_2+\lambda_8+\mu_a}\right)^{n_8}\left(\frac{\mu_a}{\lambda_2+\lambda_8+\mu_a}\right)\binom{n_2+n_8}{n_2} + \\ & p_b\left(\frac{\lambda_2}{\lambda_2+\lambda_8+\mu_b}\right)^{n_2}\left(\frac{\lambda_8}{\lambda_2+\lambda_8+\mu_b}\right)^{n_8}\left(\frac{\mu_b}{\lambda_2+\lambda_8+\mu_b}\right)\binom{n_2+n_8}{n_2} \end{aligned} \quad (5.5)$$

5.5.4 Combination of probability mass functions

With exhaustive service it cannot occur that n_2 or n_8 and n_5 or n_{11} are greater than zero. Therefore it is obvious that we can combine these two probability mass functions to one function $p(n_2, n_8, n_5, n_{11})$, indicating the probability that n_2, n_8, n_5 and n_{11} vehicles are waiting at approaches 2, 8, 5 and 11 respectively when the signals are switched. This can be done in the following way:

$$\begin{aligned} p(n_2, n_8, 0, 0) &= p_{2,8}(n_2, n_8) z_{5,11} \\ p(0, 0, n_5, n_{11}) &= p_{5,11}(n_5, n_{11}) z_{2,8} \\ p(0, 0, 0, 0) &= p_{2,8}(0, 0) z_{5,11} + p_{5,11}(0, 0) z_{2,8} \end{aligned}$$

with

$$z_{2,8} := \mathbb{P}(T_2 > 0) = \sum_{n_2+n_8>0} p(n_2, n_8, 0, 0) + p(0, 0, 0, 0) \frac{\lambda_2 + \lambda_8}{\lambda_2 + \lambda_5 + \lambda_8 + \lambda_{11}} \qquad (5.6)$$

$$z_{5,11} := \mathbb{P}(T_5 > 0) = \sum_{n_5+n_{11}>0} p(0, 0, n_5, n_{11}) + p(0, 0, 0, 0) \frac{\lambda_5 + \lambda_{11}}{\lambda_2 + \lambda_5 + \lambda_8 + \lambda_{11}} \qquad (5.7)$$

with $z_{2,8}$ as the probability that when the signals are switched the signals of group G_1 are turned green (after a possible all empty situation). The same holds for $z_{5,11}$ and group G_2.

5.6 Approximation based on one moment fit

In this section a simple iterative algorithm, developed to approximate the effective green-times in the exhaustive control will be explained. This algorithm is based on a one moment fit of the busy period, as well as a one moment fit of the effective red-time. The two *all empty* strategies, i.e. *stay green* and *all red*, result in slightly different algorithms and will therefore be discussed separately.

5.6.1 Stay green

With the all empty strategy *stay green*, the signals of group G_1 and G_2 have right of way alternatingly. We have determined $M_{(n_2,n_8)(0,0)}(1)$ and $M_{(n_5,n_{11})(0,0)}(1)$ for all different values of n_2, n_8, n_5 and n_{11}. Furthermore we can approximate the probability mass function $p_{2,8}(n_2, n_8)$. Now we can derive an expression for the average effective green-time of traffic signals 2 and 8 in group G_1 ($E[T_2] = E[T_8]$):

$$\begin{aligned} E[T_2] &= \sum_{n_2+n_8>0} M_{(n_2,n_8)(0,0)}(1) p_{2,8}(n_2, n_8) + \\ &\quad p_{2,8}(0, 0) \left(\frac{1}{\lambda_2 + \lambda_8} + \frac{\lambda_2}{\lambda_2 + \lambda_8} M_{(1,0)(0,0)}(1) + \frac{\lambda_8}{\lambda_2 + \lambda_8} M_{(0,1)(0,0)}(1) \right) \end{aligned} \qquad (5.8)$$

The effective green-time of traffic signals 2 and 8 corresponds to the effective red-time of traffic signals 5 and 11. So an approximation for the probability mass function $p_{5,11}(n_5, n_{11})$ can be determined. So the average effective green-time of traffic signals 5 and 11 in group G_2 ($E[T_5] = E[T_{11}]$) is given by:

$$\begin{aligned} E[T_5] &= \sum_{n_5+n_{11}>0} M_{(n_5,n_{11})(0,0)}(1) p_{5,11}(n_5, n_{11}) + \\ &\quad p_{5,11}(0, 0) \left(\frac{1}{\lambda_5 + \lambda_{11}} + \frac{\lambda_5}{\lambda_5 + \lambda_{11}} M_{(1,0)(0,0)}(1) + \frac{\lambda_{11}}{\lambda_5 + \lambda_{11}} M_{(0,1)(0,0)}(1) \right) \end{aligned} \qquad (5.9)$$

This results in a new approximation of the probability mass function $p_{2,8}(n_2, n_8)$ and therefore in a new approximation of $E[T_2]$ and $E[T_8]$, and so on. The iterative algorithm as explained above is outlined below.

Iterative algorithm

The iterative algorithm as explained above is outlined below.

```
Determine values M_{(n_2,n_8)(0,0)}(1);
Determine values M_{(n_5,n_11)(0,0)}(1);
Take E[T_2] = E[T_5] = E[T_8] = E[T_11] =startvalue;
Take E[T_2]_prev = E[T_5]_prev = 0;
WHILE Abs(E[T_2] - E[T_2]_prev) > ε OR Abs(E[T_5] - E[T_5]_prev) > ε DO
    Determine p_{2,8}(n_2, n_8);
    E[T_2]_prev := E[T_2];
    Compute E[T_2] as in equation (5.8);
    Determine p_{5,11}(n_5, n_11);
    E[T_5]_prev := E[T_5];
    Compute E[T_5] as in equation (5.9);
ENDWHILE.
```

5.6.2 All red

In general a cycle consists of two phases. The first phase consists of a (possible) all red period. With probability $p(0,0,0,0)$ the cycle starts with an all red period. This all red period lasts on average $1/(\lambda_2 + \lambda_5 + \lambda_8 + \lambda_8)$. The second phase of the cycle consists of an effective green time of signals 2 and 8 or 5 and 11.

The first moment of the effective green-time of signals 2 and 8, given that the second phase of the cycle consists of an effective green-time of signals 2 and 8 is equal to the first moment of the effective green-time of signals 2 and 8, divided by the probability that the second phase of the cycle is an effective green-time of signals 2 and 8 ($= z_{2,8}$). The same holds for the first moment of the effective green-time of signals 5 and 11.

We start with the following probability mass function:

$$\begin{aligned} p(0,0,0,0) = p(1,0,0,0) = p(0,1,0,0) = p(0,0,1,0) = p(0,0,0,1) = 1/5 \\ p(n_2,n_8,0,0) = p(0,0,n_5,n_{11}) = 0 \quad \text{, with } n_2,n_5,n_8,n_{11} > 0 \end{aligned} \tag{5.10}$$

The first moment ($E[T_2] = E[T_8]$ and $E[T_5] = E[T_{11}]$) of the effective green-time can now be determined as follows:

$$\begin{aligned} E[T_2] &= \sum_{n_2+n_8>0} p(n_2,n_8,0,0) M_{(n_2,n_8)(0,0)}(1) + p(0,0,0,0) \\ &\quad \left(\frac{\lambda_2}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(1,0)(0,0)}(1) + \frac{\lambda_8}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(0,1)(0,0)}(1) \right) \\ E[T_2|T_2>0] &= E[T_2]/z_{2,8} \end{aligned} \tag{5.11}$$

$$\begin{aligned} E[T_5] &= \sum_{n_5+n_{11}>0} p(0,0,n_5,n_{11}) M_{(n_5,n_{11})(0,0)}(1) + p(0,0,0,0) \\ &\quad \left(\frac{\lambda_5}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(1,0)(0,0)}(1) + \frac{\lambda_{11}}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(0,1)(0,0)}(1) \right) \\ E[T_5|T_5>0] &= E[T_5]/z_{5,11} \end{aligned} \tag{5.12}$$

On the first conditional moment of the red-time we will fit an exponential distribution. From this we can find a new and better approximation of the probability mass function of the number of vehicles waiting (described in subsection 5.5.1), at the moment the signals switch. Then we compute the first moment of the effective green-time again, and so on. We will end the iterative algorithm, when the first moment of the effective green-time (for both streams) converges to a specific value. In all tested cases we have seen that, when the system is stable the system always converges to the same specific value.

Iterative algorithm

The iterative algorithm as explained above is outlined below.

```
Determine values M_{(n_2,n_8)(0,0)}(1);
Determine values M_{(n_5,n_11)(0,0)}(1);
Take a start probability mass function as given in equation (5.10);
Take E[T_2] = E[T_5] = startvalue;
Take E[T_2]_prev = E[T_5]_prev = 0;
WHILE Abs(E[T_2] - E[T_2]_prev) > ε OR Abs(E[T_5] - E[T_5]_prev) > ε DO
    E[T_2]_prev := E[T_2] and E[T_5]_prev := E[T_5];
    Compute E[T_2], E[T_5] as in equations (5.11) and (5.12);
    Determine p(n_2, n_8, n_5, n_11) based on one moment fit;
ENDWHILE.
```

5.7 Approximation based on two-moment fit

In this section an improved iterative algorithm, developed to approximate the effective green-times in the exhaustive control will be explained. This algorithm is based on a two-moment fit of the busy period, as well as a two-moment fit of the effective red-time. So, the algorithms discussed in this section are the most accurate ones. The two all empty strategies, i.e. *stay green* and *all red* result in slightly different algorithms and will therefore be discussed separately.

5.7.1 Stay green

We have determined $M_{(n_2,n_8)(0,0)}(1)$, $M_{(n_5,n_{11})(0,0)}(1)$, $M_{(n_2,n_8)(0,0)}(2)$ and $M_{(n_5,n_{11})(0,0)}(2)$ for all different values of n_2, n_8, n_5 and n_{11}. Furthermore we can approximate the probability mass function $p_{2,8}(n_2, n_8)$. Now we can derive an expression for the first and second moment of the effective green-time of traffic signals 2 and 8 ($E[T_2] = E[T_8]$ and $E[T_2^2] = E[T_8^2]$):

$$\begin{aligned} E[T_2] &= \sum_{n_2+n_8>0} M_{(n_2,n_8)(0,0)}(1)\, p_{2,8}(n_2,n_8) + \\ &\quad p_{2,8}(0,0)\left(\frac{1}{\lambda_2+\lambda_8} + \frac{\lambda_2}{\lambda_2+\lambda_8} M_{(1,0)(0,0)}(1) + \frac{\lambda_8}{\lambda_2+\lambda_8} M_{(0,1)(0,0)}(1)\right) \qquad (5.13) \\ E[T_2^2] &= \sum_{n_2+n_8>0} M_{(n_2,n_8)(0,0)}(2)\, p_{2,8}(n_2,n_8) + p_{2,8}(0,0) \\ &\quad \left(\frac{\lambda_2}{\lambda_2+\lambda_8}\left(\frac{2}{(\lambda_2+\lambda_8)^2} + M_{(1,0)(0,0)}(2) + \frac{2M_{(1,0)(0,0)}(2)}{\lambda_2+\lambda_8}\right) + \right. \\ &\quad \left. \frac{\lambda_8}{\lambda_2+\lambda_8}\left(\frac{2}{(\lambda_2+\lambda_8)^2} + M_{(0,1)(0,0)}(2) + \frac{2M_{(0,1)(0,0)}(2)}{\lambda_2+\lambda_8}\right)\right) \qquad (5.14) \end{aligned}$$

The effective green-time of traffic signals 2 and 8 corresponds to the effective red-time of traffic signals 5 and 11. So an approximation for the probability mass function $p_{5,11}(n_5, n_{11})$ can be determined. So the average effective green-time of traffic signals 5 and 11 ($E[T_5] = E[T_{11}]$) and the second moment of this green time ($E[T_5^2] = E[T_{11}^2]$) are given by:

$$\begin{aligned} E[T_5] &= \sum_{n_5+n_{11}>0} M_{(n_5,n_{11})(0,0)}(1)\, p_{5,11}(n_5,n_{11}) + \\ & p_{5,11}(0,0)\left(\frac{1}{\lambda_5+\lambda_{11}} + \frac{\lambda_5}{\lambda_5+\lambda_{11}} M_{(1,0)(0,0)}(1) + \frac{\lambda_{11}}{\lambda_5+\lambda_{11}} M_{(0,1)(0,0)}(1)\right) \quad (5.15) \\ E[T_5^2] &= \sum_{n_5+n_{11}>0} M_{(n_5,n_{11})(0,0)}(2)\, p_{5,11}(n_5,n_{11}) + p_{5,11}(0,0) \\ & \left(\frac{\lambda_2}{\lambda_5+\lambda_{11}}\left(\frac{2}{(\lambda_5+\lambda_{11})^2} + M_{(1,0)(0,0)}(2) + \frac{2M_{(1,0)(0,0)}(2)}{\lambda_5+\lambda_{11}}\right) + \right. \\ & \left. \frac{\lambda_{11}}{\lambda_5+\lambda_{11}}\left(\frac{2}{(\lambda_5+\lambda_{11})^2} + M_{(0,1)(0,0)}(2) + \frac{2M_{(0,1)(0,0)}(2)}{\lambda_5+\lambda_{11}}\right)\right) \quad (5.16) \end{aligned}$$

This results in a new approximation of the probability mass function $p_{2,8}(n_2, n_8)$ and therefor in a new approximation of $E[T_2]$, $E[T_8]$, $E[T_2^2]$ and $E[T_8^2]$ and so on. The iterative algorithm as shown above is outlined below.

Iterative algorithm

The iterative algorithm as explained above is outlined below.

```
Determine values M_{(n_2,n_8)(0,0)}(1);
Determine values M_{(n_5,n_11)(0,0)}(1);
Take E[T_2] = E[T_5] = E[T_8] = E[T_11] =startvalue;
Take E[T_2]_{prev} = E[T_5]_{prev} = 0;
WHILE Abs(E[T_2] - E[T_2]_{prev}) > ε OR Abs(E[T_5] - E[T_5]_{prev}) > ε DO
    Determine p_{2,8}(n_2, n_8);
    E[T_2]_{prev} := E[T_2];
    Compute E[T_2] and E[T_2^2] as in equation (5.13) and (5.14);
    Determine p_{5,11}(n_5, n_11);
    E[T_5]_{prev} := E[T_5];
    Compute E[T_5] and E[T_5^2] as in equation (5.15) and (5.16);
ENDWHILE.
```

5.7.2 All red

In general a cycle consists of two phases. The first phase consists of a (possible) all red period. With probability $p(0,0,0,0)$ the cycle starts with an all red period. This all red period lasts on average $1/(\lambda_2 + \lambda_5 + \lambda_8 + \lambda_{11})$. The second phase of the cycle consists of an effective green time of signals 2 and 8 or 5 and 11.

The first moment of the effective green-time of signals 2 and 8, given that the second phase of the cycle consists of an effective green-time of signals 2 and 8, is equal to the first moment of the effective green-time of signals 2 and 8, divided by the probability that the second phase of the cycle is an effective green-time of signals 2 and 8 ($= z_{28}$). The same holds for the first moment of the effective green-time of signals 5 and 11 and the second moment of both variables.

We start with the following probability mass function:

$$\begin{aligned} p(0,0,0,0) = p(1,0,0,0) = p(0,1,0,0) = p(0,0,1,0) = p(0,0,0,1) = 1/5 \\ p(n_2,n_8,0,0) = p(0,0,n_5,n_{11}) = 0, \qquad \text{with } n_2, n_5, n_8, n_{11} > 0 \end{aligned} \quad (5.17)$$

The first moment ($E[T_2] = E[T_8]$ and $E[T_5] = E[T_{11}]$) and second moment ($E[T_2^2] = E[T_8^2]$ and

$E[T_5^2] = E[T_{11}^2]$) of the effective green-time can now be determined as follows:

$$\begin{aligned}
E[T_2] &= \sum_{n_2+n_8>0} p(n_2, n_8, 0, 0) M_{(n_2,n_8)(0,0)}(1) + p(0,0,0,0) \\
&\quad \left(\frac{\lambda_2}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(1,0)(0,0)}(1) + \frac{\lambda_8}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(0,1)(0,0)}(1) \right) \\
E[T_2|T_2>0] &= E[T_2]/z_{2,8} \qquad (5.18) \\
E[T_2^2] &= \sum_{n_2+n_8>0} p(n_2, n_8, 0, 0) M_{(n_2,n_8)(0,0)}(2) + p(0,0,0,0) \\
&\quad \left(\frac{\lambda_2}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(1,0)(0,0)}(2) + \frac{\lambda_8}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(0,1)(0,0)}(2) \right) \\
E[T_2^2|T_2>0] &= E[T_2^2]/z_{2,8} \qquad (5.19) \\
E[T_5] &= \sum_{n_5+n_{11}>0} p(0, 0, n_5, n_{11}) M_{(n_5,n_{11})(0,0)}(1) + p(0,0,0,0) \\
&\quad \left(\frac{\lambda_5}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(1,0)(0,0)}(1) + \frac{\lambda_{11}}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(0,1)(0,0)}(1) \right) \\
E[T_5|T_5>0] &= E[T_5]/z_{5,11} \qquad (5.20) \\
E[T_5^2] &= \sum_{n_5+n_{11}>0} p(0, 0, n_5, n_{11}) M_{(n_5,n_{11})(0,0)}(2) + p(0,0,0,0) \\
&\quad \left(\frac{\lambda_5}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(1,0)(0,0)}(2) + \frac{\lambda_{11}}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}} M_{(0,1)(0,0)}(2) \right) \\
E[T_5^2|T_5>0] &= E[T_5^2]/z_{5,11} \qquad (5.21)
\end{aligned}$$

On the first and second conditional moment of the red-time we will fit a Coxian-2 distribution with parameters as given by Adan and Resing [1]. From this we can find a new and better approximation of the probability mass function of the number of vehicles waiting (described in subsection 5.5.2), at the moment the signals turn green. Then compute the first and second moment of the effective green-time again, and so on. We will end the iterative algorithm, when the first moment of the effective green-time (for both streams) converges to a specific value.

Iterative algorithm

The iterative algorithm as explained above is outlined below.

Determine values $M_{(n_2,n_8)(0,0)}(1)$;
Determine values $M_{(n_5,n_{11})(0,0)}(1)$;
Take a start probability mass function as given in equation (5.17);
Take $E[T_2] = E[T_5] = startvalue$;
Take $E[T_2]_{prev} = E[T_5]_{prev} = 0$;
WHILE $Abs(E[T_2] - E[T_2]_{prev}) > \epsilon$ OR $Abs(E[T_5] - E[T_5]_{prev}) > \epsilon$ DO
    $E[T_2]_{prev} := E[T_2]$ and $E[T_5]_{prev} := E[T_5]$;
    Compute $E[T_2]$, $E[T_5]$ and $E[T_2^2]$, $E[T_5^2]$ as in equation (5.18),(5.19),(5.20) and (5.21);
    Determine $p(n_2, n_8, n_5, n_{11})$ based on two-moment fit;
ENDWHILE.

5.8 Average delay

For the two different *all empty* strategies, we can now derive the average delay of vehicles at the different approaches, using mean value analysis.

5.8.1 Stay green

Now the average effective green-times have been approximated, we can compute the average delay of vehicles. The average delay $E[D_i]$ (the own 'service time' included) for approach i is given by:

$$E[D_i] = \frac{E[T_j]}{E[T_i]+E[T_j]}E[RT_j] + E[L_i^q]\frac{1}{\mu_i} + \rho_i E[RS_i] + \frac{1}{\mu_i} \quad (5.22)$$

where $E[RT_j]$ is the residual effective red-time of approach j and $E[RS_i]$ is the residual departure time of a vehicle at approach i. $E[RT_j]$ and $E[RS_i]$ are given by:

$$E[RT_j] = \frac{1}{2}E[T_j](C_{T_j^2}+1)$$
$$E[RS_i] = \frac{1}{2}E[S_i](C_{S_i^2}+1)$$

with $C_{T_j^2}$ and $C_{S_i^2}$ the squared coefficient of variation of the effective red-time and the departure time respectively.

Formula (5.22) is determined as follows. The first term is the average delay caused by the fact that a vehicle arrives during the effective red-time, so the effective green-time of the other group. According to PASTA [1] the probability that a vehicle arrives during the effective red-time is equal to the fraction of time the signal is red. This fraction is given by $E[T_j]/(E[T_i]+E[T_j])$ for vehicles arriving at signal i. Then on average vehicles will have to wait a residual effective red-time $E[RT_j]$.

The second and third term of formula (5.22) can be determined as follows: Based on PASTA we know that the average number of vehicles on approach i seen by an arriving vehicle equals $E[L_i^q]$ and each of them has a departure time with mean $1/\mu_i$. Furthermore the vehicle has 'to wait' for its own departure.

The last term of the formula above can again be explained by PASTA. We know that with probability ρ_i there is a departure when a vehicle arrives and the arriving vehicle has to wait until the departure has been completed. On average this time is given by the residual departure time $E[RS_i]$.

As we assumed earlier, vehicles leave the intersection with interdeparture times that are exponentially, deterministic and Erlang-4 distributed. The residual departure time is then given by:

$M/M/1$

$$E[RS_i] = E[S_i]$$

$M/D/1$

$$E[RS_i] = \frac{1}{2}E[S_i]$$

$M/E_4/1$

$$E[RS_i] = \frac{5}{8}E[S_i]$$

From (5.22) and with Little's law [1] ($E[L_i^q]+\rho_i = \lambda_i E[D_i]$), the following formula for the average delay can be found:

$$E[D_i] = \frac{1}{1-\rho_i}\left(\frac{E[T_j]}{E[T_i]+E[T_j]}(\frac{1}{2}E[T_j](C_{T_j^2}+1)) + \rho_i E[RS_i]\right) + \frac{1}{\mu_i}$$

The last two terms, multiplied by $\frac{1}{1-\rho_1}$ form the ordinary $M/G/1$ part. The other terms are extra due to service to the other group.

5.8.2 All red

For the *all red* strategy, the average delay formula is slightly different and given by:

$$E[D_i] \;=\; \frac{E[T_j]}{E[T_i]+E[T_j]+\frac{p(0,0,0,0)}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}}} E[RT_j] + E[L_i^q]\frac{1}{\mu_i} + \rho_i E[RS_i] + \frac{1}{\mu_i}$$

with $E[RT_j]$ and $E[RS_i]$ as given in the previous subsection.

The formula above is determined in the same way as the formula in the previous subsection. A small difference is given by the fact that the average cycle time is given by the average *all red* time (given by $p(0,0,0,0)/(\lambda_2+\lambda_5+\lambda_8+\lambda_{11})$) and the effective green-times of both groups.

Again with Little's law the following approximation formula can be found:

$$E[D_i] \;=\; \frac{1}{1-\rho_i}\left(\frac{E[T_j]}{E[T_i]+E[T_j]+\frac{p(0,0,0,0)}{\lambda_2+\lambda_5+\lambda_8+\lambda_{11}}}\left(\frac{1}{2}E[T_j](C_{T_j^2}+1)\right) + \rho_i E[RS_i]\right) + \frac{1}{\mu_i}$$

5.9 Convergence

The first problem that may arise is that the iterative procedure does not converge at all, that means, the changes in the estimates from iteration to iteration do not get smaller and may even get larger. We cannot prove, that the iterative procedure always converges, but in practice we have always seen it converges.

In most cases the values of the estimates are monotonically increasing from iteration to iteration. It is always increasing because the start probability mass function has a mass of 0.2 in the states $(n_2, n_5, n_8, n_{11}) = \{(0,0,0,0), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)\}$. In some cases the value of the estimates is alternating around the end value. First it is smaller than the end value, subsequently larger, then smaller again, and so on. The absolute deviation of the value of the estimates with the specific end value is monotonically decreasing from iteration to iteration.

The iteration stops, when the deviation between the values of two successive iterations is smaller than a specific value ϵ. We have chosen $\epsilon = 0.001$. Of course, the smaller the epsilon the more accurate the approximation, but the larger the computation time.

In the cases we have examined, the number of iterations varied from 5 to 40. The smaller the occupation rate and the more balanced the occupation rates of the signals within one group, the smaller the number of iterations needed.

The computation time depends on three factors. Of course the speed of the computer affects the computation time. Another factor is the computer program that is used. We have used Mathematica 4.1 to implement the iterative method. The most important factor is given by the number of states $(Q_i(t), Q_j(t)) = (n_i, n_j)$ for which the time until state $(Q_i(t), Q_j(t)) = (0,0)$ is reached is computed. For each case we have computed a square $((Q_i(t), Q_j(t)) = (n_i, n_j)$ with $0 \le n_i \le n_i^{MAX}$ and $0 \le n_j \le n_j^{MAX}$). The values n_i^{MAX} and n_j^{MAX} depend on the value of occupation rates and on the way the occupation rates of the signals within one group are balanced. We have chosen $n_i^{MAX} = n_j^{MAX}$. The values varied from 50 to 150.

The computation time of the cases we have examined varied from 5 minutes to 45 minutes. This seems long, but with a faster computer and a more clever programmed method, this computation time can decrease enormously. With another start probability mass function that corresponds better to the specific case the computation time of this method will decrease as well, because the total number of iterations will decrease.

5.10 Discrete-event simulation

To simulate the dynamic traffic control, we have written a discrete-event simulation with an event scheduling approach. The event scheduling approach concentrates on the events and how they affect the system. The three events that can be distinguished are:

1. Arrival of a vehicle
2. Departure of a vehicle
3. Beginning of the effective green-time

The signals are divided into n groups, where the signals within each group are turned green simultaneously. The server serves the groups in cyclic order $(1, 2, \cdots, n)$. We investigate at each departure whether there are waiting vehicles at signals in group G_k. When at each of the signals in group G_k no vehicles are waiting anymore, the signals are turned red. The signals of the next group in order G_{k+1} are turned green after a clearance time of $s_{G_k,G_{k+1}}$. During this clearance time all signals are turned red and as a result no vehicles can drive off.

We keep track of the time points at which the next events of the different types occur. This Future Event Set (FES) is implemented as a binary search tree to determine the next event efficiently. In a tree, the time points are not ordered in a straight line, like earliest event first, and so on. Instead, the starting time point, called the root, is linked to two other nodes, called its children, and those nodes in turn are linked to other children, and so on. Formally, a tree is either empty, or a root, which is connected to one or more other trees, called the subtrees of the root. The order of all time points in this tree is important. Formally, in a binary search tree the following holds:

- All the time points in the left subtree take place earlier than the time point of the root
- All the time points in the right subtree take place later than the time point of the root
- The left and right subtrees are also binary search trees

We can conclude, that the event that takes place first, is the leftmost node in the tree. We use a binary search tree in order to minimize the distance we have to go to reach any given element. Searching for an element in a binary tree containing n nodes is an $O(\log n)$ process and building the tree in the first place is an $O(n \log n)$ process, if the tree is reasonably well balanced.

The simulation then consists of finding the smallest time point in this tree, setting the current time to this event time and executing the corresponding activities. Here we will describe how these events affect the system and which activities are carried out.

1. Arrival of a vehicle

With an arrival, the vehicle is placed in the waiting queue. For each vehicle in the queue we keep track of the position of the vehicle in the queue as well as the point of time the vehicle joined the queue. When a vehicle arrives, a new arrival at the same signal is simulated. When no vehicles were waiting at the moment the vehicle arrived at the signal, a departure is simulated.

2. Departure of a vehicle

With the departure of a vehicle, we first compute the delay of that particular vehicle. Because we keep track of the points of time vehicles arrive at the queue this can easily be done. All (possible) remaining vehicles shift one place forward in the queue. If there are still vehicles waiting, the signal is not turned red, but a new departure is simulated. With each departure we investigate whether all queues within the group are empty. When this is true, a new event of type 3 (the beginning of the effective green-time) is simulated after the clearance time between group G_k and G_{k+1}. Furthermore, the state of the signals in subset G_k is changed from green to red.

3. Beginning of the effective green-time

At the beginning of the effective green-time of subset G_k, the state of the signals within this subset changes from red to green. When there are waiting vehicles at the queues a new departure is generated and added to the binary search tree.

As stopping criterium for this simulation the runlength is taken. When the current simulation time exceeds this runlength the simulation is ended.

5.11 Results

In this section we will give the results of eight cases that we have investigated. These eight cases represent most of the (extreme) situations that can occur. In table 5.1, an explanation of the abbreviations used in the tables in this section is given.

i	Number of the traffic signal used in figure 5.1.
Int_i	Total number of arriving vehicles per hour ($Int_i = 3600\lambda_i$).
Cap_i	Maximum number of leaving vehicles per hour ($Cap_i = 3600\mu_i$).
$E[T_i]^{sim}$	First moment of the effective green-time, determined by simulation (in seconds).
$E[T_i^2]^{sim}$	Second moment of the effective green-time, determined by simulation.
$E[D_i]^{sim}$	Average delay of vehicles (in seconds), determined by simulation (with 95% confidence interval).
$E[T_i]^{app}$	Approximation of the first moment of the effective green-time (in seconds).
$E[T_i^2]^{app}$	Approximation of the second moment of the effective green-time.
$E[D_i]^{app}$	Approximation of the average delay of vehicles (in seconds).

Table 5.1: Explanation of the abbreviations used in the tables in this section

In the first two tables the results of the all empty strategy *stay green* are given for respectively the exponential and deterministic departure process. Subsequently in the last three tables the results for the all empty strategy *all red* are given when the departure process is respectively exponential, deterministic and Erlang-4.

We have used Mathematica 4.1 to implement the different methods.

5.11.1 Results *stay green*

Exponential service times

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	280	1800	7.29	114.9	6.42(± 0.016)	7.36	116.5	6.44
5	530	1900	6.98	97.6	8.21(± 0.017)	7.08	99.2	8.22
8	700	1900	7.29	114.9	8.42(± 0.017)	7.36	116.5	8.44
11	400	1700	6.98	97.6	8.04(± 0.021)	7.08	99.2	8.04

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	37.24	3098.8	44.16(± 0.528)	37.01	2968.0	43.76
5	930	1900	39.48	3636.8	42.16(± 0.473)	39.20	3474.6	41.85
8	700	1900	37.24	3098.8	39.52(± 0.457)	37.01	2968.0	39.09
11	500	1700	39.48	3636.8	30.95(± 0.310)	39.20	3474.6	30.59

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	15.49	771.0	29.67(± 0.363)	15.32	764.2	29.84
5	930	1900	17.14	967.8	26.74(± 0.317)	17.00	963.8	26.87
8	200	1900	15.49	771.0	18.69(± 0.210)	15.32	764.2	18.78
11	200	1700	17.14	967.8	15.76(± 0.179)	17.00	963.8	15.80

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	13.45	430.0	8.09(± 0.017)	13.42	429.9	7.93
5	250	1900	7.44	104.9	13.93(± 0.047)	7.39	103.8	14.08
8	930	1900	13.45	430.0	8.61(± 0.021)	13.42	429.9	8.60
11	300	1700	7.44	104.9	14.85(± 0.053)	7.39	103.8	15.11

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	830	1800	31.69	2481.9	32.04(± 0.253)	31.67	2208.0	30.47
5	830	1900	26.32	1855.3	39.08(± 0.294)	26.07	1665.1	37.32
8	830	1900	31.69	2481.9	29.64(± 0.246)	31.67	2208.0	28.97
11	200	1700	26.32	1855.3	25.49(± 0.183)	26.07	1665.1	24.07

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	100	1800	7.93	133.3	5.39(± 0.016)	7.94	133.5	5.41
5	200	1900	6.93	92.4	7.13(± 0.017)	6.95	92.7	7.13
8	300	1900	7.93	133.3	5.94(± 0.013)	7.94	133.5	5.95
11	400	1700	6.93	92.4	8.60(± 0.020)	6.95	92.7	8.63

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	200	1800	8.06	142.8	10.92(± 0.056)	8.10	143.2	10.96
5	400	1900	10.63	290.2	7.28(± 0.063)	10.66	290.6	7.23
8	600	1900	8.06	142.8	14.21(± 0.065)	8.10	143.2	14.09
11	800	1700	10.63	290.2	11.85(± 0.078)	10.66	290.6	11.21

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	700	1800	19.10	786.4	21.13(± 0.124)	19.19	786.6	21.07
5	700	1900	19.75	847.8	19.03(± 0.104)	19.84	848.9	18.96
8	700	1900	19.10	786.4	20.23(± 0.116)	19.19	786.6	20.22
11	700	1700	19.75	847.8	20.81(± 0.118)	19.84	848.9	20.73

Deterministic service times

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	280	1800	6.14	59.9	4.76(± 0.007)	6.25	61.0	4.76
5	530	1900	5.89	52.2	5.72(± 0.009)	6.00	53.2	5.71
8	700	1900	6.14	59.9	5.88(± 0.008)	6.25	61.0	5.89
11	400	1700	5.89	52.2	5.70(± 0.009)	6.00	53.2	5.70

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	22.32	927.6	23.61(± 0.172)	24.00	1006.8	23.81
5	930	1900	23.68	1088.0	22.52(± 0.160)	25.48	1180.3	22.73
8	700	1900	22.32	927.6	21.14(± 0.149)	24.00	1006.8	21.33
11	500	1700	23.68	1088.0	16.82(± 0.108)	25.48	1180.3	16.97

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	11.72	311.9	16.74(± 0.111)	12.76	338.4	16.87
5	930	1900	12.95	390.7	15.17(± 0.098)	14.14	430.0	15.12
8	200	1900	11.72	311.9	10.85(± 0.064)	12.76	338.4	10.94
11	200	1700	12.95	390.7	9.42(± 0.054)	14.14	430.0	9.39

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	10.65	204.2	6.23(± 0.009)	11.16	212.7	6.13
5	250	1900	6.26	66.5	8.99(± 0.020)	6.32	66.6	9.04
8	930	1900	10.65	204.2	6.65(± 0.011)	11.16	212.7	6.53
11	300	1700	6.26	66.5	9.68(± 0.019)	6.32	66.6	9.73

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	830	1800	20.36	792.1	17.52(± 0.119)	21.86	851.0	17.51
5	830	1900	17.00	590.9	21.45(± 0.136)	18.18	632.6	21.50
8	830	1900	20.36	792.1	16.67(± 0.108)	21.86	851.0	16.66
11	200	1700	17.00	590.9	14.26(± 0.084)	18.18	632.6	14.30

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	100	1800	7.67	121.7	4.75(± 0.011)	7.67	121.5	4.80
5	200	1900	6.56	72.2	6.77(± 0.018)	6.72	74.4	6.72
8	300	1900	7.67	121.7	5.08(± 0.011)	7.67	121.5	5.14
11	400	1700	6.56	72.2	8.03(± 0.019)	6.72	74.4	7.96

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	200	1800	6.81	71.8	7.07(± 0.021)	6.82	72.2	7.09
5	400	1900	8.75	136.9	5.20(± 0.027)	8.88	138.7	5.06
8	600	1900	6.81	71.8	8.90(± 0.021)	6.82	72.2	8.79
11	800	1700	8.75	136.9	7.58(± 0.026)	8.88	138.7	7.40

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	700	1800	12.44	261.6	11.75(± 0.035)	13.23	297.7	11.81
5	700	1900	12.89	282.4	10.62(± 0.032)	13.69	301.9	11.20
8	700	1900	12.44	261.6	11.26(± 0.033)	13.23	297.7	11.32
11	700	1700	12.89	282.4	11.62(± 0.033)	13.69	301.9	12.25

5.11.2 Results *all red*

Exponential service times

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	280	1800	4.98	72.5	5.44(± 0.017)	4.96	72.1	5.43
5	530	1900	4.78	61.3	7.00(± 0.021)	4.76	60.7	7.01
8	700	1900	4.98	72.5	7.10(± 0.019)	4.96	72.1	7.10
11	400	1700	4.78	61.3	6.88(± 0.020)	4.76	60.7	6.90

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	35.05	2890.4	44.66(± 0.506)	35.10	2789.7	43.26
5	930	1900	37.30	3399.8	42.64(± 0.457)	37.23	3272.0	41.47
8	700	1900	35.05	2890.4	39.94(± 0.435)	35.10	2789.7	38.65
11	500	1700	37.30	3399.8	31.16(± 0.306)	37.23	3272.0	30.31

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	12.58	638.4	28.90(± 0.352)	11.51	572.3	28.50
5	930	1900	13.53	769.5	26.02(± 0.306)	12.52	698.9	25.64
8	200	1900	12.58	638.4	18.19(± 0.202)	11.51	572.3	17.93
11	200	1700	13.53	769.5	15.28(± 0.171)	12.52	698.9	15.09

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	8.16	219.5	5.94(± 0.014)	8.17	220.3	5.96
5	250	1900	4.72	53.0	12.34(± 0.045)	4.59	51.3	12.40
8	930	1900	8.16	219.5	6.40(± 0.017)	8.17	220.3	6.40
11	300	1700	4.72	53.0	13.31(± 0.048)	4.59	51.3	13.34

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	830	1800	27.36	2041.0	31.58(± 0.277)	27.62	2040.9	31.00
5	830	1900	24.28	1641.2	39.37(± 0.328)	23.62	1593.7	38.81
8	830	1900	27.36	2041.0	30.02(± 0.275)	27.62	2040.9	29.48
11	200	1700	24.28	1641.2	25.46(± 0.199)	23.62	1593.7	25.02

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	100	1800	2.54	15.2	3.38(± 0.009)	2.55	15.2	3.38
5	200	1900	2.98	22.9	2.78(± 0.006)	2.98	23.0	2.79
8	300	1900	2.54	15.2	3.67(± 0.008)	2.55	15.2	3.67
11	400	1700	2.98	22.9	3.55(± 0.007)	2.98	23.0	3.55

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	200	1800	5.70	97.1	9.91(± 0.053)	5.59	94.9	9.88
5	400	1900	7.37	186.9	6.12(± 0.026)	7.31	185.1	6.22
8	600	1900	5.70	91.1	12.70(± 0.067)	5.59	94.9	12.68
11	800	1700	7.37	186.9	9.70(± 0.042)	7.31	185.1	9.70

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	700	1800	17.40	719.2	20.91(± 0.107)	17.42	710.1	20.76
5	700	1900	18.00	774.5	18.85(± 0.093)	18.03	767.0	18.71
8	700	1900	17.40	719.2	20.06(± 0.097)	17.42	710.1	19.92
11	700	1700	18.00	774.5	20.60(± 0.100)	18.03	767.0	20.47

Deterministic service times

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	280	1800	4.23	30.9	3.73(± 0.005)	4.32	31.6	3.73
5	530	1900	4.05	26.0	4.46(± 0.007)	4.14	26.7	4.47
8	700	1900	4.23	30.9	4.51(± 0.006)	4.32	31.6	4.51
11	400	1700	4.05	26.0	4.51(± 0.007)	4.14	26.7	4.52

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	21.53	897.1	23.57(± 0.187)	22.61	938.2	23.46
5	930	1900	22.91	1054.3	22.48(± 0.166)	24.01	1100.3	22.46
8	700	1900	21.53	897.1	21.11(± 0.161)	22.61	938.2	21.01
11	500	1700	22.91	1054.3	16.78(± 0.116)	24.01	1100.3	16.78

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	9.32	234.8	15.35(± 0.120)	9.87	247.0	15.29
5	930	1900	10.13	286.1	13.86(± 0.105)	10.76	302.2	13.79
8	200	1900	9.32	234.8	9.96(± 0.072)	9.87	247.0	9.94
11	200	1700	10.13	286.1	8.65(± 0.056)	10.76	302.2	8.62

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	6.91	94.9	3.98(± 0.005)	7.16	99.6	3.99
5	250	1900	3.74	21.1	7.24(± 0.017)	3.83	21.9	7.33
8	930	1900	6.91	94.9	4.15(± 0.006)	7.16	99.6	4.16
11	300	1700	3.74	21.1	7.82(± 0.019)	3.83	21.9	7.93

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	830	1800	18.28	696.4	17.13(± 0.117)	19.24	725.2	16.86
5	830	1900	15.81	546.3	21.05(± 0.129)	16.43	560.3	20.95
8	830	1900	18.28	696.4	16.25(± 0.106)	19.24	725.2	16.03
11	200	1700	15.81	546.3	14.03(± 0.080)	16.43	560.3	13.95

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	100	1800	2.43	7.2	2.70(± 0.004)	2.43	7.3	2.69
5	200	1900	2.87	11.1	2.34(± 0.002)	2.87	11.1	2.34
8	300	1900	2.43	7.2	2.78(± 0.003)	2.43	7.3	2.78
11	400	1700	2.87	11.1	2.83(± 0.002)	2.87	11.1	2.84

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	200	1800	4.59	39.2	5.97(± 0.017)	4.77	40.9	5.96
5	400	1900	6.12	78.2	4.06(± 0.009)	6.36	81.1	4.07
8	600	1900	4.59	39.2	7.33(± 0.022)	4.77	40.9	7.32
11	800	1700	6.12	78.2	5.91(± 0.014)	6.36	81.1	5.92

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	700	1800	11.51	238.2	11.49(± 0.040)	12.21	253.8	11.52
5	700	1900	11.95	257.6	10.38(± 0.032)	12.65	274.2	10.43
8	700	1900	11.51	238.2	11.00(± 0.037)	12.21	253.8	11.05
11	700	1700	11.95	257.6	11.38(± 0.040)	12.65	274.2	11.43

Erlang-4 service times

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	280	1800	4.46	40.7	4.16(± 0.008)	4.55	41.6	4.16
5	530	1900	4.28	34.3	5.10(± 0.012)	4.37	35.1	5.11
8	700	1900	4.46	40.7	5.16(± 0.010)	4.55	41.6	5.16
11	400	1700	4.28	34.3	5.11(± 0.010)	4.37	35.1	5.12

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	24.94	1277.3	28.51(± 0.227)	26.22	1350.3	28.68
5	930	1900	26.52	1500.3	27.21(± 0.201)	27.84	1585.1	27.45
8	700	1900	24.94	1277.2	25.54(± 0.199)	26.22	1350.3	25.67
11	500	1700	26.52	1500.3	20.19(± 0.137)	27.84	1585.1	20.33

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	10.25	321.8	18.64(± 0.182)	10.46	326.6	18.60
5	930	1900	11.11	391.6	16.81(± 0.156)	11.39	399.3	16.76
8	200	1900	10.25	321.8	11.98(± 0.109)	10.46	326.6	11.94
11	200	1700	11.11	391.6	10.26(± 0.089)	11.39	399.3	10.24

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	7.32	124.8	4.47(± 0.007)	7.54	129.6	4.49
5	250	1900	4.03	28.4	8.52(± 0.023)	4.11	29.1	8.60
8	930	1900	7.32	124.8	4.72(± 0.008)	7.54	129.6	4.73
11	300	1700	4.03	28.4	9.21(± 0.021)	4.11	29.1	9.28

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	830	1800	20.91	990.0	20.78(± 0.142)	21.67	1013.5	20.52
5	830	1900	18.21	781.6	24.70(± 0.155)	18.52	785.7	25.56
8	830	1900	20.91	990.0	19.78(± 0.131)	21.67	1013.5	19.51
11	200	1700	18.21	781.6	16.89(± 0.097)	18.52	785.7	16.81

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	100	1800	2.46	9.17	2.86(± 0.005)	2.47	9.2	2.86
5	200	1900	2.91	14.02	2.45(± 0.003)	2.91	14.0	2.46
8	300	1900	2.46	9.17	3.01(± 0.005)	2.47	9.2	3.00
11	400	1700	2.91	14.02	3.01(± 0.003)	2.91	14.0	3.02

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	200	1800	4.92	52.4	6.95(± 0.027)	5.07	54.0	6.94
5	400	1900	6.51	103.4	4.59(± 0.012)	6.70	106.5	4.61
8	600	1900	4.92	52.4	8.66(± 0.033)	5.07	54.0	8.65
11	800	1700	6.51	103.4	6.85(± 0.023)	6.70	106.5	6.87

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	700	1800	13.14	338.2	13.80(± 0.052)	13.86	358.2	13.86
5	700	1900	13.60	364.7	12.46(± 0.043)	14.36	386.9	12.53
8	700	1900	13.14	338.2	13.21(± 0.050)	13.86	358.2	13.29
11	700	1700	13.60	364.7	13.64(± 0.053)	14.36	386.9	13.72

5.11.3 Discussion of results

We only have given numerical results of the approximation based on a two-moment fit of the busy period as well as a two moment fit of the effective red-time. No exact results were given of the approximation based on a one moment fit. For low occupation rates the one moment fit approximation yields reasonably good results. For example, for case 1, 6 and 7 the approximation differs less than 3% from the simulation results. For high occupation rates the results are not accurate enough. The deviation can run to more than 30%.

From all the results in this section we can conclude that the approximation described in this chapter is very accurate. In table 5.2 we can find the deviation (in percents) of the approximation of the first and second moment of the effective green-time from the simulation results for the eight different cases. So for each case the approximation of the effective green-time of group G_1 (Signal 2 and 8) and G_2 (Signal 5 and 11) is compared with the simulation results.

		Stay green				All red						
		Exp		Det		Exp		Det		Erl4		Total
Case		Dev1	Dev2	Dev1	Dev2	Dev1	Dev2	Dev1	Dev2	Dev1	Dev2	Dev
1:	G_1	0.96	1.39	1.79	1.84	0.40	0.55	2.13	2.27	2.02	2.21	1.66
	G_2	1.43	1.64	1.87	1.92	0.42	0.98	2.22	2.69	2.10	2.33	
2:	G_1	0.62	4.22	7.53	8.54	0.14	3.48	5.02	4.58	5.13	5.72	4.50
	G_2	0.71	4.46	7.60	8.48	0.19	3.76	4.80	4.36	4.98	5.65	
3:	G_1	1.10	0.88	8.87	8.50	8.51	10.35	5.90	5.20	2.05	1.49	5.32
	G_2	0.82	0.41	9.19	10.06	7.46	9.17	6.22	5.63	2.52	1.97	
4:	G_1	0.22	0.02	4.79	4.16	0.12	0.36	3.62	4.95	3.01	3.85	2.23
	G_2	0.67	1.05	0.96	0.15	2.75	3.21	2.41	3.79	1.99	2.46	
5:	G_1	0.06	11.04	7.37	7.44	0.95	0.00	5.25	4.14	3.63	2.37	4.09
	G_2	0.95	10.25	6.94	7.06	2.72	2.89	3.92	2.56	1.70	0.52	
6:	G_1	0.13	0.15	0.00	0.16	0.39	0.00	0.00	1.39	0.41	0.33	0.48
	G_2	0.29	0.32	2.44	3.05	0.00	0.44	0.00	0.00	0.00	0.14	
7:	G_1	0.50	0.28	0.15	0.56	1.93	4.17	3.92	4.34	3.05	3.05	1.93
	G_2	0.28	0.17	1.49	1.31	0.81	0.96	3.92	3.71	2.92	3.00	
8:	G_1	0.47	0.03	6.35	13.80	0.11	1.27	6.08	6.55	5.48	5.91	4.24
	G_2	0.46	0.13	6.21	6.91	0.17	0.97	5.86	6.44	5.59	6.09	
Total		0.60	2.28	4.60	5.25	1.69	2.66	3.83	3.91	2.91	2.94	

Table 5.2: Deviation (in percents) of the approximation of the first two moments of the effective green-time from the simulation results

In table 5.2 can be seen that the developed approximation for the *stay green* strategy with an exponential departure process yields the best results, with an average deviation of 0.60% and 2.28% for the first and second moment respectively. On the other hand, when a deterministic departure process is assumed, the deviation comes to 4.60% and 5.25% for respectively the first and second moment of the effective green-time. In some cases (for example case 3), the deviation of the first moment is even almost 10%. The higher the occupation rates of the approaches the larger the deviation will be. But this is not the only reason for a large deviation. When the occupation rates of the two approaches in the same group differ quite a lot, then the deviation will be large as well.

For the strategy *all red* we can remark other things. The overall results for the three different departure processes don't differ that much. The exponential departure process yields the best results, followed by the Erlang-4 departure process. The worst results are once more achieved, when the departure process is deterministic. So the higher the coefficient of variation of the departure process, the smaller the deviation will be. The departure process influences the accuracy of the approximation the most, when the occupation rates of the approaches are high. For example look at case 6 and 8. The occupation rate of the approaches of case 6 is low, where the occupation rate of approaches of case 8 is high. For all three different departure processes, the results of case 6 are very accurate (deviation of less than 2%). This in contrast with the results of case 8, where the occupation rate for all approaches is high.

For case 3 the results are very inaccurate (especially with the *all red* strategy and exponential departure process). In the next subsection we will investigate whether the use of other two-moment fits will yield better results for case 3, when the departure process is exponential and with all empty strategy *all red*. As you can see in the table above the deviation of the first and second moment of the effective green-time is almost 10%, which is extremely bad.

5.11.4 Special case

In this subsection we consider one of the cases that is investigated earlier in this section. The input and output of this case is given in table 5.3. The departure process is exponential. The deterministic and Erlang-4 departure processes already yield reasonably good results and will therefore not be investigated here.

In this particular case, the occupation rates of the two approaches that have right simultaneously differ a lot. As can be seen in table 5.2, the approximation of the effective green-time varies enormously from the simulation results. This deviation is almost 10%.

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	12.58	638.4	28.90($\pm$ 0.352)	11.51	572.3	28.50
5	930	1900	13.53	769.5	26.02($\pm$ 0.306)	12.52	698.9	25.64
8	200	1900	12.58	638.4	18.19($\pm$ 0.202)	11.51	572.3	17.93
11	200	1700	13.53	769.5	15.28($\pm$ 0.171)	12.52	689.9	15.09

Table 5.3: Results of special case with original two-moment fit

Here we investigate whether the choice of moment fit influences the accuracy of the approximation. Instead of using a Coxian-2 distribution for a two-moment fit (where the parameters are given by Adan and Resing [1]) on the effective red-time, we will investigate the use of a hyperexponential distribution and another choice for the Coxian-2 parameters given by Tijms [25, Appendix B].

Two-moment fit: Hyperexponential

The following is suggested by Tijms [25, Appendix B]:

$$\begin{aligned} p_a &= 1/2\left(1+\sqrt{\frac{c_{T_5}^2-1}{c_{T_5}^2+1}}\right) \\ p_b &= 1-p_a \\ \mu_a &= \frac{2p_a}{E[T_5]} \\ \mu_b &= \frac{2p_b}{E[T_5]} \end{aligned}$$

The results of the approximation are given in table 5.4.

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	12.58	638.4	28.90(± 0.352)	11.30	561.8	28.52
5	930	1900	13.53	769.5	26.02(± 0.306)	12.30	687.1	25.66
8	200	1900	12.58	638.4	18.19(± 0.202)	11.30	561.8	17.95
11	200	1700	13.53	769.5	15.28(± 0.171)	12.30	687.1	15.10

Table 5.4: Results of special case with hyperexponential two-moment fit

Two-moment fit: Coxian-2

The other choice for the Coxian-2 parameters is given by:

$$\begin{aligned}
\mu_a &= \frac{2}{E[T_5]}\left(1+\sqrt{\frac{c_{T_5}^2-1/2}{c_{T_5}^2+1}}\right) \\
\mu_b &= \frac{4}{E[T_5]}-\mu_a \\
p_a &= \frac{\mu_b}{\mu_a}(\mu_a E[T_5]-1)
\end{aligned}$$

This particular Coxian-2 density has the remarkable property that its third moment is also the same as that of the gamma density with the same mean and squared coefficient of variation. According to Tijms [25, Appendix B], this normalization is a natural one in many applications.

The results of the approximation are given in table 5.5.

i	Int_i	Cap_i	$E[T_i]^{sim}$	$E[T_i^2]^{sim}$	$E[D_i]^{sim}$	$E[T_i]^{app}$	$E[T_i^2]^{app}$	$E[D_i]^{app}$
2	780	1800	12.58	638.4	28.90(± 0.352)	12.34	614.9	28.47
5	930	1900	13.53	769.5	26.02(± 0.306)	13.33	745.1	25.61
8	200	1900	12.58	638.4	18.19(± 0.202)	12.34	614.9	17.91
11	200	1700	13.53	769.5	15.28(± 0.171)	13.33	745.1	15.07

Table 5.5: Results of special case with other Coxian-2 two-moment fit

The use of a hyperexponential distribution for a two-moment fit doesn't yield better results (see table 5.4). The results are even a little worse. On the other hand, the other choice for the parameters of the Coxian-2 distribution yields results that have a deviation of 1.91% and 1.48% for the first moment of the effective green-time and a deviation of 3.68% and 3.17% for the second moment of the effective green-time (see table 5.5), which is much better than the results in table 5.3.

5.12 Conclusions and recommendations

From all the results in the previous section we can conclude that the approximation described in this chapter is very accurate. Especially when we assume that the departure process of vehicles at the signals is exponentially distributed, the deviation is small. On the other hand when we assume that the departure process is deterministic, the deviations can be large. In reality the departure process is a mixture of these two extreme cases (exponential and deterministic). Therefore, we have also tested for the *all red* strategy the approximation method with an Erlang-4 departure process. The accuracy of these results lies between the results of the exponential and deterministic departure process.

In all cases where we made use of a Coxian-2 distribution for a two-moment fit, we used the parameter set proposed by Adan and Resing [1]. In special cases when the occupation rates of approaches that have right of way simultaneously differs a lot you can probably better make use of the parameter set proposed by Tijms [25]. This particular Coxian-2 density has the remarkable property that its third moment is also the same as that of the gamma density with the same mean and squared coefficient of variation. Maybe all results would improve when we make use of this set of parameters for the Coxian-2 distribution instead of the other one.

In this chapter we investigated the approximation for a simple intersection as given in figure 5.1 and with clearance times equal to zero. The method described in this chapter can be extended to larger and more comprehensive intersections.

In some cases, when the squared coefficient of variation is less than 0.5, we cannot make use of a Coxian-2 distribution for a moment fit. Instead of the Coxian-2 distribution, we have to fit an $E_{k-1,k}$ distribution (a mix of an E_{k-1} and E_k distribution) with a certain set of parameters. Because only in rare cases $c^2_{T_i}$ was smaller than 0.5, we didn't implement the two-moment fit of a mix of Erlang distributions in this chapter.

CHAPTER 6

FULLY-ACTUATED CONTROL SYSTEM

In this chapter a fully-actuated control system is considered. This type of control system is nowadays used on most of the intersections around the world. We use the *k*-limited polling system to model this type of control and to find eventually a good fully-actuated control of the traffic signals on a single isolated intersection.

The outline of this chapter is as follows: In section 6.1 some background information is given about polling systems and especially the *k*-limited polling system. A detailed model description is presented in section 6.2. In section 6.3 an approximation of the average delay is proposed and this approximation is used to optimize the *k*-values for the *k*-limited strategy. The *k*-limited strategy can be used to determine settings for the fully-actuated traffic control system using a simple heuristic. This heuristic is explained in section 6.4. In the next section the heuristic is further developed. The program, that is written to simulate the fully-actuated traffic control is explained in section 6.6. We have developed an Excel tool, to implement the ideas in this chapter. In section 6.7 this tool is demonstrated. Subsequently, the tool is used for determining a fully-actuated control system of an intersection in the city of Eindhoven in section 6.8. In the same section we will compare different controls with each other by simulation. Finally, in the last section conclusions are drawn.

6.1 Introduction

Polling systems are a class of multi-queue systems attended by a single server, visiting the queues one at a time, cf. figure 6.1. Such systems are encountered very frequently in communications and computer systems. Moving from one queue to another, the server typically incurs a non-negligible switch-over time. The queues are assumed to have infinite buffer capacity. As usual in the polling literature, customers are assumed to arrive according to Poisson processes. Having no detailed information about the characteristics of the arrival process, the assumption of Poisson arrival processes is quite often a reasonable approximation, which makes the analysis easier. Each of the queues has its own Poisson arrival process. These arrival processes are mutually independent. The service process of each of the queues is stochastic, but independent of the service process of the other queues.

In the present chapter, we consider a polling system with a *k*-limited service strategy. Under *k*-limited service, when visiting a queue, the server works until a prespecified number of *k* customers has been served, or the queue becomes empty, whichever occurs first and then moves to the next queue. The problem is how the limits *k* should be set as to minimize the weighted average delay.

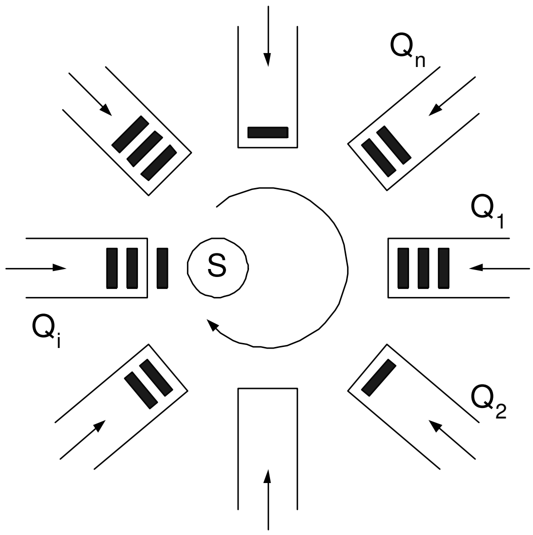


Figure 6.1: The basic polling system

Polling system have many characteristics in common with traffic signal systems. For example, customers can be seen as vehicles. The service process of the customers can be compared with the departure process of the vehicles. The switch-over time from one queue to another, matches the clearance-time of two incompatible queues. In the next section a detailed model description of a polling system is given.

6.2 Model description

In this section a detailed model description of the polling system is given. So in this section, we speak about customers instead of vehicles. Later on in this chapter, we will use the k-limited polling system to model the fully-actuated control system. Then we will speak about vehicles instead of customers.

6.2.1 The basic model

Consider a polling model consisting of multiple ($= n$) queues, $Q_1, \ldots, Q_n$, attended by a single server S. The server visits the queues in strictly cyclic order, $Q_1, \ldots, Q_n$. The cycle time is given by the stochastic variable C_i and denotes the time between successive visits by the server to queue i. However, $E[C_i]$ is independent of i and is also independent of the service discipline, as long as all queues remain stable. We take $E[C]$ to denote this common value. Customers arriving at Q_i are also referred to as type-i customers, $i = 1, \ldots, n$. The queues are assumed to have infinite buffer capacity.

6.2.2 The arrival process

Customers are assumed to arrive according to independent Poisson processes. The first moment of the interarrival times of type-i customers is given by α_i, $i = 1, \ldots, n$. Denote by $\lambda_i := 1/\alpha_i$ the arrival rate at Q_i, $i = 1, \ldots, n$. The total arrival rate is $\lambda := \sum_{i=1}^{n} \lambda_i$.
We focus here on models with single arrivals, i.e., customers are assumed to arrive one by one, instead of arriving in platoons.

6.2.3 The service process

Type-i customers require deterministic service times $1/\mu_i$, $i = 1, \ldots, n$. Define $\rho_i := \lambda_i/\mu_i$ as the occupation rate at Q_i, $i = 1, \ldots, n$. The total occupation rate is $\rho := \sum_{i=1}^{n} \rho_i$.

6.2.4 The switch-over process

Moving from Q_i to Q_j, the server needs a deterministic switch-over time s_{ij}, $i, j = 1, \ldots, n$. Because the server visits the queues in strictly cyclic order only s_{ii+1} is important. In the polling literature, switch-over times are usually assumed to depend only on the previous queue visited, i.e., $s_{ij} = s_i$, $i, j = 1, \ldots, n$. The total mean switch-over time is given by $s := \sum_{i=1}^{n} s_i$.

6.2.5 The service policy

The service policy prescribes which number of customers server S should serve. There are three classical service disciplines, and the k-limited service belongs to one of these.

I. Exhaustive service

Under exhaustive service, the server continues to work until the queue becomes empty. Customers that arrive during the visit of the server to that queue, are served in the current visit.

II. Gated service

Under gated service, S serves only the customers that were present at the start of the visit. Customers that arrive during the visit of the server to that queue, are served in the next visit.

III. *K*-limited service

Under k-limited service, the server continues to work on queue Q_i until either a prespecified number of k_i customers have been served, or the queue becomes empty, whichever occurs first. A special case of this limited service strategy is the case when $k_i = 1, i = 1, \ldots, n$ (serve just one customer, if any). This is called *1-Limited service*.

There are two versions of limited service: *Exhaustive-limited service* and *Gated-limited service*, depending on whether or not S only serves the customers that were present at the start of the visit. Note that the case $k_i = \infty, i = 1, \ldots, n$ results in exhaustive service.

In this chapter we will focus on the third type of service, the k-limited service, and more specifically, the exhaustive limited service. The order of service is always assumed to be First Come First Served (FCFS), i.e., customers are assumed to be served in order of arrival.

6.3 Constrained optimization problem

A simple traffic balance argument shows that if the system is stable, the server is working a fraction ρ of the time. This results in the simple equation:

$$E[C] = s + \rho E[C]$$

So that the mean cycle time is given by:

$$E[C] = \frac{s}{1-\rho}$$

If the system is stable, on average the number of customers arriving at queue i during a cycle must be smaller than the maximum number of customers leaving the queue during the service. For the k-limited service strategy this can be translated into the following condition for stability [10]:

$$\frac{\lambda_i s}{1-\rho} < k_i, \qquad i = 1, \ldots, n \tag{6.1}$$

Throughout this chapter, the stability condition (6.1) is assumed to hold.

Denote by D_i the delay of an arbitrary type-i customer, $i = 1, \ldots, n$. Let w_i represent the delay cost per unit of time of a type-i customer, $i = 1, \ldots, n$. The average delay cost per unit of time is then given by $\sum_{i=1}^{n} w_i \lambda_i E[D_i]$. In this chapter we are interested in the problem of finding the values for $k_1, \ldots, k_n$, that minimize that quantity. To have a bound on the cycle time, we therefore study the more specific version of the problem with the constraint $\sum_{i=1}^{n} \gamma_i k_i \leq K$. When $\gamma_i = \beta_i$ and $K = L - s$ is chosen, then a limit L on the mean cycle time is given at periods of overload (namely when all queues are loaded and in every visit k_i customers are served). This suggests a rule for the optimal setting of time-limits in polling models with a time-limited service discipline. Note that in the case of constant service times, the k-limited and time-limited service disciplines coincide.

Borst et al. [4] have proposed four approaches for determining these k-limit values so as to minimize the mean waiting cost of customers in the polling system. The approaches have been tested for a variety of cases. All four approaches perform reasonably well. Especially one, which we discuss in more detail in the rest of this section.

6.3.1 A Fuhrmann & Wang-like k-limited approximation

Fuhrmann & Wang [12] developed an approximation for the mean waiting time $E[D_i]$. They proposed that the waiting time of a customer can be broken down into three parts. Therefore we have to introduce the following stochastic variables. Let X_i denote the number of customers waiting at queue Q_i seen by an arbitrary customer arriving at queue Q_i. And let RC_i denote the residual cycle time with regard to Q_i. Then the waiting time of a customer can be broken down into:

- The time until the server next visits queue Q_i. This time has mean length $E[RC_i]$.
- Next there are $\lfloor X_i/k_i \rfloor$ 'busy i-cycles', i.e., in which the maximum number of k_i customers are served at queue Q_i.
- Finally, there are $X_i - k_i \lfloor X_i/k_i \rfloor$ individual service times, each of mean length $1/\mu_i$.

Kuehn [17] suggested that for each of the 'busy i-cycles' denoted by $C_{b,i}$ the following balance argument holds:

$$E[C_{b,i}] \approx k_i/\mu_i + s + E[C_{b,i}] \sum_{j \neq i} \rho_j, \qquad i = 1, \ldots, n$$

So that $E[C_{b,i}]$ is approximately given by:

$$E[C_{b,i}] \approx \frac{k_i/\mu_i + s}{1 - \rho + \rho_i}, \qquad i = 1, \ldots, n \tag{6.2}$$

In the rest of this subsection two extreme cases are studied: The 1-Limited service and the Exhaustive service. For the general case, the results for the two latter cases are combined to find a good approximation for the $E[D_i]$'s.

1-Limited service case

When $k_i = 1, i = 1, \ldots, n$, summing the listed terms in the beginning of this subsection yields the following approximate formula:

$$E[D_i] \approx E[RC_i] + E[X_i]E[C_{b,i}], \qquad i = 1, \ldots, n$$

Applying Little's Law, $\lambda_i E[D_i] = E[X_i]$ gives:

$$E[D_i] \approx \frac{E[RC_i]}{1 - \lambda_i E[C_{b,i}]}, \qquad i = 1, \ldots, n$$

Using approximation (6.2) and rearranging yields:

$$E[D_i] \quad \approx \quad \frac{(1+\frac{\rho_i}{1-\rho})E[RC_i]}{1-\lambda_i E[C]}, \qquad i=1,\dots,n \tag{6.3}$$

Exhaustive service case

With the exhaustive service discipline two things can happen. A customer can arrive during the visit of the server to that queue and is served in the same visit. On the other hand, a customer can arrive, when the server is not visiting that particular queue. In that case, the customer has to wait until the server visits the queue and is then served in that visit.

The average delay of a customer in the first situation is given by:

$$E[D_i] \quad = \quad -E[C](1-\rho_i)^2 + E[RC_i](1-\rho_i), \qquad i=1,\dots,n \tag{6.4}$$

The average delay of a customer in the second situation is given by:

$$E[D_i] \quad = \quad E[C]\rho_i(1-\rho_i) + E[RC_i](1-\rho_i), \qquad i=1,\dots,n \tag{6.5}$$

By PASTA we can easily determine the probability that a customer arrives during the visit of the server to that queue. This probability is given by ρ_i (situation 1). Multiplying equations (6.4) and (6.5) with the probability these two situations will happen (ρ_i and $(1-\rho_i$ respectively) and summing over these two possible situations, leads to the following result for the exhaustive discipline:

$$E[D_i] \quad = \quad (1-\rho_i)E[RC_i], \qquad i=1,\dots,n \tag{6.6}$$

General case

Summing the terms listed earlier now yields approximately:

$$E[D_i] \quad \approx \quad E[RC_i] + E[X_i]E[C_{b,i}]/k_i, \qquad i=1,\dots,n$$

Applying Little's Law, $\lambda_i E[D_i] = E[X_i]$ and rearranging gives:

$$E[D_i] \quad \approx \quad \frac{E[RC_i]}{1-\frac{\lambda_i}{k_i}E[C_{b,i}]}, \qquad i=1,\dots,n$$

Now using approximation (6.2) we have:

$$E[D_i] \quad \approx \quad \frac{(1+\frac{\rho_i}{1-\rho})E[RC_i]}{1-\frac{\lambda_i}{k_i}E[C]}, \qquad i=1,\dots,n \tag{6.7}$$

Note that, as $k_i \to \infty$, the above approximation converges to $(1+\frac{\rho_i}{1-\rho})E[RC_i]$. But as we have seen in (6.6), it should converge to $(1-\rho_i)E[RC_i]$. Taking also into account the results for the 1-Limited service case given in (6.3), the approximation in (6.7) can be heuristically modified to:

$$E[D_i] \quad \approx \quad \frac{1-\rho_i+\frac{\rho_i}{k_i}(1+\frac{1}{1-\rho})}{1-\frac{\lambda_i}{k_i}E[C]}E[RC_i], \qquad i=1,\dots,n \tag{6.8}$$

Given the approximation in (6.8) and assuming $E[RC_i] \approx E[RC] = BE[C]$, with B some unknown constant, the following minimization problem has to be solved:

Minimize:

$$\sum_{i=1}^n w_i\lambda_i \frac{1-\rho_i+\frac{\rho_i}{k_i}(1+\frac{1}{1-\rho})}{1-\frac{\lambda_i}{k_i}E[C]}BE[C]$$

Subject to:

$$\sum_{i=1}^n \gamma_i k_i \quad \le \quad K$$
$$k_i \quad \ge \quad 0, \quad i=1,\dots,n$$

In fact, the approximation ignores the integrality of $k_1, \ldots, k_n$.

Solving this optimization problem yields the following optimal k_i^* :

$$k_i^* = \frac{\lambda_i s}{1-\rho} + (K - \sum_{j=1}^{n} \gamma_j \frac{\lambda_j s}{1-\rho}) \frac{\sqrt{w_i \lambda_i \delta_i / \gamma_i}}{\sum_{j=1}^{n} \gamma_j \sqrt{w_j \lambda_j \delta_j / \gamma_j}}, \qquad i = 1, \ldots, n \tag{6.9}$$

with $\delta_i = \rho_i(2-\rho) + \lambda_i s(1-\rho_i)$.

The various steps of the nonlinear optimization problem are worked out in Appendix B. Because we assume an $M/D/1$ system, the optimal k_i^*'s can be translated into optimal time limits u_i^* in the following way:

$$u_i^* = \frac{k_i^*}{\mu_i}, \qquad i = 1, \ldots, n \tag{6.10}$$

6.4 Fully-actuated traffic control system

We can use the k-limited service strategy, to model a fully-actuated traffic control system. In this section the k-limited service strategy will be adapted, so it can be used to find a good fully-actuated traffic control.

6.4.1 Current fully-actuated control

In this subsection it is explained how the currently used fully-actuated control system works. The set of traffic signals on an intersection is divided into m subsets G_i, with the restriction that the signals in one subset are not incompatible with each other. Within each subset, the traffic signals have right of way simultaneously. This is possible because the signals are not incompatible. For each subset, there is a signal in one of the other subsets, which is incompatible with one of the signals in that particular subset. So each subset has right of way separately from the other subsets. The subsets have right of way in a particular cyclic order $G_1, G_2, \ldots, G_m$, which is specified. That means that first subset G_1 has right of way, secondly subset G_2, and so on, until subset G_m has had right of way. Then subset G_1 has right of way again, and so on. After the effective green-time of a subset a clearance time is waited, until the traffic signals of the next subset in order will have right of way.

When the signals in subset G_i have right of way next, but no vehicles are waiting at each of the signals in that particular subset, the signals will not turn green, and the subset will be skipped. Then subset G_{i+1} is next in order.

The effective green-time of signals in a subset lasts until no vehicles are present at any of the approaches anymore. That means, when no vehicles are waiting to drive off, the signals turn red and the next subset in order will (possibly) have right of way.

The control described above, corresponds to the exhaustive control analyzed in chapter 5. To this fully-actuated control three extra requirements are added. The effective green-time of the signals within subset G_i will not exceed a specific maximum value u_i. When the signals of subset G_i turn green, they stay green during a specific minimum value l_i. These maximum and minimum effective green-time values don't have to be the same for all signals within one subset. When the maximum total cycle time ($\sum_{i=1}^{n} u_i$) is increasing, this will result in shorter average delays. However, the variance of the delays will increase, leading to very short but sometimes very long delays. To prevent this, the maximum total cycle time is bounded by a maximum value L.

Current method to determine maximum effective green-times

Now we will discuss the current method, that is used to determine the maximum effective green-times.

The minimum effective green-times are prescribed, based on the design of the intersection. First a fixed-time signal scheme is determined using *Cocon* [6]. This signal scheme is used as a basis for the fully-actuated control. Signals that have right of way simultaneously in the fixed-time control, will be placed in one subset in the fully-actuated control. When the signals are manually divided into subsets, then for each signal a maximum effective green-time is determined based on the length of the effective green-time of the fixed-time signal scheme. So not all signals within one subset have the same maximum effective green-time. This is slightly different from our model where all signals within one subset have the same maximum effective green-time. To explain why the model used at this moment is unnecessarily difficult, we look at the case, where two signals i_1 and i_2 within subset G_i have different maximum effective green-times u_{i_1} and u_{i_2}. We assume that $u_{i_1} < u_{i_2}$ when $\rho_{i_1} < \rho_{i_2}$. The only case, when $u_{i_1} < u_{i_2}$ yields better results on average in comparison with $u_{i_1} = u_{i_2}$, is when the queue at signal i_2 is empty already, but still vehicles are waiting at signal i_1. But because of $\rho_{i_1} < \rho_{i_2}$, it will rarely happen that at signal i_1 more vehicles have arrived during the effective red-time of subset G_i than at signal i_2. That's the reason, why we choose the maximum effective green-times of signals within one subset to be all equal.

The length of the maximum effective-green time is equal to the length of the effective green-time of the fixed-time control scheme. The sum of the lengths of the maximum effective green-times of the subsets, i.e. the maximum value of the maximum effective green-time of the signals within the subset, is equal to the maximum value L minus the total clearance-time (s).

6.4.2 Translation to k-limited polling

In this subsection we will model the current fully-actuated traffic control system as a k-limited polling system. Here we assume that the set of signals is already divided into subsets.

With k-limited polling we have a single server, serving the queues one at a time. The signals within one subset have right of way simultaneously, resulting in a sort of multi-server polling system. In the field of multi-server polling systems not much is known. To translate the multi-server polling system into a single server polling system, we make the assumption, that the busiest approach within a subset, that is the approach with the highest occupation rate ρ_i ($= \lambda_i/\mu_i$), represents the whole subset. In other words, all signals within one subset are replaced by the signal belonging to the approach with the highest occupation rate.

Setting $\gamma_i = 1/\mu_i$, $K = L - s$ and $w_i = 1/\lambda = 1/\sum_{i=1}^{n} \lambda_i$ reduces the constraint optimization problem to:

Minimize:

$$\sum_{i=1}^{n} \frac{\lambda_i}{\lambda} E[D_i]$$

Subject to:

$$\begin{aligned} \sum_{i=1}^{n} \frac{k_i}{\mu_i} &\leq L - s \\ k_i &\geq 0, \quad i = 1, \ldots, n \end{aligned}$$

The first constraint in the optimization problem above imposes a limit L on the mean cycle time at periods of overload (namely, when all queues are loaded). Then the optimal k_i^* and u_i^* are given by (compare with (6.9) and (6.10)):

$$\begin{aligned} k_i^* &= \frac{\lambda_i s}{1-\rho} + \mu_i \left(L - \frac{s}{1-\rho}\right) \frac{\sqrt{\rho_i \delta_i/\lambda}}{\sum_{j=1}^{n} \sqrt{\rho_j \delta_j/\lambda}} \\ u_i^* &= \frac{k_i^*}{\mu_i} \end{aligned}$$

with $\delta_i = \rho_i(2-\rho) + \lambda_i s(1-\rho_i)$.

The objective that was minimized is equal to the weighted average delay of vehicles at the intersection, where the weight of signal i, is given by λ_i/λ. Each subset of signals is represented by the busiest one. So the optimal k_i^* for the busiest signal is found and the average delay of vehicles at the busiest signal is minimized. The larger the value u_i is, the smaller the delays of the signals of that particular subset will be, but the larger the delays of the signals of the other subsets will be. Although the average delay of vehicles at the busiest signals is minimized, the average delay of vehicles at the other signals is not (necessarily) minimized. But the weight of these signals is (much) smaller than the weight of the busiest signal, resulting in small increments of the objective. That's the reason, why we have chosen that all signals within one subset have the same maximum effective green-time determined by the busiest signal within that subset.

In figure 6.2 the average delay as function of the maximum effective green-time value is plotted. When u_i goes to infinity, the k-limited polling system is equal to the exhaustive polling system. So x_i in figure 6.2 corresponds to the average delay of the exhaustive system, analyzed in chapter 5. As we have seen earlier in this chapter, the system is stable for $k_i > \lambda_i s/(1-\rho)$ and as a result $u_i > \rho_i s/(1-\rho)$. So y_i in figure 6.2 corresponds to $\rho_i s/(1-\rho)$.

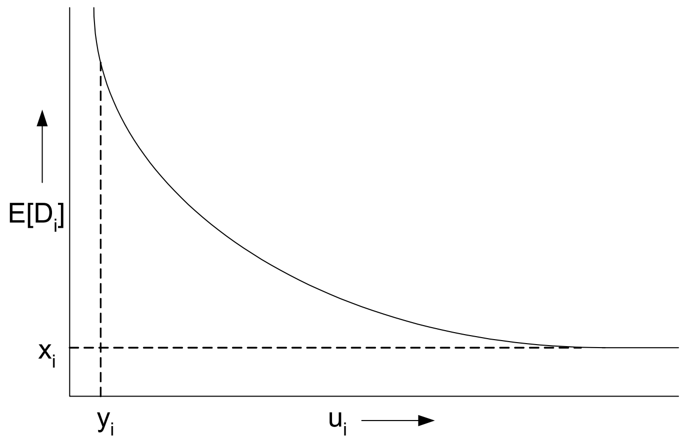


Figure 6.2: The average delay d_i as function of the maximum effective green-time u_i

6.4.3 Division of signals into subsets

In this subsection we consider the division of the traffic signals into subsets. For this purpose, a simple heuristic is developed. This heuristic is discussed below.

The set of traffic signals is arranged in order of values of ρ_i, from high to low. So the signal corresponding to the highest value of ρ_i, is the first element in this arranged set of signals. The order in which the signals are placed in subsets, is the order of the arranged set of signals. Assume we have already m subsets. We first try to place the next signal in order in the first subset. When the signal is incompatible with one of the signals in this subset, we try to place the signal in the second subset, and so on. When the signal cannot be placed in any of the existing subsets, a new subset is created and added at the end of all subsets. The signal is placed in this subset. We continue this heuristic, until each element in the arranged subset is placed in a subset.

The heuristic discussed above is summarized in the following mathematical program:

Arrange the set traffic signals in order of values of ρ_i, from the highest value to the lowest value;
This arranged set S^* is given by $\{i_{(1)}, i_{(2)}, \ldots, i_{(n)}\}$;
The number of different subsets, $m := 0$;
WHILE $n > 0$ DO
    Take the first element of S^*, $i_{(1)}$;
    Take $s = 0$;
    Bool:=True;
    WHILE $s < m$ AND Bool=True DO
        IF $i_{(1)}$ is incompatible with any of the signals in subset G_s THEN
            $s := s + 1$;
        ELSE
            Signal $i_{(1)}$ is placed in subset G_s;
            Bool:=False;

```
            ENDIF;
        ENDWHILE;
        IF Bool=True THEN
            Signal i(1) is placed in subset Gs;
            m := m + 1;
        ENDIF;
        Remove element i(1) from the arranged set S*;
        n := n − 1;
        FOR k = 1 TO n DO
            i(k) := i(k+1);
    ENDWHILE.
```

Example

Consider the simple intersection with four traffic signals as given in figure 6.3. Suppose signals 2 and 8 are not incompatible and can have right of way simultaneously. On the other hand signals 5 and 11 must have separate effective green-times. In the figure we can easily see, that signals 2 and 8 are incompatible with signal 5 and 11 and the other way around.

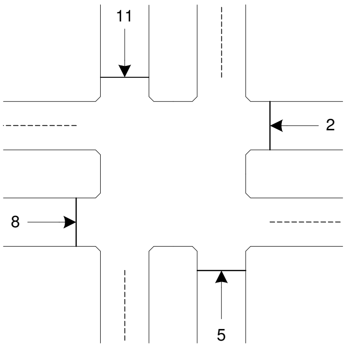


Figure 6.3: Example of a simple intersection

The traffic intensities (number of arriving vehicles per hour) and the signal capacities (maximum number of leaving vehicles per hour) are given in table 6.1.

i	λ_i	μ_i	ρ_i
2	0.1111	0.5000	0.222
5	0.3056	0.5278	0.579
8	0.1389	0.5278	0.263
11	0.0694	0.4722	0.147

Table 6.1: Input of intersection in example

Explanation of algorithm above: **Step 1**
$S^* = \{5, 8, 2, 11\}$
Step 2
$G_1 = \{5\}$
Step 3
$G_1 = \{5\}, G_2 = \{8\}$
Step 4
$G_1 = \{5\}, G_2 = \{8, 2\}$
Step 5
$G_1 = \{5\}, G_2 = \{8, 2\}, G_3 = \{11\}$

6.4.4 Optimal sequence of subsets

When the signals are divided into subsets, the optimal sequence of these subsets has to be determined. The optimal sequence of the subsets, is the sequence where the total clearance-time s is minimized. So all permutations of the n subsets are determined. The total number of (cyclic) permutations is given by $(n-1)!$. For every permutation the total clearance-time is computed. The permutation with the smallest value of this clearance-time is chosen as optimal sequence. Because the total number of subsets is always small this action will not be very time consuming.

The mathematical program that is used to determine all permutations (in a recursive way) is given in Appendix F.

In the preceding subsections we have discussed all different steps to determine a good fully-actuated traffic control. We have given a simple heuristic to divide the set of signals into subsets, where each subset has right of way separately. Given the division into subsets, we can easily determine the optimal cyclic order in which the 'server' should visit the subsets. To determine the maximum effective green-times of these subsets, we have translated the fully-actuated traffic control system to the k-limited polling system. Until now, we didn't take the minimum effective green-times l_i into consideration. In the next section we will discuss the effect of these minimum effective green-times on the method described in this section.

6.5 Introduction of minimum effective green-time

We cannot directly apply the k-limited service policy to derive values for u_i. The reason is, that if a signal turns green, it must stay green for at least l_i time-units. It may occur that no vehicles are waiting at approach i, while the length of the effective green-time didn't reach the minimal value l_i. This results in a loss of capacity. Because of the loss of capacity, the u_i's as determined in the previous section don't have to yield a stable system.

In this section we first investigate the stability of the system. Subsequently two adaptations of the k-limited policy are proposed, taking into account the existence of minimum effective green-times.

6.5.1 Stability

For the normal k-limited polling model, we have seen in (6.1) that the following stability condition holds:

$$k_i > \frac{\lambda_i s}{1-\rho}, \qquad i = 1, \ldots, n$$

which corresponds to:

$$u_i > \frac{\rho_i s}{1-\rho}, \qquad i = 1, \ldots, n$$

The stability condition of the fully-actuated control, isn't that simple. It is still an unsolved problem, which needs further attention in future research.

6.5.2 First adaptation of k-limited strategy

Here we assume, that during the minimum effective green-time no vehicles can leave. So, we can treat the minimum effective green-times l_i as clearance times. So:

$$s^* := s + \sum_{i=1}^{n} l_i$$

The advantage of this approximation, is that the new found k_i^* always results in a stable system on condition that:

$$u_i^* > \frac{\rho_i (s + \sum_{i=1}^{n} l_i)}{1-\rho} \tag{6.11}$$

Then the optimal k_i^* and u_i^* are given by:

$$\begin{aligned} k_i^* &= l_i + \frac{\lambda_i(s + \sum_{j=1}^n l_j)}{1-\rho} + \mu_i(L - \frac{s + \sum_{j=1}^n l_j}{1-\rho})\frac{\sqrt{\rho_i \delta_i/\lambda}}{\sum_{j=1}^n \sqrt{\rho_j \delta_j/\lambda}} \\ u_i^* &= \frac{k_i^*}{\mu_i} \end{aligned}$$

The advantage of this strategy lies in the fact that, when condition (6.11) is satisfied, the u_i^*'s always yield a stable system. On the other hand, by assuming that no vehicles depart during the minimum effective green-time, the u_i's of signals with a low occupation rate (ρ_i), will be too large in proportion to the u_i's of signals with a high occupation rate. The signals with a high occupation rate affect the objective (the sum of the weighted average delay of vehicles) the most. In this approximation, the signals with a high occupation rate have u_i's that are too small compared with the u_i's of signals with a low occupation rate. This will result in high values of the objective.

6.5.3 Second adaptation of k-limited strategy

To find another approximation we look back at the original k-limited polling model. The optimal u_i's are given by:

$$u_i^* = \rho_i \frac{s}{1-\rho} + (L - \frac{s}{1-\rho})\frac{\sqrt{\rho_i \delta_i/\lambda}}{\sum_{j=1}^n \sqrt{\rho_j \delta_j/\lambda}} \tag{6.12}$$

with $s/(1-\rho) = E[C]$.

One may interpret (6.12) as follows. Signal i should have an effective green-time of at least $\rho_i E[C]$, to satisfy the stability condition. The remaining non-clearance-time, $L - E[C]$, should be assigned proportional to $\sqrt{\rho_i \delta_i/\lambda}$.

When introducing a minimum effective green-time, a closed expression for the average cycle-time $E[C]$ cannot be found. However, the average cycle time can be approximated recursively, with the following equation:

$$E[C^{app}] = s + \sum_{i=1}^n max(\rho_i E[C^{app}], l_i) \tag{6.13}$$

So for the u_i's in the model with minimum effective green-times the following inequality must hold:

$$u_i > \rho_i E[C^{app}]$$

with $E[C^{app}]$ as determined in (6.13).

This stability condition is too strong, because $E[C^{app}] > E[C]$. When $E[C^{app}]$ is greater than L, we cannot use this approximation to determine the u_i^*'s. When this is not the case the optimal u_i^*'s are determined in the same way as in (6.12): Signal i should have an effective green-time of at least $\rho_i E[C^{app}]$. The remaining non-clearance-time will be assigned proportional to $\sqrt{\rho_i \delta_i/\lambda}$, in the same way as in the original Fuhrmann & Wang k-limited approximation.

$$u_i^* = \rho_i E[C^{app}] + (L - E[C^{app}])\frac{\sqrt{\rho_i \delta_i/\lambda}}{\sum_{j=1}^n \sqrt{\rho_j \delta_j/\lambda}}$$

The advantage of this strategy, is the fact that the signals have a value u_i that is proportional to $\sqrt{\rho_i \delta_i/\lambda}$, which will result in a nearly optimal objective. On the other hand, it is not clear whether for these values of u_i the system is stable, because we make use of an approximation of $E[C]$.

6.6 Discrete-event simulation

To simulate the fully-actuated traffic control, a discrete-event simulation program with an event scheduling approach has been designed. The event scheduling approach concentrates on the events and how they affect the system. The three events that can be distinguished are:

1. Arrival of a vehicle;
2. Departure of a vehicle;
3. Beginning of the effective green-time.

The signals are divided into n subsets, where the signals within each subset are turned green simultaneously. The server serves the subsets in cyclic order $(1, 2, \ldots, n)$. The effective green-time of signals in subset i lasts at least l_i seconds. When these l_i seconds are elapsed, we investigate at each departure whether there are waiting vehicles at signals in subset i. When at each of the signals in subset i no vehicles are waiting anymore, the signals are turned red. If there are still vehicles waiting at one of the signals in subset i, after an effective green-time of u_i seconds, then the signals are turned red as well. The signals of the next subset in order $i+1$ are turned green after a clearance time of $s_{i,i+1}$. During this clearance time all signals are turned red and as a result no vehicles can drive off.

We keep track of the time points at which the next events of the different types occur. This Future Event Set (FES) is implemented as a binary search tree to determine the next event efficiently. In a tree, the time points are not ordered in a straight line, like earliest event first, and so on. Instead, the starting time point, called the root, is linked to two other nodes, called its children, and those nodes in turn are linked to other children, and so on. Formally, a tree is either empty, or a root, which is connected to one or more other trees, called the subtrees of the root. The order of all time points in this tree is important. Formally, in a binary search tree the following holds:

- All the time points in the left subtree take place earlier than the time point of the root
- All the time points in the right subtree take place later than the time point of the root
- The left and right subtrees are also binary search trees

We can conclude, that the event that takes place first, is the leftmost node in the tree. We use a binary search tree in order to minimize the distance we have to go to reach any given element. Searching for an element in a binary tree containing n nodes is an $O(\log n)$ process and building the tree in the first place is an $O(n \log n)$ process, if the tree is reasonably well balanced.

The simulation then consists of finding the smallest time point in this tree, setting the current time to this event time and executing the corresponding activities. Below it is described how these events affect the system and what activities are carried out.

1. Arrival of a vehicle

With an arrival, the vehicle is placed in the waiting queue. For each vehicle in the queue we keep track of the position of the vehicle in the queue as well as the point of time the vehicle joined the queue. When a vehicle arrives, a new arrival at the same signal is simulated. When no vehicles were waiting at the moment the vehicle arrived at the signal, a departure is simulated.

2. Departure of a vehicle

With the departure of a vehicle, we first compute the delay of that particular vehicle. Because we keep track of the points of time vehicles arrive at the queue this can easily be done. All (possible) remaining vehicles shift one place forward in the queue. If there are still vehicles waiting, the signal is not turned red, but a new departure is simulated. When the minimum effective green-time of l_i seconds is passed by, we investigate whether all queues within the subset are empty. When this is true, a new event of type 3 (the beginning of the effective green-time) is simulated after the clearance time between subset i and $i+1$. Furthermore, the state of the signals in subset i is changed from green to red.

3. Beginning of the effective green-time

At the beginning of the effective green-time of subset i, the state of the signals within this subset changes from red to green. When there are waiting vehicles at the queues a new departure is generated and added to the binary search tree.

As stopping criterium for this simulation the runlength is taken. When the current simulation time exceeds this runlength the simulation is ended.

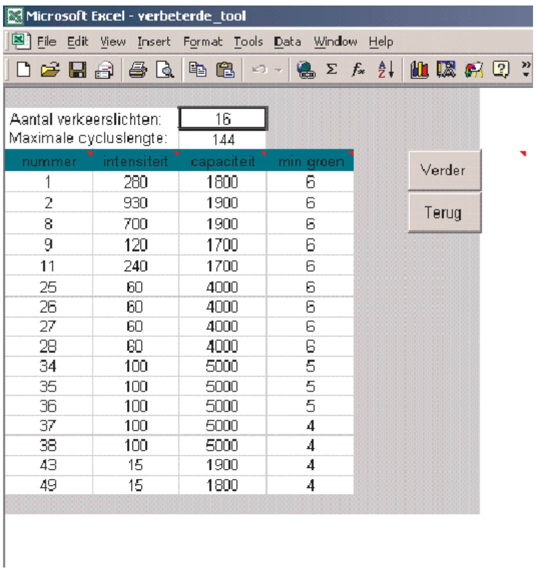
6.7 Excel tool

The heuristics and approximations described in this chapter are implemented in an Excel tool. In this section this tool is described.

6.7.1 Input

To start the tool, first the total number of traffic signals and the upper limit L on the total cycle time has to be given. Subsequently the rest of the input can be given on the input-sheet. An example of the input-sheet can be seen in figure 6.4. As can be seen in this figure, the following input has to be given for each signal:

- i: The number of the traffic signal, according to table 1.1.
- λ_i: The total number of arriving vehicles per second.
- μ_i: The maximum number of leaving vehicles per second.
- l_i: The minimum length of the effective green time in seconds.



nummer	intensiteit	capaciteit	min groen
1	280	1800	6
2	930	1900	6
8	700	1900	6
9	120	1700	6
11	240	1700	6
25	60	4000	6
26	60	4000	6
27	60	4000	6
28	60	4000	6
34	100	5000	5
35	100	5000	5
36	100	5000	5
37	100	5000	4
38	100	5000	4
43	15	1900	4
49	15	1800	4

Figure 6.4: Print of Excel input screen

6.7.2 Output

After the essential input is given, the maximum length of the effective green-times is determined. First the signals are divided into subsets according to the heuristic described in subsection 6.4.3. Secondly the optimal sequence of these subsets is determined according to the method described in subsections 6.4.4. To give an approximation of the optimal u_i's, we have implemented approximation 1 of the k-limited polling strategy (described in subsection 6.5.2) in this Excel tool. The reason of this is, that by using this approximation a stable system can be guaranteed when certain stability conditions are satisfied. Maybe these conditions are a little bit too strong, but we prefer this to approximation 2, where no stability conditions can be given.

The procedures and heuristics described in this subsection are schematically presented in the following mathematical program:

Give input;
Determine division of subsets;
Determine optimal sequence of subsets;
Determine u_i's;
Give output.

In figure 6.5 the output screen of this Excel tool can be seen.

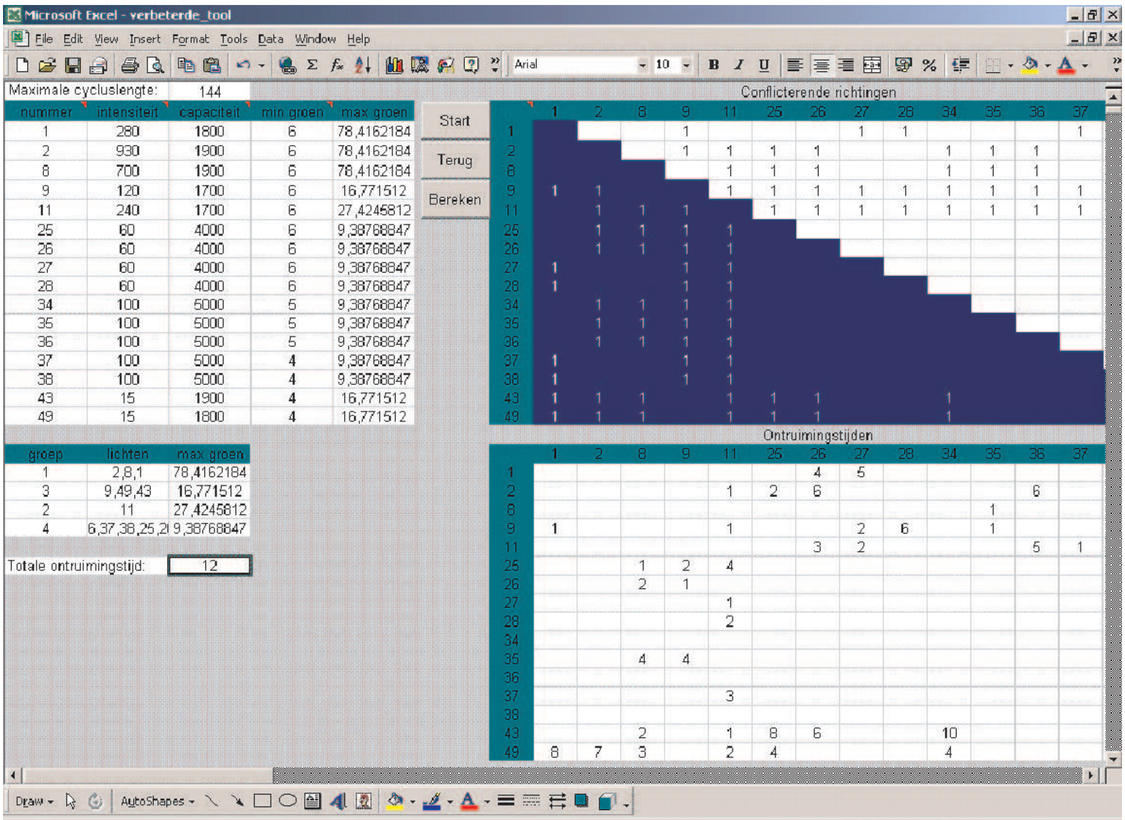


Figure 6.5: Print of Excel output screen

6.8 Case

In this section we will test the developed method on an intersection in Eindhoven. This intersection is part of a set of three new intersections in the western part of Eindhoven. We will investigate the fully-actuated control during the evening rush-hour.

6.8.1 Schematic overview

The investigated intersection is schematically presented in figure 6.6.

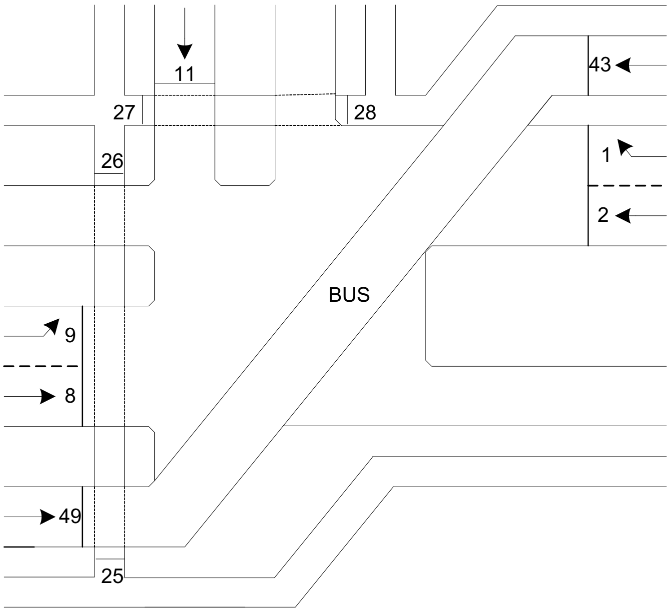


Figure 6.6: Overview of intersection (Scale 1:300)

As can be seen in this figure, there are five car signals (numbered 1, 2, 8, 9, 11). Furthermore, there are eleven other signals for cyclists (numbered 25, 26, 27, 28), pedestrians (numbered 34, 35, 36, 37, 38) and buses (numbered 43, 49). For simplicity the pedestrian signals are not shown in figure 6.6. In Appendix D can be found, which signals are incompatible with each other. In the same Appendix the clearance-times of these incompatible signals are given.

6.8.2 Results

The results generated by the developed heuristic (with different upper-limits on the total cycle time) are compared with each other by simulation. First, the results of the current situation are given. Secondly, the current division of the signals into subsets is maintained, but different u_i's are determined. Finally for upper-limits L is 144, 120 and 90 seconds, the u_i's are determined and compared with previous results. As objective, we have taken the weighted average delay of the car and bicycle signals. The average delays of pedestrian and bus signals are not included in this function, because no u_i's have to be determined for these groups.

For each signal, the number i, arrival rate λ_i, departure rate μ_i and minimum effective green-time l_i is given. The resulting u_i's are given, as well as the average length of the effective green-time ($E[T_i]$) and the average delay ($E[D_i]$) with a 95%-confidence interval. Furthermore, the fraction of the vehicles that has to wait more than 90 seconds ($= f_{E[D_i]>90}$) is given in the last column of the table. Finally, on top of each table the upper-limit L and the total clearance-time s are presented. The variables l_i, u_i, $E[T]_i$, $E[D_i]$ and the objective are expressed in seconds.

Current division & current u_i's

In table 6.2 the results of the current fully-actuated control are presented. Here we can see, that the difference between the average effective green-time of signals 1 and 11 (= 20.47 seconds) and the maximum effective green-time of these signals (= 25.0 seconds) is small. This results in reasonably high delays.

The value of the average effective green-time of signals 25 and 26 is close to the minimum effective green-time. So we would expect that the average delays are small. This isn't true, because the average effective red-time is large. Therefore vehicles have to wait a long time before the signals turn green.

L=144 s=19		Sequence: (2,8,27,28,37,38)-(9,43,49)-(25,26,34,35,35)-(1,11)					
i	λ_i	μ_i	l_i	u_i	$E[T_i]$	$E[D_i]$	$f_{E[D_i]>90}$
1	0.0778	0.5000	6.0	25.0	20.47	60.88($\pm$ 0.170)	0.218
2	0.2583	0.5278	6.0	70.0	56.76	47.42($\pm$ 0.161)	0.115
8	0.1944	0.5278	6.0	70.0	56.76	26.18($\pm$ 0.024)	0.003
9	0.0333	0.4722	6.0	15.0	9.37	62.96($\pm$ 0.102)	0.243
11	0.0666	0.4722	6.0	25.0	20.47	55.72($\pm$ 0.044)	0.180
25	0.0167	1.1111	6.0	15.0	6.12	54.08($\pm$ 0.042)	0.188
26	0.0167	1.1111	6.0	15.0	6.12	54.37($\pm$ 0.027)	0.188
27	0.0167	1.1111	6.0	70.0	56.76	14.77($\pm$ 0.022)	0.000
28	0.0167	1.1111	6.0	70.0	56.76	14.60($\pm$ 0.015)	0.000
Objective: 43.30							

Table 6.2: Current division & current u_i's with upper-limit $L = 144$

Current division & new u_i's

In table 6.3 the results (with the same division of the signals into subsets, but the u_i's according to the new method) are presented. In this table we can see, that u_i's don't differ that much from the original values. The u_i's for the signals 25 and 26 are decreased significantly and this results in small increments of the u_i's of the other signals. In the current situation L is chosen to be 144 seconds. This value is so high, that it rarely will happen that the maximum of u_i is needed to empty queues of subset i. Therefore the results of the current situation, with objective value 43.30 seconds, compared to the situation with the same division, but newly determined u_i's, with objective value 41.55 seconds, are not that spectacular. This decrement of the objective value is mostly caused by the fact that the delay of the busiest signal 2 is decreased.

L=144 s=19		Sequence: (2,8,27,28,37,38)-(9,43,49)-(25,26,34,35,35)-(1,11)					
i	λ_i	μ_i	l_i	u_i	$E[T_i]$	$E[D_i]$	$f_{E[D_i]>90}$
1	0.0778	0.5000	6.0	27.1	20.85	58.41($\pm$ 0.114)	0.203
2	0.2583	0.5278	6.0	74.6	57.59	43.57($\pm$ 0.130)	0.081
8	0.1944	0.5278	6.0	74.6	57.59	26.04($\pm$ 0.027)	0.002
9	0.0333	0.4722	6.0	15.5	9.47	63.81($\pm$ 0.071)	0.249
11	0.0667	0.4722	6.0	27.1	20.85	54.38($\pm$ 0.043)	0.173
25	0.0167	1.1111	6.0	7.8	6.11	55.79($\pm$ 0.029)	0.200
26	0.0167	1.1111	6.0	7.8	6.11	55.78($\pm$ 0.062)	0.201
27	0.0167	1.1111	6.0	74.6	57.59	14.84($\pm$ 0.028)	0.000
28	0.0167	1.1111	6.0	74.6	57.59	14.85($\pm$ 0.028)	0.000
Objective: 41.55							

Table 6.3: Current division & new u_i's with upper-limit $L = 144$

New division & new u_i's

In table 6.4 the results of the control (with the new division of signals according to the newly developed heuristic and the u_i's according to the new method) are presented, with L still equal to 144 seconds. With the new division of the signals into subsets we have a total clearance-time s of 12 seconds, compared with the current situation where s is 19 seconds. Therefore we can increase the u_i's without increasing the value L, leading to better results. This new division has decreased the objective dramatically to 25.44 seconds.

A high value of L leads to high values of u_i and this will lead to a robust system, meaning that when the traffic intensities vary, the objective value will not vary that much. The smaller the L, the more important the way of determining the u_i's will be. That's why we have investigated the fully-actuated control with smaller values of L as well.

L=144 s=12		Sequence: (1,2,8)-(9,43,49)-(11)-(25,26,27,28,34,35,36,37,38)					
i	λ_i	μ_i	l_i	u_i	$E[T_i]$	$E[D_i]$	$f_{E[D_i]>90}$
1	0.0778	0.5000	6.0	78.9	42.15	13.31(± 0.012)	0.000
2	0.2583	0.5278	6.0	78.9	42.15	24.04(± 0.017)	0.005
8	0.1944	0.5278	6.0	78.9	42.15	17.90(± 0.007)	0.000
9	0.0333	0.4722	6.0	16.9	8.21	44.40(± 0.037)	0.090
11	0.0667	0.4722	6.0	27.7	12.37	42.59(± 0.042)	0.073
25	0.0167	1.1111	6.0	8.5	6.13	40.38(± 0.021)	0.077
26	0.0167	1.1111	6.0	8.5	6.13	40.47(± 0.038)	0.077
27	0.0167	1.1111	6.0	8.5	6.13	40.38(± 0.033)	0.077
28	0.0167	1.1111	6.0	8.5	6.13	40.26(± 0.020)	0.077
Objective: 25.44							

Table 6.4: New division & new u_i's with upper-limit $L = 144$

New division & new u_i's

In table 6.5 the results of the fully-actuated control (with the new division of signals and the u_i's according to the new method) are presented. The upper-limit L on the total cycle time is 120 seconds. The results in this table compared with the results in the previous table don't differ that much (compare the objective of 25.44 seconds with the objective 26.43 seconds). The average delay of vehicles at the signals 25, 26, 27 and 28 is decreased a little bit, because the effective red-time is decreased.

L=120 s=12		Sequence: (1,2,8)-(9,43,49)-(11)-(25,26,27,28,34,35,36,37,38)					
i	λ_i	μ_i	l_i	u_i	$E[T_i]$	$E[D_i]$	$f_{E[D_i]>90}$
1	0.0778	0.5000	6.0	63.8	41.40	13.23(± 0.008)	0.000
2	0.2583	0.5278	6.0	63.8	41.40	26.62(± 0.055)	0.014
8	0.1944	0.5278	6.0	63.8	41.40	18.19(± 0.012)	0.000
9	0.0333	0.4722	6.0	14.2	8.09	44.32(± 0.035)	0.078
11	0.0667	0.4722	6.0	22.3	12.14	43.80(± 0.033)	0.071
25	0.0167	1.1111	6.0	7.6	6.12	38.78(± 0.058)	0.050
26	0.0167	1.1111	6.0	7.6	6.12	38.75(± 0.036)	0.050
27	0.0167	1.1111	6.0	7.6	6.12	38.77(± 0.028)	0.050
28	0.0167	1.1111	6.0	7.6	6.12	38.80(± 0.056)	0.050
Objective: 26.43							

Table 6.5: New division & new u_i's with upper-limit $L = 120$

New division & new u_i's

In table 6.6 the results of the fully-actuated control (with the division of signals according to the newly developed heuristic and the u_i's according to the new approximation algorithm) are presented. The upper-limit on the total cycle time is given by L is 90 seconds. In this table we can see, that the decrement of the value L to 90 seconds leads to higher values of the delay of signals 8 and 11, but especially signal 2. The delays of the other signals didn't change dramatically. Because the traffic intensity of signal 2 is high, the value of the objective did increase to 33.04 seconds.

The smaller the value of L the larger the value of the objective will be. When we choose $L = \infty$, no constraints on the maximum effective green-time are given. This results in an exhaustive system, where the objective value is the smallest. On the other hand, the larger the value of L, the larger the variance of the delays will be. This is not advisable, because sometimes vehicles have to wait a very long time before they can move on. So the issue is to choose the value of L, in such a way that the objective value as well as the variance of the delays is acceptable.

L=90 s=12		Sequence: (1,2,8)-(9,43,49)-(11)-(25,26,27,28,34,35,36,37,38)					
i	λ_i	μ_i	l_i	u_i	$E[T_i]$	$E[D_i]$	$f_{E[D_i]>90}$
1	0.0778	0.5000	6.0	44.8	39.09	13.29($\pm$ 0.007)	0.000
2	0.2583	0.5278	6.0	44.8	39.09	42.93($\pm$ 0.167)	0.109
8	0.1944	0.5278	6.0	44.8	39.09	19.88($\pm$ 0.021)	0.002
9	0.0333	0.4722	6.0	10.8	7.78	43.66($\pm$ 0.049)	0.059
11	0.0667	0.4722	6.0	15.8	11.52	48.69($\pm$ 0.125)	0.111
25	0.0167	1.1111	6.0	6.6	6.10	35.16($\pm$ 0.027)	0.004
26	0.0167	1.1111	6.0	6.6	6.10	35.26($\pm$ 0.036)	0.004
27	0.0167	1.1111	6.0	6.6	6.10	35.24($\pm$ 0.014)	0.004
28	0.0167	1.1111	6.0	6.6	6.10	35.16($\pm$ 0.027)	0.004
Objective: 33.04							

Table 6.6: New division & new u_i's with upper-limit $L = 90$

In the last three tables in this section we could see, that the fraction of the vehicles that have to wait more than 90 seconds is decreasing when L decreases and the u_i for that particular signal is more than large enough to handle the amount of traffic. But when the u_i is just large enough, the average delay is increasing, as well as the fraction of the vehicles that have to wait more than 90 seconds.

The upper-limit L can be decreased to a certain minimum level, i.e. a value of L for which the system is just stable. By trial and error, we have determined that in this case the value is approximately 75 seconds.

6.9 Conclusions

From the simulation results in the previous section we conclude, that the newly found fully-actuated traffic control (with the same division of the signals into subsets, but the u_i's according to the new approximation algorithm) will lead to an improvement of the objective function, compared to the current situation. Moreover, when the signals are divided into subsets in a different way, the objective improves enormously compared to both other situations. From this we can conclude that the current way to divide the signals into subsets is not always effective.

Furthermore we can conclude that the smaller the value L is, the larger the objective will be, but the smaller the frequency of peaks (vehicles that have to wait a very long time) is. Of course, the frequency of peaks will only decrease when the average delay is not too large. The value of L cannot be decreased endlessly. The upper-limit can be decreased to a certain minimum level for which the system is just stable.

The biggest improvement of the approximation developed in this chapter is given by the fact that the old method to determine the fully-actuated control settings is very time-consuming compared to the new method. Furthermore, we have seen that when the upper-limit L is chosen big enough the system will be very robust, i.e. the objective value is not very sensitive to (modest) changes in the parameters u_i.

Because the computation time for the derived method is very small, the method can also be used for an adaptive control. In normal fully-actuated control, the duration of the maximum effective green-times are fixed and have to be specified by the traffic engineer. As traffic situations and conditions change, these timings should be changed as well. This is done not very frequently and certainly not systematically. With adaptive control, methods are derived to automatically adjust these maximum effective green-times. In future research, the effect of adaptive control (based on the method derived in this chapter) on the objective can be investigated by using simulation.

CHAPTER 7

PHASING SIGNALS

Arterial systems occur often in big cities where the traffic is led around the city center to prevent having too many cars in the center. This is also the case in Eindhoven, where the arterial consists of twenty-four intersections. Traffic signals force vehicles to stop and to remain stopped for a certain time, and then releases vehicles in platoons. The delays and speed changes caused by traffic signals considerably reduce the capacity of an urban arterial and lower the quality of traffic flow. To limit the discomfort of traffic signals, the signals can be phased. A large number of the intersections of the arterial in Eindhoven is phased, in such a way that the signals of the main stream turn green just before the platoon of vehicles arrives at the intersection. This chapter presents a new solution method to find a fixed-time control of phased intersections on an arterial.

The outline of this chapter is as follows: In section 7.1 the problem of determining a fixed-time control for intersections that are part of an arterial is described. The current method of phasing intersections and the newly developed method are discussed in section 7.2 and 7.3 respectively. In section 7.4 a simplified problem is translated into a Mixed Integer Program. In the next chapter we will apply the method described in this chapter to an intersection in the city of Eindhoven.

7.1 Problem description

This section introduces the problem of finding a method to control the signals of phased intersections. Consider an example of two intersections, as given in figure 7.1, where the adjacent signalized intersections 1 and 2 are connected by a two-way link. The traffic signals on both intersections have a fixed-time cyclic control, that means that the effective green-times and the order in which approaches turn green are fixed.

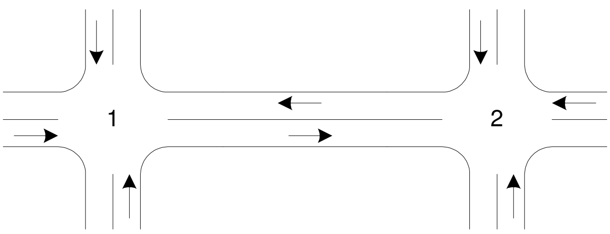


Figure 7.1: Example of a simple arterial

We speak about (partly) phased intersections, but in fact not the intersections are phased, but the signals on these intersections. We say that signal i is phased with signal j when:

- The platoon departing from signal i is arriving at signal j within the effective green-time of this signal.
- Between signal i and j vehicles don't pass another signal.

Indeed, vehicles moving from one intersection to the next intersection and the other way around, are moving in platoons. We define the length of the platoon as the distance between the front and the back vehicle. This length depends on two different things:

- The acceleration of the vehicles, when leaving the queue.
- The number of vehicles leaving the platoon because they turn left or right.

In figure 7.2 the connection between the arriving flow, the queue length and the departure flow is graphically represented.

For the intersections of the arterial the same model is used as described in section 2.1 and 2.2. The main objective is to find a fixed-time control in such a way that the sum over all intersections of the weighted average delay of vehicles on the intersections is minimized.

Remark 1

For the cycle time of all intersections the following must hold: Either the cycle time of intersection k is a multiple of the cycle time of intersection l, or the cycle time of intersection l is a multiple of the cycle time of intersection k. Most of the time, the cycle times of all intersections are equal. We define the master cycle time as the cycle time used for all phased intersections.

Remark 2

It will rarely happen that the arterial is phased in two directions.

7.2 Current way of phasing signals

Since the early seventies researchers have investigated how signals of adjacent intersections should be phased in order to optimize the traffic flow of these intersections. The method developed then, is nowadays still being used.

For the 'busiest' intersection a fixed-time control is developed. We define 'busiest' as, the intersection with the largest minimal cycle time. The minimal cycle time is defined as the smallest cycle time, for which the system is stable. For this intersection the cycle time and sequence and durations of effective green-times are determined. Then the master cycle is taken equal to this cycle. The duration of the effective green-time of the phased signal, the so-called phase green-time, can be directly derived from this scheme.

For the other intersections of the arterial the durations of the signals that have to be phased is then fixed. The duration of the effective green-time depends on the duration of the effective green-time of the busiest intersection and dispersion of the platoon. Moving along the intersections, the length of the effective green-time of phased signals will increase. The exact beginning of these effective green-times depends on the travel time from one intersection to the other. The signal turns green, just before the platoon arrives, in order to let the vehicles that were already waiting drive-off just before the platoon.

When the duration of the phased signal is determined, for the other signals a fixed-time control is determined by minimizing the total average delay of vehicles on that particular intersection.

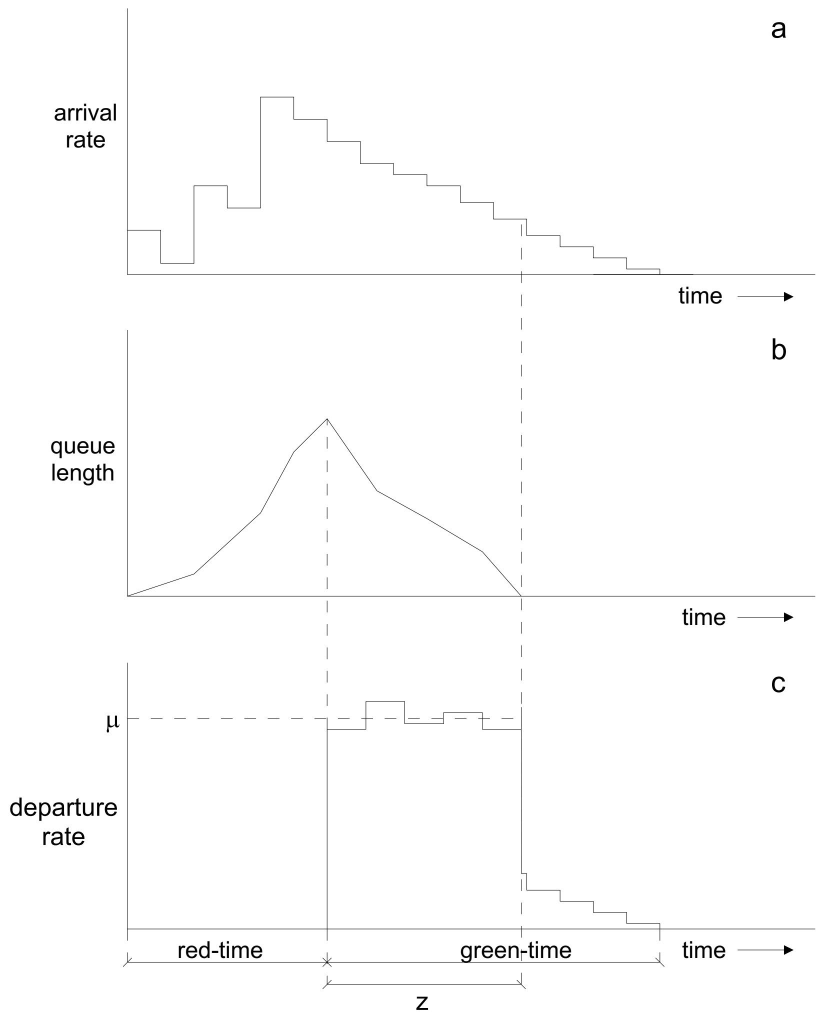


Figure 7.2: Figure 7.2a indicates the shape of the platoon arriving at an intersection; in figure 7.2b the queuelength is given as a function of the time; in figure 7.2c the platoon leaving the intersection is formed by a first part with average height equal to μ until the time at which the queue vanishes (after time z), then the platoon shape coincides with the arrival pattern.

7.3 New way of phasing signals

In this chapter we will develop an alternative method to phase the traffic signals of different intersections. In the current method, the cycle time and the duration of the effective green-time of the phased signals are fixed, when the fixed-time control for the busiest signal is determined. We develop an integral method, where the master cycle time and the length of the effective green-times of the phased signals are determined, based on all intersections instead of the busiest one.

Let $K = \{k_1, k_2, \ldots, k_m\}$ be the set of intersections which have to be phased. So k_1 is the first intersection along the road, etcetera. For the first intersection, given the cycle time c and phase green-time p the optimal sequence and durations of the effective green-times are determined, where the effective green-time of the phased signal is equal to p. This is done by minimizing the total average delay of vehicles on that particular intersection.

Vehicles move from the first intersection to the second intersection and arrive in a platoon at the phased signal. Just before the platoon arrives the signal turns green, and the vehicles already waiting at the signal drive off, in order to let the platoon continue without stopping. So we assume that the average delay of the platoon is zero.

For this intersection the cycle time c and the phase green-time are known. The duration and the sequence of the rest of the effective green-times can be determined by minimizing the weighted average delay of vehicles at the other signals.

This process continues until for all intersections a signal scheme is determined. These schemes depend on the cycle time c and the phase green-time p. We choose that cycle time c and phase green-time c, so that the sum over all intersections of the weighted average delay is minimized.

7.4 The initial model formulation

In this section the description of the problem in section 7.3 is translated into an optimization model. First, an informal verbal presentation of the model is provided, followed by the extensive notation needed to describe all aspects of the model. The objective function and each constraint is developed separately.

The model for optimizing a single intersection is thoroughly described in chapter 2. Therefore, we will only discuss the new constraints and sum up the other constraints in the mathematical model in subsection 7.4.3. For an explanation of the old constraints we refer to subsection 2.2.3.

7.4.1 Verbal model statement

Minimize: *Sum over all intersections of the weighted average delay of vehicles on the intersection.*
Subject to:

1. *For all traffic signals and all intersections: Length of the effective green-time must be smaller than or equal to the maximum green-time and must be greater than or equal to the minimum green-time,*

2. *For all traffic signals and all intersections: The beginning and the end of the effective green-times must fall in the range* $[0, Cycle)$,

3. *For all traffic signals and all intersections: The delay of vehicles as a function of the length of the effective green-time is given by the formula of Webster,*

4. *For all traffic signals and all intersections: The green-period cannot overlap the green-periods of incompatible signals and incompatible approaches have to take into account a clearance-time for safety,*

5. *For all traffic signals and all intersections: The length of the effective green-time must be large enough to handle all the arriving vehicles at that signal,*

6. *For the phased traffic signal on the first intersection: The length of the effective green-time is equal to the phase green-time and the beginning of the effective green-time is equal to zero,*

7. *For the phased traffic signal on all intersections (except the first intersection): The length of the effective green-time is equal to the length of the effective green-time of the previous intersection, plus the time needed to let all vehicles that were already waiting at the signal drive off,*

8. *For the phased traffic signal on all intersections (except the first intersection): The beginning of the effective green-time is equal to the beginning of the effective green-time of the previous intersection, plus the time needed to travel from the previous intersection to this intersection, minus the time needed to let all vehicles that were already waiting at the signal drive off.*

7.4.2 Notation

The verbal model statement of the problem can be specified as a mathematic model using the following notation.

Indices:

K	set of intersections, index k (second index l)
I_k	set of traffic signals on intersection k, index i (second index j)
Q_k	set consisting of the phased signal on intersection k, index q

Parameters:

m	number of intersections of the arterial
n_k	number of signals on intersection k
c	cycle time (integer)
p	phase green-time (integer)
λ_{ik}	arrival rate of vehicles at traffic signal i on intersection k
μ_{ik}	departure rate of vehicles at traffic signal i on intersection k
ρ_{ik}	occupation rate of traffic signal i on intersection k ($\rho_{ik} = \lambda_{ik}/\mu_{ik}$)
w_{ik}	weight of traffic signal i on intersection k ($w_{ik} = \lambda_{ik}/\sum_j^{n_k} \lambda_{jk}$)
mg_{ik}	minimum effective green-time of signal i on intersection k
Mg_{ik}	maximum effective green-time of signal i on intersection k
s_{ijk}	necessary time between the end of the effective green-time of signal i and the beginning of signal j on intersection k
r_{kl}	time a vehicles needs, to drive from intersection k to l (in seconds)

Variables:

b_{ik}	time when signal i turns green on intersection k (range $[0, c)$)
e_{ik}	time when signal i turns red on intersection k (range $[0, c)$)
g_{ik}	length of the effective green-time of signal i on intersection k ($g_{ik} = (e_{ik} - b_{ik}) \bmod c$)
d_{ik}	average delay of vehicles at traffic signal i on intersection k
D_k	weighted average delay of vehicles on intersection k ($D_k = \sum_i^{n_k} w_{ik} d_{ik}$)
z_{ijk}	1 if $b_{ik} - e_{jk} < 0$
	0 otherwise

7.4.3 Mathematical model

The constraints as formulated in the first section of this chapter can be translated into mathematical constraints, using the notation above. The constraints formulated in this subsection have the same numbers as the constraints in the earlier described verbal model statement. By determining an optimal signal scheme for one intersection, we made use of the optimization problem as formulated in chapter 2.

The constraints in the model hold for all i, j and k.

Constraint 1 & 2 & 3 & 4 & 5

$$\begin{array}{rcll} b_{jk} - e_{ik} & \leq & cz_{ijk} & \\ e_{ik} - b_{jk} & < & c(1 - z_{ijk}) & \\ g_{ik} & \leq & mg_{ik} & \\ g_{ik} & \geq & Mg_{ik} & \\ d_{ik} & \geq & \alpha_{ikh} g_{ik} + \beta_{ikh}, & h = 1, ..., N_{ik} \\ z_{ijk} + z_{jik} - z_{iik} - z_{jjk} & = & 1 & \\ b_{jk} - e_{ik} + cz_{ijk} & \geq & s_{ijk} & \\ \mu_{ik} g_{ik} & > & \lambda_{ik} c & \\ b_{ik} & < & c & \\ e_{ik} & < & c & \\ g_{ik} & < & c & \\ b_{ik} & \geq & 0 & \\ e_{ik} & \geq & 0 & \\ g_{ik} & \geq & 0 & \\ d_{ik} & \geq & 0 & \\ z_{ijk} & \in & \{0, 1\} & \end{array}$$

Constraint 6 & 7

We require that the length of the effective green-time of the phased signal is equal to the length of the effective green-time of the phased signals of the previous intersection, plus the time needed to let all vehicles that were already waiting at the signal drive off. For signals that are phased the following constraints for the effective green-time must hold:

$$\begin{array}{rcll} e_{qk} & = & p, & k = 1 \\ e_{qk} & = & e_{qk-1} + \frac{a_{qk-1}}{\mu_{qk}}, & k = 2, \ldots, m \end{array}$$

where a_{qk-1} is the amount of traffic already waiting at the phased signal before the platoon of vehicles of the previous intersection arrives. So a_{qk-1} is the amount of traffic coming from intersection $k-1$ during one cycle (the phased signal excluded) and is defined as:

$$a_{qk-1} = \sum_t \lambda_{tk-1} c, \qquad k = 2, \ldots, m$$

where index t contains all signals of intersection $k-1$, where vehicles move from signal t of intersection $k-1$ to signal q of intersection k.

Constraint 6 & 8

The beginning of the effective green-time of the phased signal is equal to the beginning of the effective green-time of the phased signal of the previous intersection, plus the time needed to travel from the previous intersection to this intersection, minus the time needed to let all vehicles that were already waiting at the signal drive off.

$$\begin{array}{rcll} b_{qk} & = & 0, & k = 1 \\ b_{qk} & = & b_{qk-1} + r_{k-1k} - \frac{a_{qk-1}}{\mu_{qk}}, & k = 2, \ldots, m \end{array}$$

Objective

The objective function, being the sum over all intersections of the weighted average delay of all vehicles on the intersection, has to be minimized

Minimize:

$$\sum_{k=1}^{m} D_k = \sum_{k=1}^{m} \sum_{i=1}^{n_k} w_{ik} d_{ik}$$

The program described above (MainProgram) is called a Mixed Integer Program, because some of the variables used to formulate the problem are continuous and some of them are integer (binary). The MainProgram can be solved with AIMMS, a linear and mixed-integer program solver.

The proposed algorithm relies on formal mathematical techniques rather than heuristics. Under the assumptions, it is expected that the optimal solution found by the algorithm is globally optimal. Since the algorithm is mathematically based, it may be shown that the solution produced is optimal for the conditions and within the constraints imposed. The solution can thus be used with confidence that there is no better alternative under the given circumstances.

CHAPTER 8

PHASING SIGNALS: EXAMPLE

The two intersections that are investigated in this chapter are situated very close to each other in the center of the city of Eindhoven. Because the mutual distance is only 80 meters, it will often occur that the roads between both intersections are congested and that this will affect the traffic flow. These problems may be solved by phasing the intersections. In this chapter we will investigate how to phase the two intersections during the rush hour in the afternoon from 4.45 PM until 5.45 PM.

The outline of this chapter is as follows: First, in section 8.1 the input, as well as some background information of the intersection is given. The method described in chapter 7 is implemented with the optimization software package AIMMS. The current signal scheme and the newly found signal scheme are presented in section 8.2 and 8.3 respectively. In section 8.4 the structure of the simulation program is discussed. The newly found signal scheme is compared with the current scheme by simulation in section 8.5. In the last section, these results are discussed.

8.1 Introduction

In this section all the necessary input and information of the investigated intersections is given.

8.1.1 Notation

The following notation is used in the rest of this chapter.

K = Set of intersections (index k)
I_k = Set of traffic signals on intersection k (index i)
c = Cycle time
λ_{ik} = Arrival rate of vehicles at traffic signal i on intersection k
μ_{ik} = Departure rate of vehicles at traffic signal i on intersection k
b_{ik} = Time when signal i turns green on intersection k (range $[0, c)$)
e_{ik} = Time when signal i turns red on intersection k (range $[0, c)$)
g_{ik} = Length of the effective green-time of signal i on intersection k
w_{ik} = Weight of signal i (used in the objective) on intersection k

d_{ik} = The average delay of vehicles at signal i on intersection k
ρ^*_{ik} = Degree of saturation of signal i on intersection k
r^*_{kl} = The driving time from intersection k to l

8.1.2 Input and background information

The investigated intersections are schematically presented in figure 8.1. The driving time $r_{12} = r_{21}$ between

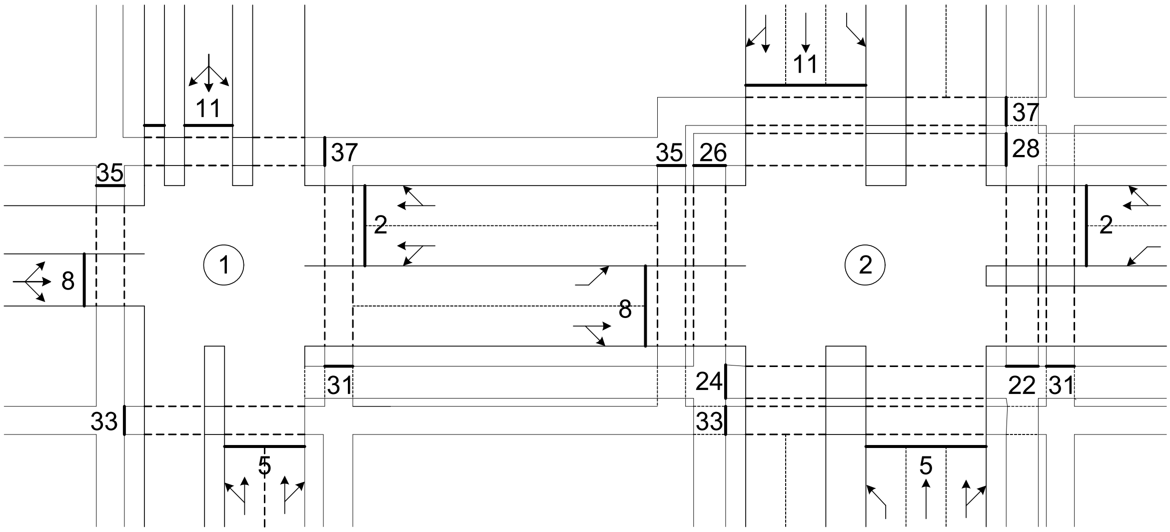


Figure 8.1: Overview of intersection (Scale 1:1000)

both intersections is 10 seconds.

As can be seen in this figure, there are four car signals (numbered 2, 5, 8, 11) for each intersection. For intersection 1, there are four other signals for pedestrians (numbered 31,33,35,37). The cyclists use the vehicle signals. For intersection 2, there are eight other signals for cyclists (numbered 22,24,26,28) and pedestrians (numbered 31,33,35,37). For all pedestrian and cyclists signals of the intersections, the minimum effective green-time is equal to 12.0 seconds.

In 1994 the number of arriving vehicles during one hour was counted at all signals. We have increased these numbers to give a more realistic view of the current intensities. The arrival rate λ_{ik} of signal i on intersection k, is the traffic intensity given in number of arriving cars per second and can be directly calculated from this counting. The drive off capacity, given in the maximum number of cars leaving per hour, of vehicles for the different signals can be easily computed, based on the design of the approaches, resulting in the total amount of vehicles leaving the approach per second (departure rate μ_{ik}). The objective that is minimized is the sum over the intersections of the weighted average delay of vehicles on the intersections. The weight w_{ik} for signal i on intersection k is given by:

$$w_{ik} = \frac{\lambda_{ik}}{\sum_j^{n_k} \lambda_{jk}}$$

In table 8.1 the input of both intersections is given. In Appendix E.1 the arrival intensities at the different signals, spread out over the different approaches are graphically presented.

As the computation time of the AIMMS program increases exponentially with the number of traffic signals, we try to minimize this number. Intersection 1 can be simplified by taking one cyclist/pedestrian signal instead of two signals. For example, signals 22 and 31 can be represented by signal 31, having the same set of incompatible signals. Later on, we can fit signal 22 in the signal scheme manually.

Intersection 1				Intersection 2			
i	w_{i1}	λ_{i1}	μ_{i1}	i	w_{i2}	λ_{i2}	μ_{i2}
2	0.447	0.2472	0.7778	2	0.165	0.1750	0.7778
5	0.065	0.0361	0.7778	5	0.253	0.2694	1.2778
8	0.276	0.1528	0.5000	8	0.251	0.2667	0.7778
11	0.211	0.1167	0.5000	11	0.332	0.3528	1.2778

Table 8.1: Input of intersections

From figure 8.1, we can derive which approaches (and signals) are incompatible. At this intersection, partial conflicts are permitted. For example, approach 8 and 33 can have right of way simultaneously. Possible conflicts between road users are solved by following the normal traffic rules. In Appendix E can be found, which approaches and corresponding signals are incompatible with each other, as well as the clearance-times belonging to these incompatible signals.

In the rush-hours, the roads in between the two intersections are congested. The vehicles waiting for signal 2 of intersection 1 and for signal 8 of intersection 2, will affect the traffic flow of the intersections in a negative way. In Appendix E.1 the arrival intensities at the different signals, spread out over the different approaches are schematically presented. From this we conclude, that it is most obvious that signals 8 from intersection 1 and 2 are phased. In section 8.3 we present the results of the two different ways, in which the signals can be phased.

8.2 Current situation

The current signal scheme is presented in table 8.2. The cycle time is 60 seconds. Note that the two intersections are not phased.

Intersection 1				Intersection 2			
i	b_{i1}(s)	e_{i1}(s)	g_{i1}(s)	i	b_{i2}(s)	e_{i2}(s)	g_{i2}(s)
2	30.0	56.0	26.0	2	0.0	22.0	22.0
5	2.0	14.0	12.0	5	30.0	53.0	23.0
8	30	56.0	26.0	8	0.0	21.0	21.0
11	2.0	18.0	16.0	11	30.0	53.0	23.0
31	3.0	21.0	18.0	22	27.0	56.0	29.0
33	25.0	55.0	30.0	24	59.0	24.0	25.0
35	9.0	23.0	14.0	26	28.0	56.0	28.0
37	23.0	57.0	34.0	28	59.0	24.0	25.0
				31	27.0	55.0	28.0
				33	59.0	21.0	22.0
				35	28.0	55.0	27.0
				37	59.0	21.0	22.0

Table 8.2: Current signal scheme

8.3 Results

8.3.1 Current way of phasing signals

In table 8.3 the signal scheme with a cycle time of 50 seconds and a phase green-time of 25 seconds is given.

Intersection 1				Intersection 2			
i	b_{i1}(s)	e_{i1}(s)	g_{i1}(s)	i	b_{i2}(s)	e_{i2}(s)	g_{i2}(s)
2	0.0	25.0	25.0	2	2.1	31.1	29.0
5	30.0	47.0	17.0	5	35.1	0.1	35.0
8	0.0	25.0	15.0	8	2.1	31.1	29.0
11	30.0	47.0	17.0	11	35.1	0.1	35.0
31	31.0	43.0	12.0	22	36.1	49.1	13.0
33	1.0	24.0	23.0	24	4.1	27.1	23.0
35	33.0	45.0	12.0	26	36.1	49.1	13.0
37	2.0	26.0	24.0	28	4.1	27.1	23.0
				31	36.1	48.1	12.0
				33	4.1	24.1	20.0
				35	36.1	48.1	12.0
				37	4.1	24.1	20.0

Table 8.3: Signal scheme, where the intersections are phased according to the current method

Note that signals 8 of intersection 1 and 2 are phased, but not completely: a_{81} is the amount of traffic coming from intersection 1 during one cycle (the phased signal 8 excluded) and is in this case given by:

$$\begin{aligned} a_{81} &= \frac{355+86}{3600} \cdot 50 \\ &= 6.125 \end{aligned}$$

The effective green-time e_{81} of signal 8 on the first intersection is equal to the phase green-time of 25 seconds. The effective green-time e_{82} of signal 8 on the second intersection has to be:

$$\begin{aligned} e_{82} &= e_{81} + \frac{a_{81}}{\mu_{82}} \\ &= 25.0 + \frac{6.125}{0.7778} \\ &= 32.9 \text{ seconds} \end{aligned}$$

Nevertheless, with a cycle time of 50 seconds, this yields an unstable system. The effective green-time e_{82} for which the system is just stable, is given by 29.0 seconds.

The beginning of the effective green-time b_{82} is then given by:

$$\begin{aligned} b_{82} &= b_{81} + r_{12} - \frac{a_{81}}{\mu_{82}} \\ &= 0.0 + 10.0 - \frac{6.125}{0.7778} \\ &= 2.1 \text{ seconds} \end{aligned}$$

The end of the effective green-time is given by $2.1 + 29.0 = 31.1$ seconds.

So the signals are partly phased, meaning that not the whole platoon can pass the signal before it turns red again.

8.3.2 New way of phasing signals

For various cycle times and phase green-times, we have determined a fixed time-control, where signal 8 of intersection 1 and 2 are phased. In table 8.4 the objective values for a cycle time c between 45 and 60 seconds and a phase green-time p between 15 and 24 are given. Some combinations yield an unstable system, which is indicated with '-'.

	p-15	p-16	p-17	p-18	p-19	p-20	p-21	p-22	p-23	p-24
c-45	34.81	31.73	-	-	-	-	-	-	-	-
c-46	37.70	32.30	**31.36**	-	-	-	-	-	-	-
c-47	43.51	33.40	31.46	31.74	-	-	-	-	-	-
c-48	59.90	35.16	31.98	31.40	32.96	-	-	-	-	-
c-49	432.96	38.02	32.84	31.56	31.90	35.89	-	-	-	-
c-50	-	43.22	34.10	32.02	31.68	33.08	46.54	-	-	-
c-51	-	55.73	35.91	32.72	31.84	32.25	35.51	-	-	-
c-52	-	135.57	38.64	33.69	32.24	32.07	33.33	42.35	-	-
c-53	-	-	43.28	34.97	32.82	32.20	32.65	35.40	271.46	-
c-54	-	-	53.29	36.75	33.60	32.55	32.49	33.67	40.31	-
c-55	-	-	93.85	39.33	34.60	33.05	32.62	33.08	35.46	70.64
c-56	-	-	-	43.53	35.90	33.71	32.93	32.96	34.05	39.23
c-57	-	-	-	51.84	37.63	34.53	33.38	33.08	33.55	35.64
c-58	-	-	-	77.74	40.08	35.55	33.95	33.37	33.46	34.47
c-59	-	-	-	-	43.92	36.85	34.65	33.77	33.58	34.05
c-60	-	-	-	-	50.99	38.54	35.50	34.28	33.84	33.98

Table 8.4: Table of the objective value as function of the cycle time and phase green-time: the sum over both intersections of the weighted average delay for various values of c and p

The objective value is minimized when $c = 46$ and $p = 17$. In table 8.5 the 'optimal' signal scheme with a cycle time of 46 seconds and a phase green-time of 17 seconds is given. Signals 8 from intersection 1 and 2 are phased.

Intersection 1				Intersection 2			
i	b_{i1}(s)	e_{i1}(s)	g_{i1}(s)	i	b_{i2}(s)	e_{i2}(s)	g_{i2}(s)
2	43.0	17.0	20.0	2	2.8	27.0	24.2
5	21.0	38.0	17.0	5	30.0	44.8	14.8
8	0.0	17.0	17.0	8	2.8	27.0	24.2
11	21.0	38.0	17.0	11	30.0	44.8	14.8
31	25.0	40.0	15.0	22	31.0	43.8	12.8
33	44.0	15.0	17.0	24	6.8	25.0	18.2
35	23.0	35.0	12.0	26	31.0	43.8	12.8
37	42.0	15.0	19.0	28	6.8	25.0	18.2
				31	31.0	42.8	11.8
				33	6.8	22.0	15.2
				35	31.0	42.8	11.8
				37	6.8	22.0	15.2

Table 8.5: Signal scheme, where the intersections are phased according to the new method

Note that signals 8 of intersection 1 and 2 are phased. In this case a_{qk-1} is given by:

$$\begin{aligned} a_{81} &= \frac{355 + 86}{3600} \cdot 46 \\ &= 5.635 \end{aligned}$$

The effective green-time e_{81} of signal 8 on the first intersection is equal to the phase green-time of 17 seconds. The effective green-time e_{82} of signal 8 on the second intersection has to be:

$$\begin{aligned} e_{82} &= e_{81} + \frac{a_{81}}{\mu_{82}} \\ &= 17.0 + \frac{5.635}{0.7778} \\ &= 24.2 \text{ seconds} \end{aligned}$$

The beginning of the effective green-time b_{82} is then given by:

$$\begin{aligned} b_{82} &= b_{81} + r_{12} - \frac{a_{81}}{\mu_{82}} \\ &= 0.0 + 10.0 - \frac{5.635}{0.7778} \\ &= 2.8 \text{ seconds} \end{aligned}$$

8.4 Simulation program

To test whether the developed solution method works well, a simulation program has been written. It simulates a fixed time control on multiple intersections of an arterial. In this section the structure of this simulation program is explained.

8.4.1 Input

The input needed to simulate the system is summarized below:

$$\begin{aligned} K &= \text{Set of intersections (index } k) \\ I_k &= \text{Set of traffic signals on intersection } k \text{ (index } i) \\ c &= \text{Cycle time} \\ \lambda_{ik} &= \text{Arrival rate of vehicles at traffic signal } i \text{ on intersection } k \\ \mu_{ik} &= \text{Departure rate of vehicles at traffic signal } i \text{ on intersection } k \\ b_{ik} &= \text{Time (mod } c) \text{ when signal } i \text{ turns green on intersection } k \text{ (range } [0, c)) \\ e_{ik} &= \text{Time (mod } c) \text{ when signal } i \text{ turns red on intersection } k \text{ (range } [0, c)) \end{aligned}$$

The vehicles at signals i (on intersection k) on the 'outside' of the arterial arrive according to a Poisson process with parameter λ_{ik}. Vehicles 'within' the arterial move in platoons from one intersection to the other. The departure process consists of deterministic departures. Each $1/\mu_{ik}$ seconds a vehicle leaves approach i_k. The b_{ik} and e_{ik} are chosen in such a way that incompatible approaches don't have right of way simultaneously.

8.4.2 Discrete-event simulation

We have written a discrete-event simulation with an event scheduling approach. The event scheduling approach concentrates on the events and how they affect the system. The three events that can be distinguished are:

1. Arrival of a vehicle
2. Beginning of the effective green-time
3. End of the effective green-time

The departure of a vehicle can be seen as event four. But to obtain a value for the waiting time of a vehicle, the order in which vehicles have arrived needs to be stored. We didn't make use of this departure-event, but kept track of the amount of 'work', c.q. the time to clear the vehicles, waiting in the queue. Out of the amount of departure time waiting at the signal and the beginning and end of the effective green-times, the departure time and subsequently the waiting time can immediately be calculated when a vehicle arrives.

When for example, two approaches belong to the same traffic signal, we doubled the departure rate of this signal. This is not valid when a vehicle arrives and there is no waiting queue. In that case the departure rate is equal to the original departure rate instead of the doubled one.

We keep track of the time points at which the next events of the different types occur. To record these time points we make use of a binary search tree. In a tree, the time points are not ordered in a straight line, like earliest event first, and so on. Instead, the starting time point, called the root, is linked to two other nodes, called its children, and those nodes in turn are linked to other children, and so on. Formally, a tree is either empty, or a root, which is connected to one or more other trees, called the subtrees of the root. The order of all time points in this tree is important. Formally, in a binary search tree the following holds:

- All the time points in the left subtree take place earlier than the time point of the root
- All the time points in the right subtree take place later than the time point of the root
- The left and right subtrees are also binary search trees

We can conclude, that the event that takes place first, is the leftmost node in the tree. We use a binary search tree in order to minimize the distance we have to go to reach any given element. Searching for an element in a binary tree containing n nodes is an $O(\log n)$ process and building the tree in the first place is an $O(n \log n)$ process, if the tree is reasonably well balanced.

The simulation then consists of finding the smallest time point in this tree, setting the current time to this event time and executing the corresponding activities. Here we will describe how these events affect the system and what activities are carried out.

1. Arrival of a vehicle

With an arrival, the total amount of vehicles waiting at that approach is increased and the delay of that particular vehicle is computed. When the arrival is on an approach on the 'outside' of the arterial, a new arrival at the same approach is simulated. When the vehicle doesn't leave the arterial, but drives to another intersection of the arterial, the new destination and the exact time of arrival at this destination is determined.

2. Beginning of the effective green-time

When a traffic signal turns green, the state of this signal changes. Vehicles at the approach have right of way until the end of the effective green-time. A new event, the beginning of the effective green-time in the next cycle, is generated and added to the binary search tree.

3. End of the effective green-time

When a traffic signal turns red, the state of this signal changes. A new event, the end of the effective green time, so the beginning of the effective red-time, is generated and added to the binary search tree. When the signal turns red and a vehicle is not completely driven off, two different scenarios can be followed. We have chosen for preemptive resume, that means, the drive-off time of the vehicle is preempted, but is continued at the beginning of the next effective green-time.

After each event the total number of waiting vehicles at the approaches is updated. As stopping criterium for this simulation the runlength is taken. When the current simulation time exceeds this runlength the simulation is ended.

8.5 Analysis versus simulation

In this section three signal schemes for the intersections are simulated with the program, described in the previous section. The objective to be minimized is equal to the sum over the intersections of the weighted average delay. The weight w_{ik} for signal i on intersection k is given by:

$$w_{ik} = \frac{\lambda_{ik}}{\sum_j^{n_k} \lambda_{jk}}$$

First, the current situation, with a cycle time of 60 seconds is investigated. In the current situation the intersections are not phased. Secondly, we investigate the case where the intersections are phased according to the existing method. The cycle time c is taken equal to 50 seconds and the phase green-time p is taken equal to 25 seconds. Finally, the fixed time-control determined with the new way of phasing signals is simulated, with a cycle time of 46 seconds and a phase green-time of 17 seconds.

For each case the average delay is calculated. From this, the objective immediately follows. In the second and third table, the approximation of the average delay based on Webster's formula (without correction term), is given as well.

Besides this, the degree of saturation (ρ^*_{ik}) is calculated. This degree indicates the fraction of the maximum capacity that is utilized and can be computed as follows:

$$\rho^*_{ik} = \frac{\lambda_{ik} c}{\mu_{ik} g_{ik}}$$

For reliability of the results and small confidence intervals, the system is simulated for 24 hours = 86400 seconds and with 100 repetitions. A 95%-confidence interval for the average delay is constructed.

8.5.1 Current situation - Cycle time 60 seconds

In table 8.6 the simulation results as well as the input and saturation degree of the current situation are presented. It should be observed that the degree of saturation of signal 8 on intersection 2, ρ^*_{82} is equal to 0.997. It is remarkable that the high degree of saturation doesn't yield a large average delay. Probably the reason for this is that signal 8 on intersection 2 is phased partly with signal 5, 8 and 11 of intersection 1.

Intersection 1					
i	λ_{i1}	μ_{i1}	g_{i1}(s)	ρ^*_{i1}	d_{i1}(s)
2	0.2472	0.7778	26.0	0.733	18.94(± 0.002)
5	0.0361	0.7778	12.0	0.232	21.46(± 0.010)
8	0.1528	0.5000	26.0	0.705	17.94(± 0.008)
11	0.1167	0.5000	16.0	0.875	41.56(± 0.085)
Weighted average delay: 23.60 seconds					
Intersection 2					
i	λ_{i2}	μ_{i2}	g_{i2}(s)	ρ^*_{i2}	d_{i2}(s)
2	0.1750	0.7778	22.0	0.614	17.41(± 0.005)
5	0.2694	1.2778	23.0	0.550	22.27(± 0.003)
8	0.2667	0.7778	21.0	0.977	14.71(± 0.002)
11	0.3528	1.2778	23.0	0.720	17.13(± 0.002)
Weighted average delay: 17.87 seconds					
Objective: 41.47 seconds					

Table 8.6: Output of current situation

8.5.2 Current way of phasing signals - Cycle time 50 seconds

In table 8.7 the simulation results of the new signal scheme with a cycle length of 50 seconds are presented.

Intersection 1						
i	λ_{i1}	μ_{i1}	g_{i1}(s)	ρ_{i1}^*	d_{i1}(s)	d_{i1}^{app}(s)
2	0.2472	0.7778	25.0	0.636	16.28($\pm$ 0.002)	12.13
5	0.0361	0.7778	17.0	0.137	10.96($\pm$ 0.008)	12.18
8	0.1528	0.5000	25.0	0.611	23.26($\pm$ 0.028)	12.92
11	0.1167	0.5000	17.0	0.686	15.65($\pm$ 0.008)	21.78
Weighted average delay: 11.85 seconds						
Intersection 2						
i	λ_{i2}	μ_{i2}	g_{i2}(s)	ρ_{i2}^*	d_{i2}(s)	d_{i2}^{app}(s)
2	0.1750	0.7778	29.0	0.388	7.36($\pm$ 0.002)	6.86
5	0.2694	1.2778	15.0	0.703	51.78($\pm$ 0.0018)	19.47
8	0.2667	0.7778	29.0	0.591	3.61($\pm$ 0.001)	8.94
11	0.3528	1.2778	15.0	0.920	62.71($\pm$ 0.034)	37.92
Weighted average delay: 36.04 seconds						
Objective: 47.89 seconds						

Table 8.7: Output of new situation (current way op phasing signals)

8.5.3 New way of phasing signals - Cycle time 46 seconds

In table 8.8 the simulation results of the optimal signal scheme with a cycle length of 46 seconds are presented.

Intersection 1						
i	λ_{i1}	μ_{i1}	g_{i1}(s)	ρ_{i1}^*	d_{i1}(s)	d_{i1}^{app}(s)
2	0.2472	0.7778	20.0	0.731	16.28($\pm$ 0.002)	14.79
5	0.0361	0.7778	17.0	0.126	10.96($\pm$ 0.008)	9.83
8	0.1528	0.5000	17.0	0.827	23.26($\pm$ 0.028)	26.08
11	0.1167	0.5000	17.0	0.632	15.65($\pm$ 0.008)	16.56
Weighted average delay: 17.73 seconds						
Intersection 2						
i	λ_{i2}	μ_{i2}	g_{i2}(s)	ρ_{i2}^*	d_{i2}(s)	d_{i2}^{app}(s)
2	0.1750	0.7778	24.2	0.428	8.32($\pm$ 0.004)	7.55
5	0.2694	1.2778	14.8	0.655	14.65($\pm$ 0.004)	15.78
8	0.2667	0.7778	24.2	0.652	2.92($\pm$ 0.001)	10.15
11	0.3528	1.2778	14.8	0.858	19.20($\pm$ 0.014)	22.17
Weighted average delay: 12.18 seconds						
Objective: 29.91 seconds						

Table 8.8: Output of new situation (new way op phasing signals)

8.6 Discussion of results

From the simulation results in the previous subsection we conclude, that phasing the signals in the right way can solve the problem of congested roads. The average delay of vehicles at signal 8 of intersection 2, is 14.71, 3.61 and 2.92 seconds for the current situation, the current way of phasing signals and the new way of phasing signals respectively.

The objective value in case the signals are phased according to the new method is 29.91 seconds. This is an improvement of 27.9%, compared with the current situation. From the objective value in case the signals are phased according to the existing method, we conclude that this method doesn't always work properly. The reason that this value is so large, is that the effective green-time of signals 5 and 11 of intersection 2 is just long enough that the system is stable.

In table 8.7 and 8.8 the approximations of the average delay, based on Webster's formula without correction term are given. These approximations differ sometimes quite a lot from the simulation results. There are two reasons for this. The first reason is, that we didn't make use of Webster's formula with correction term, which is much more accurate than the formula without correction term. In chapter 4 more information about this formula and better alternatives is given. The reason that we didn't make use of Webster's formula with correction term, is that we were not aware of the impact of this correction term to the formula. Later we realized that this correction term is in the range 5 to 15 percent of the original formula of Webster, which is quite a lot. The second reason is, that the average delay of vehicles at signals 'inside' the arterial is also determined by Webster's formula. But this formula is only valid for an isolated intersection, where the vehicles are assumed to arrive according to a Poisson process, instead of arriving in platoons.

In future research this method can be extended, where the arrival process within the arterial is also modelled correctly. Then a formula of the delay of vehicles within the arterial has to be determined as a function of the effective green-time, cycle time, the arrival rate and departure rate. Then this formula can be used in the same way in the optimization as the formula of Webster.

Possibly this method can be used for determining a fixed-time control for a network of intersections. Probably the method has to be adapted, to handle difficult routing issues that occur in networks.

APPENDIX A

DERIVATIVE OF WEBSTER'S FORMULA

$$d_i^{WEB}(g_i) = \frac{(c-g_i)^2}{2c(1-\rho_i)} + \frac{\rho_i c^2}{2g_i(\mu_i g_i - \lambda_i c)} \tag{A.1}$$

$$d_i^{WEBACC}(g_i) := \frac{d}{\partial g_i} d_i^{WEB}(g_i) = \frac{g_i - c}{c(1-\rho_i)} - \frac{(2\mu_i g_i - \lambda_i c)\rho_i c^2}{2g_i^2(\mu_i g_i - \lambda_i c)^2} \tag{A.2}$$

APPENDIX B

OPTIMIZATION OF FUHRMANN & WANG OPTIMIZATION PROBLEM

Minimize:

$$f(\bar{k}) = \sum_{i=1}^{n} c_i \lambda_i \frac{1-\rho_i+\frac{\rho_i}{k_i}(1+\frac{1}{1-\rho})}{1-\frac{\lambda_i}{k_i}E[C]} B E[C]$$

Subject to:

$$\sum_{i=1}^{n} \gamma_i k_i \quad \leq \quad K$$

Rewriting $f(\bar{k})$ yields:

$$\begin{aligned} f(\bar{k}) &= \sum_{i=1}^{n} \frac{c_i \lambda_i \rho_i (1+\frac{1}{1-\rho}) B E[C]}{-\lambda_i E[C]} + \frac{B c_i ((-2+\rho)\rho_i + E[C]\lambda_i(-1+\rho+\rho_i-\rho\rho_i))}{(-1+\rho)(1-\frac{\lambda_i}{k_i}E[C])} \\ \frac{df(\bar{k})}{\partial k_i} &= \frac{B c_i ((-2+\rho)\rho_i + E[C]\lambda_i(-1+\rho+\rho_i-\rho\rho_i))\frac{\lambda_i E[C]}{k_i^2}}{(1-\rho)(1-\frac{\lambda_i}{k_i}E[C])^2} \end{aligned}$$

From the gradient Karush-Kuhn-Tucker Necessary Conditions we know that:

$$\begin{aligned} \frac{df(\bar{k})}{\partial k_i} + L\gamma_i &= 0 \\ \frac{B c_i ((-2+\rho)\rho_i + E[C]\lambda_i(-1+\rho+\rho_i-\rho\rho_i))\frac{\lambda_i E[C]}{k_i^2}}{(1-\rho)(1-\frac{\lambda_i}{k_i}E[C])^2} + L\gamma_i &= 0 \end{aligned}$$

with $E[C] = \frac{s}{1-\rho}$.

$$\begin{aligned} k_i &= \frac{\lambda_i s}{1-\rho} + \frac{\sqrt{B c_i L(-1+\rho)^3 s \gamma_i \lambda_i \delta_i}}{L(-1+\rho)^3\gamma_i} \\ &= \frac{\lambda_i s}{1-\rho} + \frac{\sqrt{B}}{\sqrt{L}} \frac{\sqrt{c_i \lambda_i \delta_i s}}{\sqrt{\gamma_i(-1+\rho)^3}} \\ \text{with } \delta_i &= \rho_i(2-\rho) + \lambda_i s(1-\rho_i) \end{aligned}$$

So, we derived an expression for k_i.

Equality of the constraint $\sum_{i=1}^{n} \gamma_i k_i \leq K$ must hold. Out of this we can derive an expression for $\frac{\sqrt{B}}{\sqrt{L}}$, which is used in the formula for k_i again:

$$\sum_{i=1}^{n} \gamma_i k_i = \sum_{i=1}^{n} \gamma_i \frac{\lambda_i s}{1-\rho} + \sum_{i=1}^{n} \gamma_i \frac{\sqrt{c_i \lambda_i \delta_i s}}{\sqrt{\gamma_i (-1+\rho)^3}} \frac{\sqrt{B}}{\sqrt{L}} \quad = \quad K$$

$$k_i^* \quad = \quad \frac{\lambda_i s}{1-\rho} + (K - \sum_{j=1}^{n} \gamma_j \frac{\lambda_j s}{1-\rho}) \frac{\sqrt{c_i \lambda_i \delta_i / \gamma_i}}{\sum_{j=1}^{n} \gamma_j \sqrt{c_j \lambda_j \delta_j / \gamma_j}}$$

APPENDIX C

INPUT INTERSECTION CHAPTER 3

Incompatible traffic signals

	s-2	s-5	s-8	s-9	s-10	s-11	s-12	s-21	s-23	s-25	s-27
s-2		x		x	x	x	x	x		x	
s-5	x			x			x		x		x
s-8		x				x	x	x		x	
s-9	x	x				x	x			x	x
s-10	x									x	x
s-11	x		x	x					x		x
s-12	x	x	x	x				x			x
s-21	x		x				x				
s-23		x				x					
s-25	x		x	x	x						
s-27		x		x	x	x	x				

Clearance-times

	s-2	s-5	s-8	s-9	s-10	s-11	s-12	s-21	s-23	s-25	s-27
s-2		0		0	8	6	5	0		10	0
s-5	5			1			1		0		10
s-8		7				1	3	10		1	
s-9	9	6				3	4			1	13
s-10	0									2	0
s-11	0		6	4					6		0
s-12	0	5	3	2				9			1
s-21	0		0				0				
s-23		2				0					
s-25	0		2	0	1						
s-27		2		0	7	7	7				

APPENDIX D

INPUT INTERSECTION SECTION 6.8

Incompatible traffic signals

	s-1	s-2	s-8	s-9	s-11	s-25	s-26	s-27	s-28	s-43	s-49
s-1				x				x	x	x	x
s-2				x	x	x	x			x	x
s-8					x	x	x			x	x
s-9	x	x			x	x	x	x	x		
s-11		x	x	x		x	x	x	x	x	x
s-25		x	x	x	x					x	x
s-26		x	x	x	x					x	x
s-27	x			x	x						
s-28	x			x	x						
s-43	x	x	x		x	x	x				
s-49	x	x	x		x	x	x				

Clearance-times

	s-1	s-2	s-8	s-9	s-11	s-25	s-26	s-27	s-28	s-43	s-49
s-1				0				4	5	1	0
s-2				0	1	2	6			0	0
s-8					0	0	0			0	0
s-9	1	0			1	0	0	2	6		
s-11		0	0	0		0	3	2	0	1	0
s-25		0	1	2	4					0	0
s-26		0	2	1	0					0	4
s-27	0			0	1						
s-28	0			0	2						
s-43	0	0	2		1	8	6				
s-49	8	7	3		2	4	0				

APPENDIX E

INPUT INTERSECTION CHAPTER 8

Incompatible traffic signals ‘intersection 1’

	s-2	s-5	s-8	s-11	s-31	s-33	s-35	s-37
s-2		x		x	x		x	
s-5	x		x			x		x
s-8		x		x	x		x	
s-11	x		x			x		x
s-31	x		x					
s-33		x		x				
s-35	x		x					
s-37		x		x				

Clearance-times ‘intersection 1’

	s-2	s-5	s-8	s-11	s-31	s-33	s-35	s-37
s-2		5		5	2		8	
s-5	4		4			2		6
s-8		5		5	6		2	
s-11	4		4			6		2
s-31	8		8					
s-33		6		6				
s-35	6		6					
s-37		4		4				

Incompatible traffic signals ‘intersection 2’

	s-2	s-5	s-8	s-11	s-31	s-33	s-35	s-37
s-2		x		x	x		x	
s-5	x		x			x		x
s-8		x		x	x		x	
s-11	x		x			x		x
s-31	x		x					
s-33		x		x				
s-35	x		x					
s-37		x		x				

Clearance-times ‘intersection 2’

	s-2	s-5	s-8	s-11	s-31	s-33	s-35	s-37
s-2		4		4	2		5	
s-5	3		3			2		5
s-8		4		4	5		2	
s-11	3		3			5		2
s-31	4		4					
s-33		8		8				
s-35	4		4					
s-37		8		8				

E.1 Arrival intensities per hour

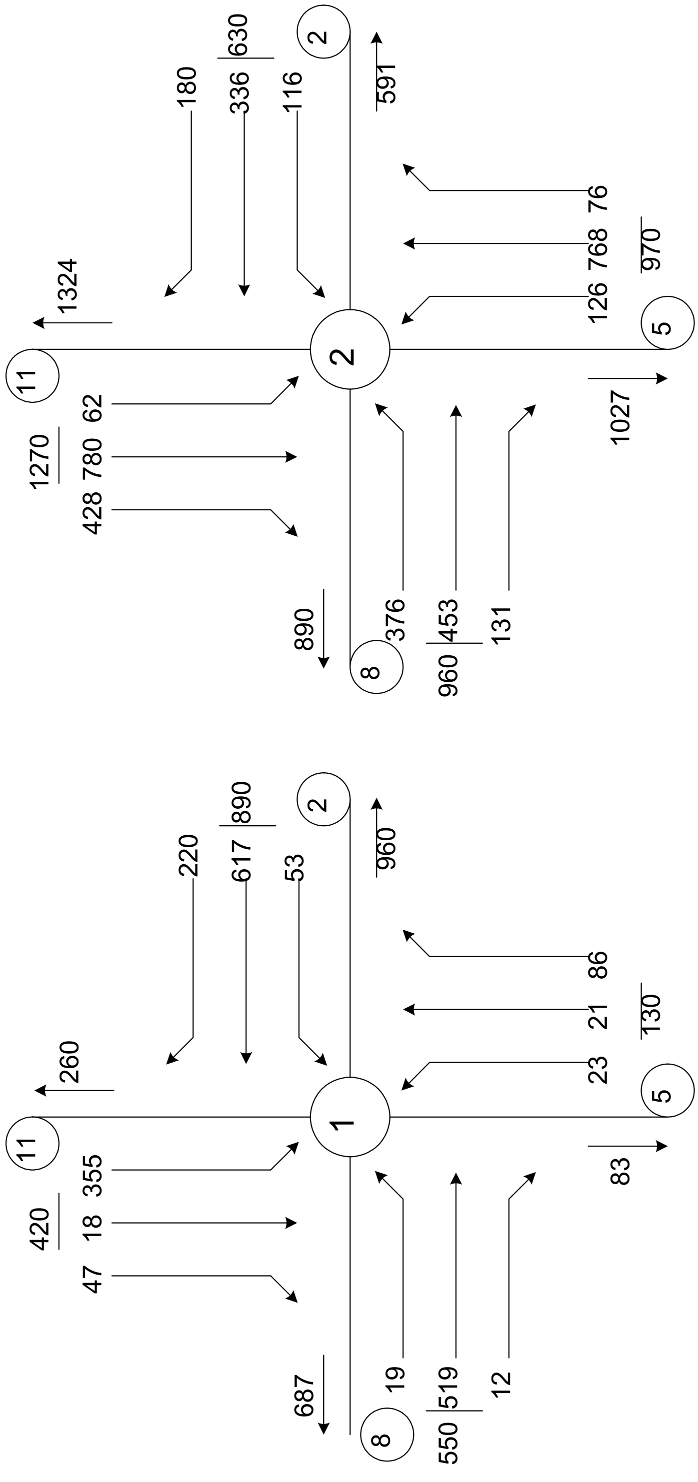


Figure E.1: Number of arriving vehicles per hour on the different approaches of the two intersections

APPENDIX F

PASCAL PROGRAM DETERMINING ALL PERMUTATIONS

```
Procedure Rotate(r_ag AS integer);
    VAR i, hulp AS integer;
    hulp := startseq(1);
    FOR i := 1 TO (r_ag − 1) DO
        startseq(i) := startseq(i + 1);
    startseq(1) := hulp;
END Procedure.


Procedure Permute(p_ag AS integer);
    VAR i AS integer;
    IF p_ag = 1 THEN
        FOR i := 1 TO (ag − 1) DO
            seq(i + 1) := startseq(i);
        seq(1) := 1;
    ELSE
        FOR i := 1 TO p_ag DO
            Rotate(p_ag);
            Permute(p_ag − 1);
        NEXT i;
    ENDIF;
END Procedure.


Main Program;
    VAR i AS integer;
    ag :=Number of subsets;
    FOR i := 1 TO (ag − 1) DO
        startseq(i) := i + 1;
    Permute(ag − 1);
END Main Program.
```

BIBLIOGRAPHY

[1] Adan, I.J.B.F. and Resing, J.A.C. (2001). *Queueing Theory*. Lecture Notes, Eindhoven University of Technology, The Netherlands.

[2] Beckmann, M.J., McGuire, C.B. and Winsten, C.B. (1956). *Studies in the Economics in Transportation*. New Haven, Yale University Press.

[3] Borst, S.C. (1996). *Polling Systems*. Amsterdam, The Netherlands.

[4] Borst, S.C., Boxma, O.J. and Levy, H. (1995). *The Use of Service Limits for Efficient Operation of Multi-Station Single Medium Communication Systems*. IEEE/ACM Transactions on Networking, Volume 3, No. 5, 602-612.

[5] Boxma, O.J. (1991). *Analysis and Optimization of Polling Systems*. Queueing Performance and Control in ATM, ITC-13, Amsterdam, Elsevier, 173-183.

[6] Cocon: http://www.dtvconsultants.nl/nl/cfsoftware.htm

[7] Darroch, J.N., Newell, G.F. and Morris, R.W.J. (1964). *Queues for a Vehicle-Actuated Traffic Light*. Operations Research, Vol. l2, No. 6, pp. 882-895.

[8] Everitt, D.E. (1986). *A Conservation-Type Law for Token Ring with Limited Service*. British Telecom Technology Journal 4, 51-61.

[9] Fedotkin, M.A. and Litvak, N.V. (1997). *An Adaptive Non-Linear Robust Traffic Control*. IEEE, 360-361.

[10] Fricker, C. and Jaïbi, M.R. (1994). *Monotonicity and Stability of Periodic Polling Models*. Queueing Systems 15, 211-238.

[11] Fuhrmann, S.W. and Cooper, R.B. (1985). *Stochastic Decompositions in the M/G/1 Queue with Generalized Vacations*. Operations Research 33, 1117-1129.

[12] Fuhrmann, S.W. and Wang, Y.T. (1988). *Analysis of Cyclic Service Systems with Limited Service: Bounds and Approximations*. Performance Evaluation 9, 34-54.

[13] Gazis, D.C. (1974). *Traffic Science*. John Wiley & Sons, New York.

[14] Harrison, P.G. and Knottenbelt, W.J. (2002). *Passage Time Distributions in Large Markov Chains*. Proc. ACM Sigmetrics 2002, London, 77-85.

[15] Improta, G. (1987). *Mathematical Programming Methods for Urban Network Control*. In Flow Control of Congested Networks, NATO ASI Series, Vol. F38, Edited by Odoni, A.R. et al., Springer-Verlag 35-67.

[16] Kleinrock, L. (1975). *Queueing Systems*. Volume 1: Theory. John Wiley & Sons, New York.

[17] Kuehn, P.J. (1979). *Multi-Queue Systems with Non-Exhaustive Cyclic Service*. The Bell System Technical Journal 58, 671-698.

[18] Miller, A.J. (1963). *Settings for Fixed-Cycle Traffic Signals*. Operational Research Quarterly 14, 373-386.

[19] Newell, G.F. (1965). *Approximation Methods for Queues with Application to the Fixed-Cycle Traffic Light*. SIAM Review 7, No. 2, 223-240.

[20] Ramanathan, B., McNally, M.G. and Jayakrishnan, R. (1995). *Formulation of Modern Signal Control Operations as a Non-Linear Mixed Integer Program*. Proceedings of the Sixth Vehicle Navigation and Information Systems Conference, Institute of Electrical and Electronics Engineers, Piscataway, 165-171.

[21] Riedel, T. and Brunner, U. (1992). *A Control Algorithm for Traffic Lights*. Automic Control Laboratory, Swiss Federal Institute of Technology, Switzerland.

[22] Shimizu, H., Sobata, Y., Kobayashi, M. and Yonezawa, Y. (2001). *Signal Control of Congestion Lengths in a Two-Way Traffic Network*. IEEE, 237-242.

[23] Taale, H. (2002). *Comparing Methods to Optimise Vehicle Actuated Signal Control*. AVV Transport Research Centre and Delft University of Technology, The Netherlands.

[24] Takagi, H. (1986). *Analysis of Polling Systems*. The MIT Press, Cambridge, Massachusetts.

[25] Tijms, H.C. (1994). *Stochastic Models, An Algorithmic Approach*. John Wiley & Sons, New York.

[26] Webster, F.V. (1958). *Traffic Signal Settings*. Road Research Technical Paper No. 39, London, England.

SAMENVATTING

Verkeerslichten regelingen zijn vandaag de dag een belangrijk middel om de doorstroming van verkeer in met name de stad te verbeteren. Hiermee wordt niet alleen de doorstroming bevorderd, maar wordt tevens de veiligheid van gevaarlijke kruispunten vergroot. Met een grote toename van het aantal motorvoertuigen op de weg, neemt ook de hoeveelheid files in de stad toe. Hierdoor wordt de afstelling van verkeerslichten binnen huidige regelingen en het ontwerpen van nieuwe verkeerslichten regelsystemen een steeds belangrijker onderwerp. Binnen de stad Eindhoven is men zich steeds meer bewust van het belang van een goede afstelling van verkeerslichten. Immers, door het ontstaan van steeds meer files ook binnen Eindhoven, zijn slechts twee oplossingen mogelijk. Of de infrastructuur en wegenstructuur worden dusdanig aangepast dat fileproblemen worden opgelost, of door de goede afstelling van verkeerslichten wordt de doorstroming sterk verbeterd en zijn de fileproblemen (tijdelijk) verdwenen. De laatste oplossing is veruit de goedkoopste en minst ingrijpende en verdient daarom op dit moment veel aandacht.

In dit rapport worden verscheidene verkeerslichten regelsystemen wiskundig geanalyseerd. In de praktijk worden er twee soorten systemen onderscheiden, te weten het starre regelsysteem en het dynamische regelsysteem. Op basis van de analyses van deze systemen en op basis van wiskundige modellen worden verbeteringen voorgesteld. Deze verbeteringen worden aan de hand van praktijkvoorbeelden getoetst, waarbij in alle gevallen de huidige en verbeterde regelsystemen worden gesimuleerd en de resultaten met elkaar worden vergeleken.

Hoofdstuk 2 tot en met 6 hebben alle betrekking op verkeerslichten regelsystemen van één geïsoleerd kruispunt. In de laatste twee hoofstukken zal een begin worden gemaakt met de problematiek omtrent de afstelling van verkeerslichten binnen een groot netwerk van kruispunten. In dit rapport wordt er bij alle problemen naar gestreefd, de doorstroming van het verkeer zo goed mogelijk te laten zijn. Om dit te bereiken wordt de totale (gewogen) gemiddelde wachttijd binnen een systeem geminimaliseerd.

In hoofdstuk 2 onderzoeken we nieuwe mogelijkheden wat betreft de bepaling van een starre verkeerslichten regeling op een geisoleerd kruispunt. Bij een starre afstelling worden de cyclustijd, de effectieve groenlengtes en de volgorde van deze groenlengtes vast genomen. Het probleem is als een optimaliseringsprobleem gemodelleerd en er is met behulp van optimaliseringssoftware een starre regeling bepaald. Vervolgens is in het daaropvolgende hoofdstuk aan de hand van een praktijkvoorbeeld, waarbij één van de drukste kruispunten van Eindhoven is genomen, de werking van het nieuwe systeem vergeleken met de huidige regeling door middel van simulatie. De nieuwe regeling gaf grote verbeteringen in vergelijking met de huidige regeling.

Al in 1958 ontwikkelde Webster een formule, om de gemiddelde wachttijd binnen een starre regeling te benaderen. Deze wachttijdformule hangt af van de cyclustijd, de aankomst- en vertrekintensiteit en de effectieve groenlengte. Sinds die tijd is er veel onderzoek geweest naar andere en betere benaderingsformules, maar tot de dag van vandaag wordt de inmiddels bijna vijftig jaar oude formule nog steeds wereldwijd gebruikt. In hoofdstuk 4 van dit rapport is een nieuwe benaderingsformule afgeleid. Door middel van simulatie van 3000 verschillende gevallen is de werking van deze formule vergeleken met die van al bestaande formules. Hieruit kunnen we concluderen dat de nieuwe formule een grote verbetering is vergeleken met de huidige benaderingsformules.

Meer en meer wordt tegenwoordig bij het afstellen van verkeerslichten gebruik gemaakt van een dynamische regeling. Bij een dynamische regeling wordt ingespeeld op de aanwezigheid van veel of juist weinig verkeer. Het verkeer wordt gedetecteerd door middel van detectielussen die bij vrijwel ieder verkeerslicht onder de weg liggen. De verkeerslichten krijgen alleen maar groen, als er ook daadwerkelijk auto's staan te wachten. Vervolgens blijft het verkeerslicht groen, totdat er geen auto's meer staan te wachten of een maximale groenlengte is bereikt. In hoofdstuk 6 is een nieuwe methode ontwikkeld om deze maximale groenlengtes te bepalen. De manier waarop dit nu gebeurt is omslachtig en zeer tijdrovend. De huidige en nieuwe methode zijn wederom met elkaar vergeleken door simulatie. Hieruit kunnen we concluderen dat de grootste verbetering niet in het feit zit dat de nieuwe methode goede resultaten oplevert, maar juist in de zeer korte tijd waarmee de gehele dynamische regeling bepaald wordt.

Wanneer de maximale groenlengtes oneindig groot worden gekozen, dan zal het dynamische systeem resulteren in een 'exhaustive' systeem. Bij een exhaustive systeem, zal het verkeerslicht net zolang groen blijven, als er auto's aankomen of nog in de rij staan te wachten. In hoofdstuk 5 is dit systeem wiskundig geanalyseerd. Hierbij is een benaderingsmethode ontwikkeld om het gemiddelde en de variantie van de effectieve groenlengtes te bepalen. De resultaten van deze benadering zijn vergeleken met simulatieresultaten om zo een goed beeld te kunnen krijgen van de nauwkeurigheid van de door ons ontwikkelde methode. De methode levert resultaten op, die hooguit 3% afwijken van de simulatieresultaten.

In de laatste twee hoofdstukken zijn verschillende methoden onderzocht om tussen naburige kruispunten een groene golf te creëren. Een groene golf is gedefinieerd als een reeks van achtereenvolgens op groen springende verkeerslichten, die zo op elkaar afgestemd zijn dat het verkeer, bij het aanhouden van een bepaalde snelheid, op een bepaald traject niet hoeft te stoppen. In hoofdstuk 7 en 8 is de huidige methode vergeleken met een nieuw ontwikkelde methode. Bij de huidige methode wordt eerst voor het drukste kruispunt een regeling bepaald, waarna de overige kruispunten hierop worden afgestemd. De nieuwe methode gaat uit van een integrale aanpak, waarbij op basis van een cyclustijd en groene golf lengte een regeling voor het hele systeem wordt bepaald. We nemen die cyclustijd en groene golf lengte, waarbij de som over alle kruispunten van de gewogen gemiddelde wachttijd per kruispunt zo klein mogelijk is. Om de verschillende systemen met elkaar te kunnen vergelijken is een simulatie geschreven, die een ringlaan simuleert. Op basis van simulatieresultaten kunnen we concluderen dat de nieuwe methode in sommige gevallen tot verbetering zal leiden.

Ieder hoofdstuk is afzonderlijk afgesloten met conclusies en mogelijk aanbevelingen. Veelal zijn deze conclusies getrokken uit simulatieresultaten, waarbij huidige systemen en nieuwe systemen zijn gesimuleerd en de resultaten met elkaar zijn vergeleken. In alle gevallen zijn de nieuw ontwikkelde methoden als beste naar voren gekomen.