Fixed-Time Schedule Based Vehicle Actuated Controller Design for Networks of Intersections

J.J.H. Karouta

DC2015.069

Master Thesis

Supervisors: Dr.ir. A.A.J. Lefeber
Ir. S.T.G. Fleuren
Ir. J.J.A de Wit (Grontmij Nederland B.V.)

EINDHOVEN UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

GRONTMIJ NEDERLAND B.V.
TRANSPORTATION & MOBILITY

Eindhoven, September 16, 2015

Abstract

This research explores the possibilities of using fixed-time schedules as a basis for controlling networks of intersections, as well as using these schedules to produce vehicle actuated control strategies. To this end, a queueing model is devised to better understand the underlying dynamics in network scenarios controlled by fixed-time schedules. This model is used by an optimisation method to find the best joint performance of the network. Both the queueing model and the optimisation method are validated using microsimulations and are found to produce good results. Moreover, a method is proposed to obtain structural information and maximal bounds on green times from a fixed-time schedule, which can be used in vehicle actuated control strategies. The proposed method is found to be very useful and yields good results so far. Especially for modelling purposes, where many intersections and different scenarios need to be modelled, labour time can be reduced, as it is very time-consuming to produce vehicle actuated control strategies the conventional way. Recommendations include, among others: possible improvements to the network scenario model, and; validations for, and extensions to the generation of the elements for vehicle actuated control strategies.

Contents

Chapter 1

Introduction

The road network is a popular means of transportation for both leisure and business purposes. In the Netherlands, the amount of traffic in this network is continuously increasing [12], as shown in Figure 1.1, which causes capacity deficits and traffic congestions. Chronic capacity deficits need to be resolved, to allow this growth to persist. Moreover, the increase in traffic causes negative environmental effects, such as greenhouse gasses. Both for the environment and the convenience of the road users, it is of great importance to reduce the amount of congestions. The way intersections handle and dispatch the traffic can influence these congestions, and can affect the travel times in both positive and negative ways. Recently, a method to determine periodic Fixed-Time Schedules (FTSs) for isolated traffic intersections with constant arrival rates has been developed at the TU/e [3], which aims to minimise the average delay experienced by the road users. An FTS prescribes the starting and ending times for the green and amber periods. However, there are two steps that still need to be taken before a usable approach can be derived. Firstly, vehicle actuated schedules are usually implemented instead of FTSs. Secondly, an optimal FTS does not guarantee optimality within a network setting. Ultimately, a solution is sought to control networks of intersections, taking into account the separate vehicle arrivals.

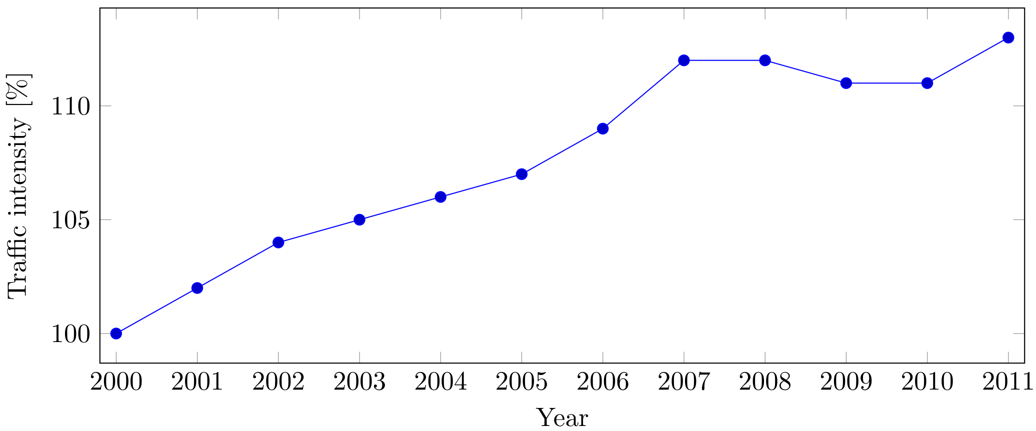


Figure 1.1: Relative traffic intensity on Dutch national roads on working days, 2000 = 100%. Source: CBS [12].

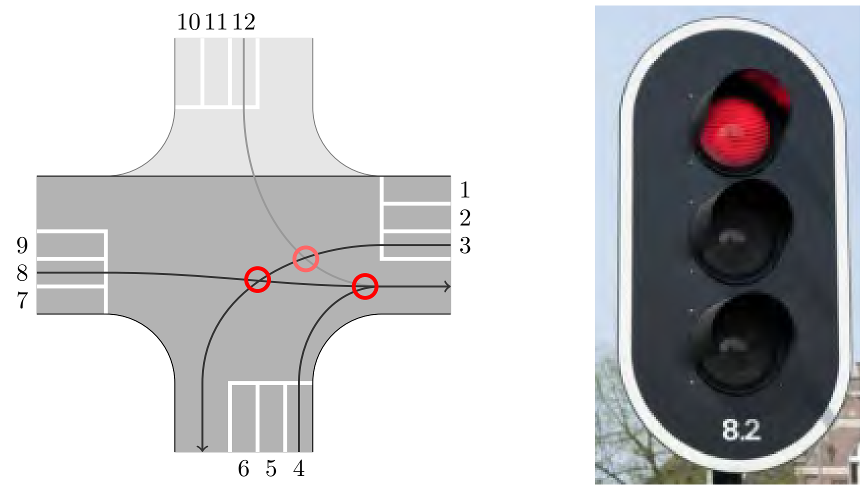
This thesis is conducted in cooperation with Grontmij Nederland B.V., a Dutch engineering consultancy, with expertise in multiple disciplines, of which Transportation and Mobility is the division hosting this project. One key aspect of the cooperation is the knowledge transfer from the industry to the academia, and vice versa. For the university, it is important to understand what the industry needs, and Grontmij Nederland B.V. benefits from innovative and progressive technology, which the academia can provide. Previous research on this topic has been conducted in exploring networks of intersections [10], in a similar cooperative way. Moreover, software to generate FTSs, using the fixed-time optimisation method based on [3], is used in this project.

This report first covers some basic background and terminology in Chapter 2, after which the tool from [3] is compared to COCON [2], the tool currently used most to produce an FTS, in Chapter 3. Chapter 4 covers the extension of the model proposed in [10], in which the schedules of intersections are adjusted to optimise traffic flows. It continues by validating the extended model using microscopic simulations, followed by the conclusions and drawbacks encountered using this method. Because FTSs are rarely used in practice, Chapter 5, looks into the possibilities of generating Vehicle Actuated Control (VAC) strategies, in which the control sequence depends on the current vehicle arrivals, instead of a fixed schedule. An approach is proposed to generate useful information from an FTS, which can be used in the design of VACs. Finally, this report is concluded with the conclusions and recommendations drawn during the project, in Chapter 6.

Chapter 2

Background

This chapter explains the background necessary to understand this report. Firstly, it discusses the terminology used in this report and in traffic control. Also conventions and guidelines are briefly mentioned. Secondly, Section 2.2 is dedicated to explain how a basic control strategy is visualised and how it is used in this report. Both sections are based on the illustrations in Figure 2.1.



(a) (b)

Figure 2.1: Examples used to explain the terminology: a shows an example of an intersection, the standard numbering of signal groups for motorised traffic, and conflicts between selected ones; b shows a standard signal head as used in the Netherlands, including the group number followed by the head number in case of multiple lights.

2.1 Terminology

Figure 2.1a shows a schematic representation of a standard signalised intersection with only the motorised traffic groups. A standard 4-way intersection generally consists of 12 origin-destination combinations, called signal groups. Each one of these signal groups can contain multiple traffic lights, called signal heads, as shown in Figure 2.1b. Within a group, all the signal heads are actuated together. In the Netherlands, the signal groups are numbered according to the following standardised rule [8]: the numbering usually starts on the east approach at the flow turning right (deviations in starting approach sometimes occur), the rest of the signal groups are numbered clockwise. Whenever an approach is not present at the junction, the associated numbers are skipped, and when flows are combined (multiple directions on 1 lane), the resulting signal group is numbered as the one going straight (e.g. lane 4 and 5 are combined, resulting in one signal group numbered 5). For the sake of simplicity in this example, a three way junction is used, missing the North bound approach (lane 10, 11, and 12). Of course, in the Netherlands, bicycle lanes are common, and indispensable. Numbers 21-28 are reserved for the bicycle traffic flows, and 31-38 for the pedestrians. Two of each set are reserved for each approach, usually used to separate the flow in the median strip. The numbering is also applied clockwise and starting in the same approach as the regular traffic.

Further aspects that characterise a junction, are a conflict matrix C, and the arrival and processing rates of each signal group. Within the conflict matrix, component c_{ij} is 1 when signal group i and j are conflicting, as shown by a red circle in Figure 2.1a. A conflict means, the two signal groups cannot have a green light simultaneously. The resulting conflict matrix for the 3-way junction Figure 2.1a is shown in

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{2.1}$$

Note that the diagonal is always zero, as a signal group cannot have a conflict with itself, and that the indices used for the matrix are the sorted list of signal groups, i.e. $i, j \in \{2, 3, 4, 6, 7, 8\}$, i.e., $c_{2,6}$ is the conflict in the top row. Arrival rates are mostly obtained from survey data, or simulation models in case of predictions of future situations. The processing rates are based on the amount of lanes the corresponding signal group encompasses, the maximum allowed speed and other physical restrictions like obstacles or turns. Both rates are expressed in Personal Car Equivalent (PCE) per time unit (usually PCE/h). This measure is introduced to express the rates as a mixed traffic flow, consisting of cars, trucks, buses, and other traffic in one number [10, 13].

Besides the fixed characteristics of a junction, a less strict aspect is the clearance time between conflicting signal groups. The minimal time necessary between the start of a green period for group j after the start of the red period for group i is called clearance time σ_{ij}. There are guidelines and safety standards to keep in mind [13], but while complying to these constraints, the times can be chosen "freely". A possible clearance time matrix Σ is

$$\Sigma = \begin{bmatrix} \cdot & & & 2 & & \\ & \cdot & & 3 & 3 & 4 \\ & & \cdot & & & -1 \\ 1 & 2 & & \cdot & & 3 \\ & 0 & & & \cdot & \\ & 2 & 4 & 3 & & \cdot \end{bmatrix}. \tag{2.2}$$

Note that negative clearance times can be adopted; in this case $\sigma_{4,8} = -1$ because the distance for signal group 8 to reach the conflict area is larger than that of group 4, to the extent that the traffic of 8 can safely leave 1 second before the traffic of 4 stops. Although negative clearance times are not yet allowed in implemented schedules in the Netherlands, they are considered in this report. Moreover, clearance times can only apply to conflicting groups. Therefore, each entry of Σ corresponding to a non-conflicting pair of signal groups, is left blank. This means that the clearance time matrix is enough to describe the conflicts, and as such, the conflict matrix is not used separately in the course of the report.

Other parameters include *minimal green time*, *lost green time*, *used amber time* and *maximal saturation*, which all have some safety guidelines to follow, explained in [13], but can be used to express desired behaviour of the control scheme. A controlled junction will be referred to as an intersection, which, in its turn, is controlled by a Traffic Control Installation (TCI). The most basic way to control an intersection, is by using a Fixed-Time Schedule, which is introduced and explained in the next section along with the use of the mentioned design parameters.

2.2 Fixed Time Schedules

As mentioned in the previous section, an FTS is the most basic way to control an intersection. An FTS is a schedule, which prescribes at what moment a signal group is green and when it should switch back to amber, and subsequently, red. In this report, when mentioning an FTS, it also implies being a periodic program. An example of such a periodic schedule is illustrated in Figure 2.2.

Figure 2.2: A *phase diagram* representation of an FTS for the three-way junction of Figure 2.1a (without the North-bound approach). The numbers on the left represent the signal groups, and time is represented horizontally, within one period T.

The state a signal group is in (green, amber or red), is called a *phase*. A phase diagram, as shown in Figure 2.2, visualises these for each signal group in the TCI, at any time $t \in [0, T)$. Because vehicles are subject to human reaction time, the beginning of the green period is not utilised, while part of the amber period is. These *lost green time*, g^{lost}, and *used amber time*, y^{lost} must therefore be subtracted from, and added to the green time g, respectively, to form the *effective green period*, τ. This can be defined for each signal group i in

$$\tau_i = g_i + y_i^{\text{lost}} - g_i^{\text{lost}}. \tag{2.3}$$

The remaining time is called the effective red period

$$r_i = T - \tau_i. \tag{2.4}$$

Applying this transformation to the FTS from Figure 2.2, results in the *effective phase diagram* in Figure 2.3. This reduces the number of phases each signal group is subject to, and simplifies the modelling. Both the regular and the effective phase diagrams are used in this report, each for different purposes. For most calculations, the effective times are employed, whereas simulations require regular phase diagrams. Moreover, the regular ones are also used for most figures, as they are more relatable.

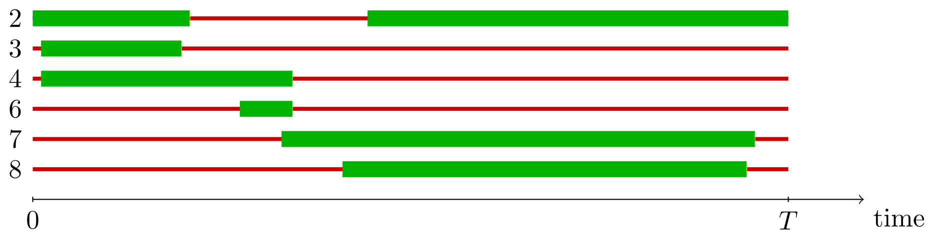


Figure 2.3: Effective phase diagram of the one from Figure 2.2

The repetitiveness of the FTS implies that one cycle should always be enough to process the average amount of traffic arriving within one period. Therefore, the effective green period should be long enough to eliminate the average maximal number of queued vehicles. This is captured in the capacity constraint in (2.5), in which λ_i and μ_i represent the average arrival rate and the processing rate, respectively, and ρ_i^* defines the saturation of the specific signal group i.

$$\rho_i^* = \frac{\lambda_i \cdot T}{\mu_i \cdot \tau_i} \leq 1. \tag{2.5}$$

Note the substantial difference between the saturation ρ_i^* and the load ρ_i, defined as

$$\rho_i = \frac{\lambda_i}{\mu_i}. \tag{2.6}$$

During this project, the FTSs are generated using a tool based on [3]. To support the use of this software, the next chapter compares the FTSs produced by this tool with those obtained using COCON [2], the conventional software used at Grontmij Nederland B.V.

Chapter 3

Comparison of Fixed Time Schedules

This chapter considers several FTSs produced using COCON [2], and compares them with schedules generated using the fixed-time optimisation method based on [3] (hereinafter Fixed-Time Tool (FTT)). COCON is a manual tool, that allows the user to produce FTSs visually, according to visualised bounds. This makes it possible to choose a certain order over another, even if this would result in a lower performance of the resulting FTS. As FTSs are rarely implemented as control strategies, Grontmij Nederland B.V. uses them to check capacity and as a basis for Vehicle Actuated Control (VAC). Their aim, therefore, is not to find the best schedule, but one that represents the needs. FTT, on the other hand, is fully automated and produces a complete, and near optimal solution. For a fair comparison, several measures are taken. Firstly, the FTSs obtained are chosen for having been produced with much consideration. Secondly, the signal group with the highest load is often the bottleneck for the performance of the entire schedule, and is therefore artificially kept below a certain value in COCON. Because the saturation can be constrained in FTT, the highest one found in the COCON databases is used as a maximum for all signal groups in FTT. Moreover, at the moment of writing, multiple green phases per period can not yet be implemented in FTT, as opposed to COCON.

The data used to compare the tools, as well as the design choices made, are shown in Appendix A.1. Both tools are compared with respect to the expected delay experienced by the road users, all computed using Akçelik's [1] expression, which is also used as an objective function in the optimisation of FTT. The results are shown in Table 3.1.

Table 3.1: Improvement in delay, computed using the expression of Akçelik.

case	# signal groups	% improvement
A.1.1 MR	6	1.2
A.1.1 ER	6	(-0.5) 0.0
A.1.2 MR	9	7.4
A.1.2 ER	9	0.0
A.1.3 Current	16	4.3
A.1.3 Desired	16	22.9
A.1.4 MR	29	30.0
A.1.4 ER	29	37.0

As can be seen in the results, using FTT generally generates schedules with a lower average delay. Especially for larger intersections, the delay can be greatly reduced. There are, however, two exceptions. In the first case (A.1.1 ER), FTT produces the exact same schedule. The negative improvement (-0.5) is the result of rounding the maximal allowed saturation to full percent instead of tenth, which requires a period time of one second longer to result in a feasible schedule. Rounding the maximal allowed saturation to the nearest tenth, in this case higher, results in the exact same schedule as COCON. The second exception (A.1.2 ER), is caused by disregarding multiple green periods in FTT. Generally, it is preferred to have an FTS with only one green phase per signal group, such to prevent confusion in the order of green phases, and consequently, mistakes in anticipations by road users. The reason why the COCON schedule included this second green phase, is because keeping the same order during the entire day is also desired, for the same anticipation reason as for the first aspect. Moreover, the second green phase is indispensable to produce a stable schedule with the same overall order as the morning rush. As multiple green phases cannot be computed yet, the obtained schedule does not have the same order, however, in this case results in the same delay. The question whether the same order is more important than single green phases, or vice versa, remains a preference depending on the specific intersection as well as the designer.

The great improvement for the large intersections is, most probably, caused by the high complexity of the intersections. In COCON, schedules are predominantly produced manually. It is possible to manage an FTS of a small intersection as the different combinations are limited. However, for large intersections, especially in networks, it is difficult to produce good schedules manually. Once a certain order is chosen, it is rarely changed. This does result in the fact that many possibilities, and thus possibly better ones, are not considered. FTT always produces a near-optimal schedule or tells you whether no solution is possible, subject to the constraints (conflicts and clearance times) given. This makes it very interesting to use, but especially valuable for large intersections.

There are, however, some drawbacks to the current version of the tool. While comparing the two methods, several design choices can be traced back to the designer, of which a few cannot be taken into account with FTT. Firstly, preserving the order of green phases is important for the users of such tools. A way of taking this into account, is by giving the signal groups in the unwanted order, higher clearance times than the desired order. E.g., signal groups 3, 6, and 8 are all conflicting and can be scheduled in that order (3-6-8), or as 3-8-6. Note that due to the periodicity, all other combinations are included in the two given sets. By artificially making $\sigma_{3,8}$ higher than $\sigma_{3,6}$, a preference will arise for 3-6-8. Unfortunately, this method is devious and makes the constraints ambiguous due to their double use. Secondly, for nearby junctions, the users of the tool are used to visually fix times between the start of two green phases of succeeding signal groups. Again this can be taken into account using the clearance time matrix, which is explained using Figure 3.1.

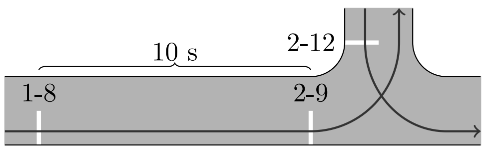


Figure 3.1: Example showing the addition of a fictive conflict.

The original matrix sets a lower bound on the time between start red, and beginning of green of another, conflicting signal group. For intersection 2 in this example, the clearance time matrix could be

$$\Sigma_2 = \begin{bmatrix} \cdot & 3 \\ 2 & \cdot \end{bmatrix}, \tag{3.1}$$

for signal groups {2-9, 2-12}. To induce a green wave, a fictive conflict can be added between 1-8 and 2-12, taking into account the travel time of 1-8. This would result in a possible combined clearance time matrix

$$\Sigma_{12} = \begin{bmatrix} \cdot & & 13 \\ & \cdot & 3 \\ -8 & 2 & \cdot \end{bmatrix}, \text{or} \begin{bmatrix} \cdot & 13 \\ -8 & \cdot \end{bmatrix}, \tag{3.2}$$

for signal groups {1-8, 2-9, 2-12} or {1-8, 2-12}, respectively. Note that 2-9 has become obsolete, because in this case, vehicles only arrive from 1-8, whose conflict is taken into account. For larger intersections, multiple constraints can be added, and if a schedule is feasible including these extra constraints, the green times will be synchronised, and allow for a continuous flow. This method is applied and explained in the case study of Appendix A.1.4. Unfortunately, this method is devious as well, and adds even more ambiguity to the matrix due to the added conflicts that are not actually present at the junction.

It can be concluded that it would be beneficial, for designing purposes, to extend the tool with sequence and follow-up requirements. This would make it possible to test multiple cases, with and without flows, and different sequences. The obtained schedules can be compared mutually and with the near optimal one (disregarding order and flow constraints), and the impact of the choices can be understood. Ultimately, the chosen option can be supported or rejected better than by common sense alone.

One last aspect to be noted, is that for the optimisation of the FTS, traffic arrivals are assumed to be constant. This is a well founded assumption when making a periodic schedule, as its performance is based on the average arrivals during a period. However, when looking at both networks of intersections and vehicle dependant control, the assumption fails to hold. The latter is discussed in Chapter 5, and a different approach regarding a network of intersections is explained in Chapter 4.

Chapter 4

Networks of Intersections

This chapter covers the research conducted on the combined performance of a network of intersections. First of all, it is important to notice that neighbouring intersections influence each other. The arrival of vehicles depend on the departures at the preceding intersection, and will more likely arrive in platoons, than uniformly distributed over a period of time. Secondly, the cases considered in this chapter are controlled with FTSs, for simplicity as well as better understanding of the underlying dynamics of a network. However, the FTSs used, are obtained under the assumption that traffic arrives with a constant arrival rate, which does not hold in a network setting, as mentioned in the first point. This platoon effect affects both the queue length build-up as well as the waiting times of the vehicles in the network. The difference between these values for the constant and the platoon arrivals are influenced by the moment at which the platoons arrive at an intersection. If the arrivals take place during a green period, the queues will generally be shorter than when they take place during red. By changing the relative starting time of the schedules, the arrivals of the platoons can be influenced, and the queue lengths of these signal groups can be minimised, without changing the FTSs. Lastly, a distinction is made between different types of signal groups, based on their location in the network. Those at which vehicles arrive when first entering the network, are called *external*, and all subsequent signal groups within the network are called *internal*.

This chapter starts by introducing the model used to describe traffic departure and queue lengths (the number of Personal Car Equivalent (PCE) behind the stopping line) for the external arrivals with one green phase in Section 4.1. Then, Section 4.2 explains how subsequent signal groups interact with each other. In Section 4.3, the model is extended to describe departure rates and queue lengths when considering multiple green phases and piecewise constant arrivals, typical for internal signal groups. Then, Section 4.4 elaborates on how the best relative starting times of the schedules are chosen. Finally, the results are validated using micro simulations in Section 4.5, which is followed by the discussion of the optimisation results and how the model can still be improved in Section 4.6.

4.1 Modelling Rate Profiles for External Arrivals

As stated in the previous paragraph, the arrivals at one intersection depend on the departures at the preceding junction. To explain how an expression for the departure rate and the subsequent arrival rate at a neighbouring junction are obtained, Figure 4.1 is used.

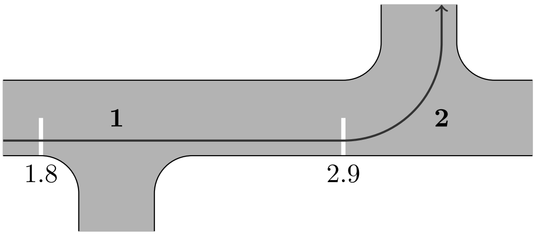


Figure 4.1: Scenario of a network, consisting of two three-way intersections, including one interconnecting traffic flow and the two signal groups it passes.

Each signal group can be described using three so-called rate profiles: the arrival rate $A_i(t)$, the maximal processing rate $P_i(t)$, and the departure rate $D_i(t)$, all defined on $[0, T)$. In this report, only piecewise constant rate profiles are considered for simplicity reasons. Figure 4.1 shows a network of two neighbouring intersections, numbered 1 and 2. First, let us consider one traffic flow, represented by the arrow. All external arrivals, including $A_{1.8}$, are considered to be constant. The maximal processing rates are not constant, but dependant on the FTS, which is explained using Figure 4.2. Moreover, this section only considers single green phases, which is extended to a more general definition for piecewise constant rate profiles, in Section 4.3.

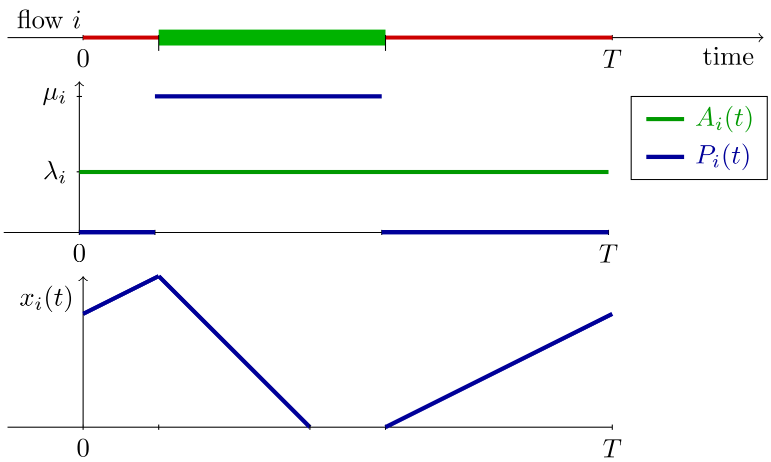


Figure 4.2: Schematic representation of the effective phase diagram, arrival- and processing rates ($A_i(t)$ and $D_i(t)$), and the queue length $x_i(t)$ of signal group i for $t \in [0, T)$.

Figure 4.2, shows an effective phase diagram of an arbitrary traffic flow, i, in which its effective green time is depicted. The maximal processing rate of this signal group is μ_i, during that time, and zero during the rest of the cycle. As assumed, its arrival rate is constant during the entire period and equal to λ_i. Throughout this report, a fluid model as introduced in [11] is used to describe the queue content, in which the traffic behaves as a fluid flow. This means the rates are deterministic and (piecewise) continuous. Using this model, and assuming the capacity constraint in (2.5) is met, the green time is enough to process all accumulated vehicles during a period. Consequently, the queue is empty at the start of the effective red period. During this red period, the queue length increases with the constant arrival rate λ_i, until the signal group turns green again. At this point, the queue will decrease in length with a rate equal to $\lambda_i - \mu_i$ until the queue is empty again. Knowing the queue length evolution of a signal group, allows the departure rate to be defined, as shown in Figure 4.3. The departure rate $D_i(t)$ can then be defined to be zero during effective red, and positive during the green phase. As long as there is a queue, vehicles depart at a rate equal to the maximal processing rate, but once empty, the so called *slow mode* starts, in which vehicles depart at the same rate as they arrive.

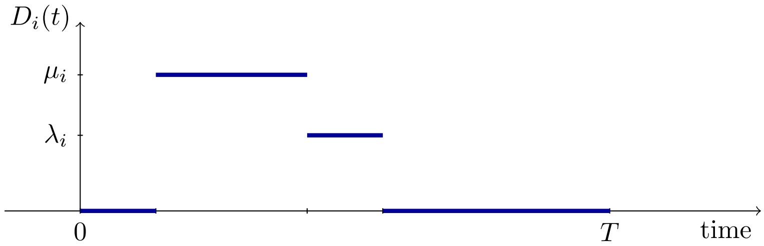


Figure 4.3: Simplified model of output rate $D_i(t)$ of signal group i for $t \in [0, T)$.

4.2 Rate Profiles of Subsequent Signal Groups

Using these models for the rate profiles of the different stages of an intersection, expressions can be derived for the arrivals at the succeeding signal groups. To this end, expression (4.1) is derived for the delay, Δ_{pq}, the traffic experiences between the two intersections, using the common distance d_{pq} between intersection p and q, as depicted in Figure 4.4, and the average speed on that part of the road $\overline{v}_{pq}$. Note that using one common delay for each vehicle, results in a model of the queue as one without a length, i.e., all traffic accumulates on one point, at the stopping line. Also, in general $d_{pq} = d_{qp}$.

$$\Delta_{pq} = \frac{d_{pq}}{\overline{v}_{pq}}. \tag{4.1}$$

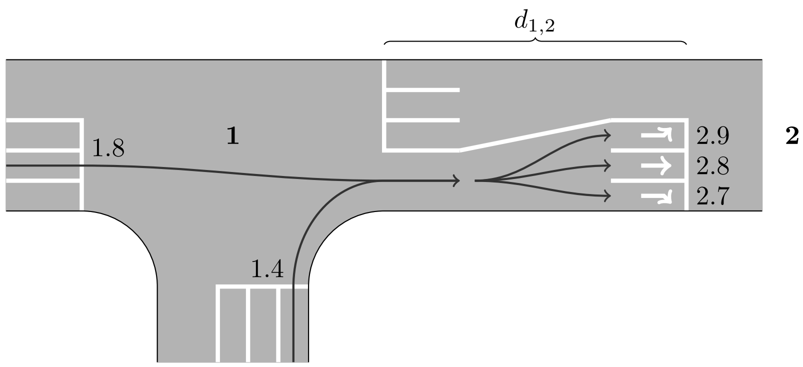


Figure 4.4: Examples of the communicating signal groups within a network of two junctions.

Figure 4.4 shows an extension of the scenario in Figure 4.1, including a representation of the communicating signal groups. With the two departing flows from intersection 1 and the three entering 2, a total of six origin-destination combinations are formed. We then define fraction of vehicles from an origin signal group i to a destination one j as f_{ij}. Moreover, j is called a *successor* of i, and i a *predecessor* of j if $f_{ij} > 0$, and when no vehicles travel from i to j, $f_{ij} = 0$. Using these fractions and the common delay, a partial arrival rate $A_{ij}(t)$ for each origin-destination(i-j) can be composed as

$$A_{ij}(t) = f_{ij} \cdot D_i \left(\mathrm{mod}(t - \Delta, T)\right), \tag{4.2}$$

defined for $t \in [0, T)$, which is also shown in Figure 4.5. Note that $D_i \left(\mathrm{mod}(t - \Delta, T)\right)$ is the departure profile of signal group i shifted in time.

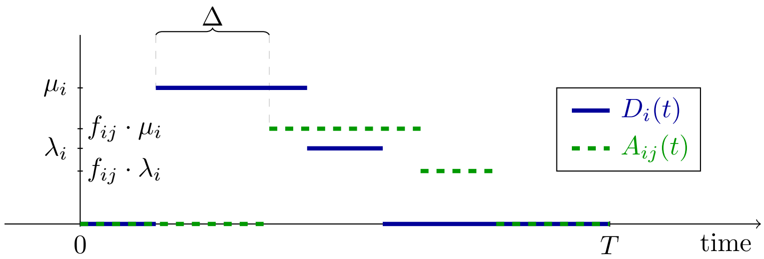


Figure 4.5: A schematic representation of the relation between the departures and the arrivals at subsequent junctions.

By superposing all partial arrival rate profiles of the same destination, the complete arrival profile $A_j(t)$ can be derived as

$$A_j(t) = \sum_i A_{ij}(t), \tag{4.3}$$

for $t \in [0, T)$ and i preceding j. An example of a complete arrival rate, including a second predecessor (h), is shown in Figure 4.6. Note that the multiple departing platoons at the origin junction cannot arrive simultaneously at their destination. These signal groups are conflicting, and therefore "processed" sequentially, resulting in non-overlapping green times.

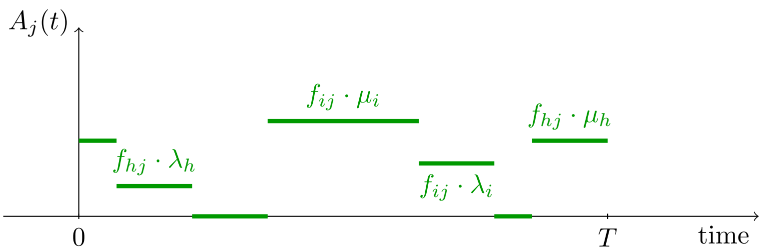


Figure 4.6: Schematic representation of the modelled arrivals at a receiving signal group.

The *internal* arrival rate profiles can now be defined, but are not constant. To evaluate the queue lengths and the departure rates of these signal groups, another approach is needed than the one explained in Section 4.1. This is done in the next section.

4.3 Modelling Piecewise Constant Rate Profiles

For an FTS to be stable, it is required that the queue needs to be emptied at least once within each cycle. For a schedule with only one green phase and constant arrivals, this means the queue is empty at the start of the red phase. A more general approach is defined in this section, evaluating the queue length evolution of one signal group during one cycle, subject to piecewise constant arrivals and processing rates. An example of such rates is depicted in Figure 4.7.

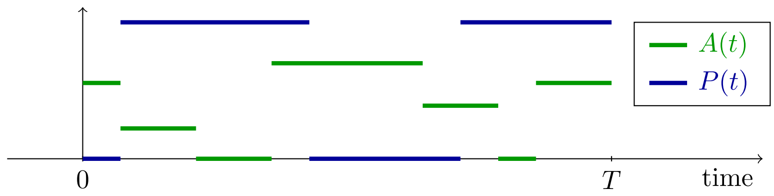


Figure 4.7: Example of piecewise constant arrival and processing rates.

Note that $P(t)$ can only adopt a value of 0 or μ, the maximal processing rate. Given the piecewise constant arrival and processing rates, $A(t)$ and $P(t)$ respectively, the periodic maximal possible derivative of the queue length can be defined as

$$x'_{max}(t) = A(t) - P(t), \tag{4.4}$$

which is piecewise constant and periodic as well. Let $x'_{max}(t)$ have N breakpoints, at which its value changes. Let x_0 be the queue length at $t = 0$, x_i be that at the i^{th} breakpoint of $x'_{max}(t)$ and x_{N+1} be that at $t = T$. Note that $x_{N+1} = x_0$. Using similar concepts as explained using Figure 4.2, the queue length at each breakpoint can be computed:

$$x_{i+1} = \max\{0, x_i + x'_{max,i} \cdot t_i\}, \tag{4.5}$$

with $x'_{max,i}$ the value of $x'_{max}(t)$ between breakpoint i and $i+1$, and t_i the time difference between these points. Note that the time of breakpoint i is defined as $t(i) = \sum_{j=0}^{i-1} t_j$.

Because $x_0 \geq 0$, and $x(t)$ becomes 0 somewhere within the cycle, choosing a starting value $x_0 = 0$, and computing x_{N+1} produces the correct value for x_0. This means that obtaining the correct queue length evolution, and subsequently the departure rate, requires the queue length to be computed twice in worst case. Whenever the queue is emptied ($x_i > 0$ and $x_{i+1} = 0$), an extra breakpoint needs to be added within that interval, coinciding with the exact moment the queue is emptied: $\sum_{j=0}^{i-1} t_j + x_i/|x'_{max,i}|$. This extra breakpoint indicates the start of the slow mode in the departure rate: $D(t)$ equals the arrival rate while the queue is empty, and the processing rate at any other time.

To compute all queues within a network, the corresponding arrival and processing rates are needed. The latter are obtained directly from the FTS of the corresponding intersection. However, the arrival rates are partly obtained after the departures are known for the preceding signal groups. Therefore, the queue lengths, and subsequently the departure rates, are computed using the steps explained in the next paragraph.

Divide all the signal groups of the network into two categories: the ones without any predecessors go into the *can be computed* category, and; the ones with predecessors go into the *cannot yet be computed* category. Then, while the first category contains signal groups, choose one i and compute its queue length $x_i(t)$, and its departure rate $D_i(t)$ according to the method explained in Section 4.3. Subsequently, if signal group i has any successors, compute their arrival rate according to the method explained in Section 4.2. For each of the successors, check whether all its predecessors have been computed. If so, move it to category *can be computed*. Lastly, remove i from that category.

Note, that by following these steps, a network in which circular flows exist, i.e., the arrivals at one or more signal group depends on its own departure, cannot be computed. These signal groups would never be moved to the *can be computed* category. To take these networks into account, the framework needs to be extended, possibly, by taking into account more detailed routing information. I.e., by considering locations where vehicles enter the network and where they exit it, and even take into account the routes they take, in case of multiple possibilities.

4.4 Minimising Queue Lengths

Using the steps explained in Section 4.3, all queue lengths in the network can be computed at any $t \in [0, T)$. This section considers an approach to minimise the average queue length within one period, subject to fixed FTSs, and varying their relative starting time. The method is explained using a simplified case of the scenario from Figure 4.1, the *Kumar-Seidman problem* [6, 7]. This scenario is depicted in Figure 4.8, and the used data can be found in Appendix A.2. In this example, amber periods, as well as lost green times and used amber times, are disregarded as the concept to be explained is based on the effective times.

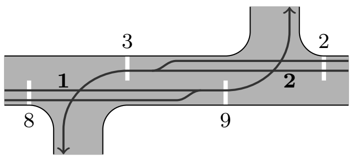


Figure 4.8: The Kumar-Seidman problem.

Using the given information, FTT produces the phase diagrams shown in Figure 4.9. With $d_{1,2} = 200$m, being large enough to allow all arriving traffic during one period to queue, and $\overline{v}_{1,2} = 50$km/h, $\Delta_{1,2} = 15$s. Moreover, using $f_{\text{1-8,2-9}} = f_{\text{2-2,1-3}} = 1$, results in the assumed arrivals at these signal groups shown in Figure 4.9 with the blue and orange dotted lines. Below the FTSs in Figure 4.9, the fluid model queue length evolution is shown, computed according to the steps explained in Section 4.3. Note that, because of the merging of the lanes, the maximal processing rate of signal group 1-3 is lower than that of 2-2. This results in an increasing queue length, even though the platoon arrives during an effective green period. Nevertheless, the arrivals during effective red, cause the queue length to increase more. Most advantageous and pleasant for the road user would be for all the arrivals to arrive during green, which is called a *green wave*. By changing the relative starting time of both FTSs, the schedules can be “synchronised” to result in a shorter average queue length. The difference in starting times of the schedule of intersection q relative to the starting

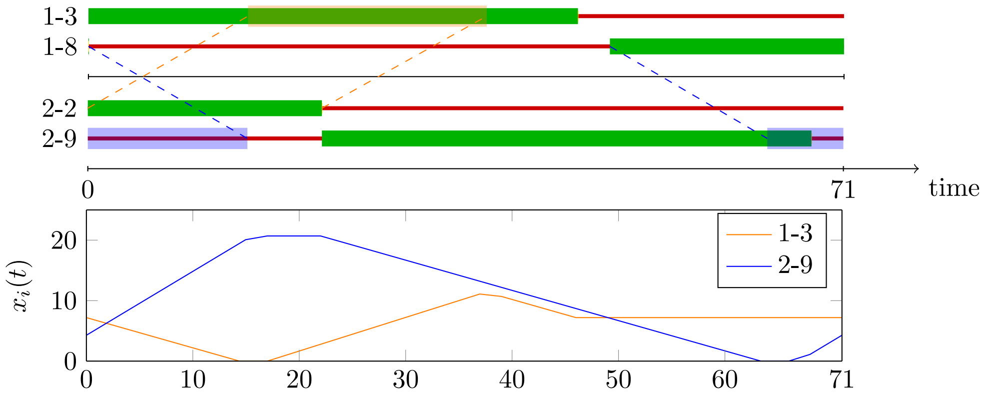
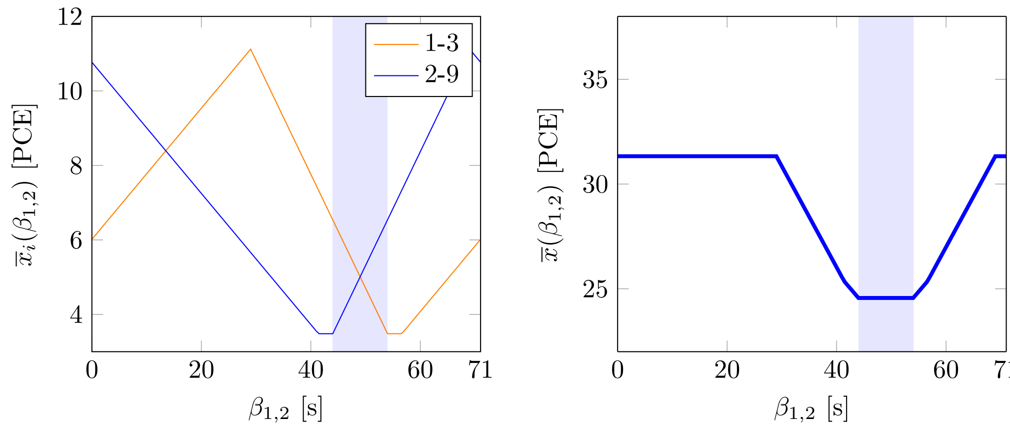


Figure 4.9: Phase diagrams of the two intersections of the Kumar-Seidman instance, showing the platoon arrivals and corresponding fluid model queue lengths at their follow-up lanes.

time of the one of intersection p, is called the phase shift β_{pq}. By varying $\beta_{1,2}$ from 0 to the period time, T, and executing the steps explained in Section 4.3 for each phase shift, Figure 4.10 can be produced, showing the average queue length of signal groups over one period, computed using

$$\overline{x}_i = \frac{1}{T}\int_0^T x_i(t)dt. \tag{4.6}$$



(a) (b)

Figure 4.10: Average queue lengths (over one period) as function of the phase shift, $\beta_{1,2}$, at intersection 2: a shows individual average queues of receiving signal groups, and; b shows the total average queue length of the entire system (all four signal groups).

In Figure 4.10a, only the internal signal groups are shown, as the external ones are assumed to have constant arrivals, resulting in a constant average queue length. In general, the overall best performance is desired, and therefore the total average queue length of the system (Figure 4.10b) is minimised, resulting in a desired phase shift of $44 \leq \beta_{1,2} \leq 54$.

This range of "optimal" values gives room to choose what is best in the specific case. Typically, the signal group with the highest arrival rate would be favoured. However, in this case with two identical queues, a phase shift exactly in the middle of the range would be chosen: $\beta_{1,2} = 49$, which would result in the updated phase diagrams of Figure 4.11.

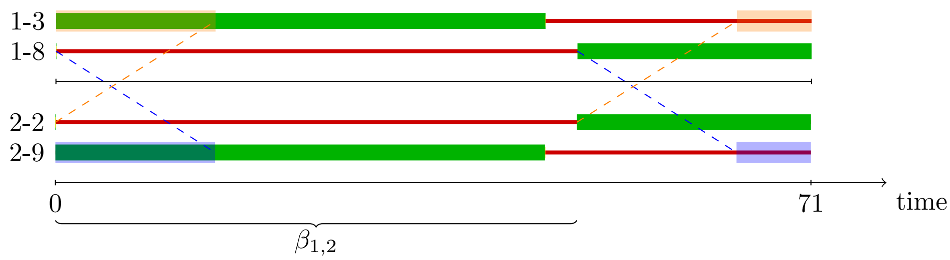


Figure 4.11: Phase diagrams of the two intersections of the Kumar-Seidman instance, with a phase shift of $\beta_{1,2} = 49$, including the platoon arrivals at their follow-up lanes.

4.5 Comparison with microsimulations

Now that the best phase shift is found for the two intersections, the question arises whether or not this approach is satisfactory. To this end, microsimulations using the PTV VISSIM software have been carried out. The intersections are modelled in the same way as described in Appendix A.2, using the FTSs from Figure 4.9. Fifteen simulations of each 1 simulation hour, were executed for each phase shift, after which the average queue length is saved for comparison. The resulting values are plotted in Figure 4.12.

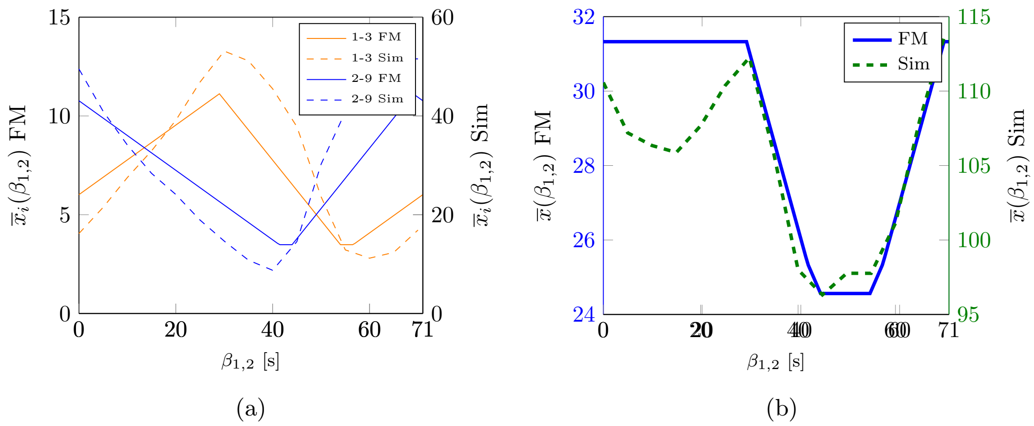
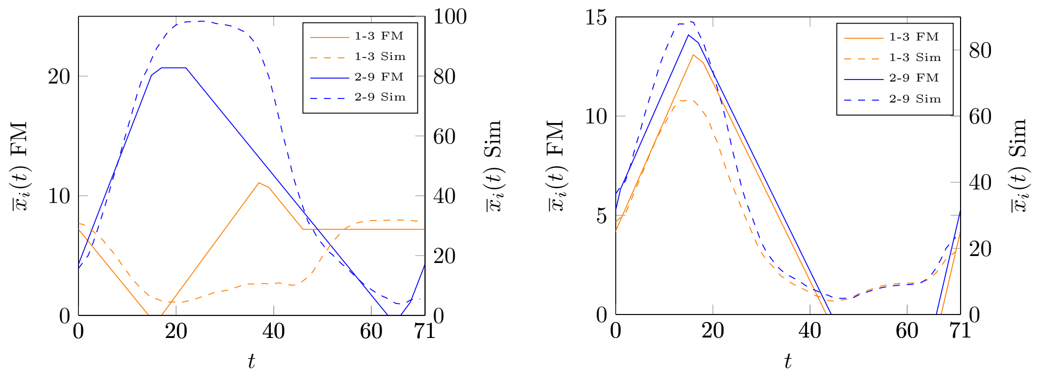


Figure 4.12: Comparison of fluid model (FM) and microsimulations (Sim) of the individual average queue lengths (a) and the total average queue length (b) as function of $\beta_{1,2}$, at intersection 2.

There are two major differences to be observed: Firstly, the amount of vehicles in the queues are much higher according to the simulations, and; secondly, the shape of the lines is slightly different, resulting in a second, but local, minimum around $\beta_{1,2} = 15$s. Nevertheless, the overall minimal value is at the same place, which is the aim of the abstraction.

Both observed differences take roots in the queue length evolution at specific phase shifts. Therefore, Figure 4.13a and 4.13b show the comparison of these for shifts of 0 (as explained in Figure 4.9) and 50 seconds (which is the closest to the optimal phase shift that has been simulated), respectively.

The difference in queue length is caused by several factors. The first one is the lack of including stochastic behaviour in the fluid model. The assumption that the queue should be empty once every period does not actually hold for the average stochastic periodic behaviour. Especially in this case, considering the high load on the system, the probability of the queue *overflowing* (having a queue length > 0 at the point it was assumed empty) increases, which results in a higher equilibrium position. The second factor, is the way the queue length is defined. In PTV VISSIM, a vehicle is considered to be queueing when its speed drops below 5 km/h and the distance from its headlight to its predecessor's one (called *headway*) is less than 20 m. In the abstracted fluid model, the queue is modelled as a point-queue, where only the vehicles that have arrived at the stopping line are counted. Assuming the average headway in a static queue is 6m, signal group 2-9's queue at $t = 17$, would be between $\frac{20 \cdot 6}{2} = 60$ m and $\frac{20 \cdot 20}{2} = 200$ m. This means that vehicles at the end of this interval would be queueing much earlier than assumed. Both discussed factors are of much influence to the shape of the queue evolution, however they do not change the location of the minimum, as the overall shape is similar and the average values scale accordingly.

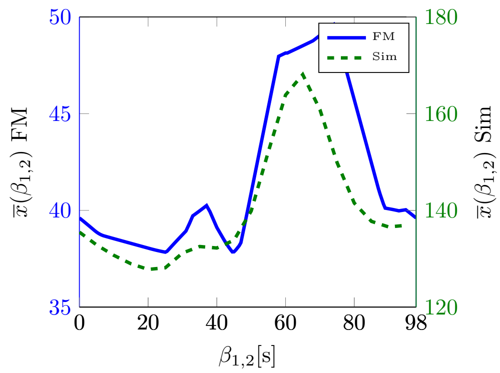


(a) $\beta_{1,2} = 0$s (b) $\beta_{1,2} = 50$s

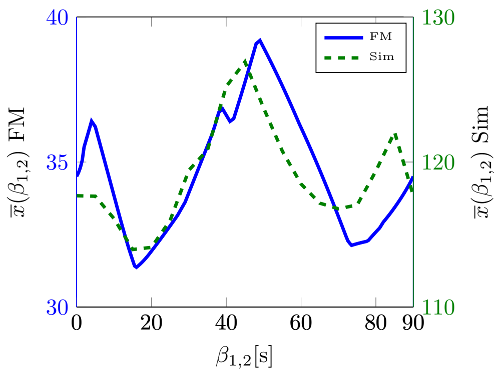
Figure 4.13: Comparison of fluid model (FM) and microsimulations (Sim) of the queue length evolution during one period with: a a phase shift of 0s, and; b a phase shift of 50s.

An interesting difference can be observed in Figure 4.13a, where the queue evolution of 1-3 does not match at all. This is caused by two underlying phenomena. Firstly, the fluid model assumes that the capacity of the successive signal group is approximately half the arrival rate, due to the higher capacity of the preceding signal group. This means that more vehicles arrive than can be processed, and the queue length has a net growth. In reality however, the vehicles have enough space in between them to interweave, and pass with a slight over capacity. Also the vehicles reduce their speeds and spread out over a longer period of time, resulting in the fast increase of queue length only after the signal has turned red. Secondly, it is assumed that the platoon arrivals arrive evenly spread at the second intersection as they departed from the first. According to [4], this assumption does not hold, and the platoon diffuses over time. Both phenomena result in a spreading of the arrivals.

A second case, intersection "De Uithof", as described in Appendix A.3, is also used for a comparison. The results are shown in Figure 4.14



(a) MR



(b) ER

Figure 4.14: Comparison of fluid model (FM) and microsimulations (Sim) of the average queue lengths (over one period) as function of the phase shift, $\beta_{1,2}$, at intersection 2: a Morning rush hour, and; b Evening rush hour.

The same conclusion as for the Kumar-Seidman case can be drawn; although the model does not match the simulations exactly, the aim of the model–to obtain an optimal phase shift–is achieved.

4.6 Drawbacks and improvements

As concluded in the previous section, the aim of the model is achieved. Nevertheless, several improvements are possible. The first improvement can be made to the queueing model. The current (fluid) model, does not represent the queue length evolution as it is in reality. To show this difference, the queue evolution of signal group 2-2 of the Kumar-Seidman case is shown in Figure 4.15.

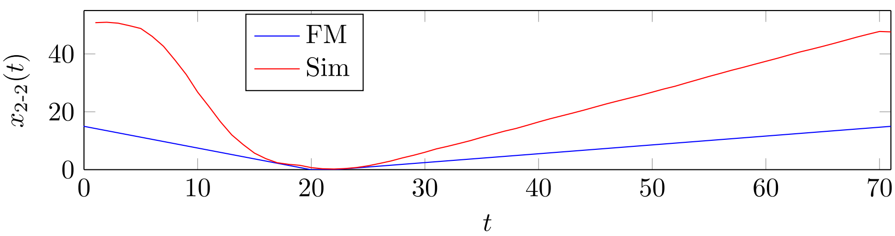


Figure 4.15: Queue length evolution comparison between the fluid model (FM) and the microsimulations (Sim) for signal group 2-2 on $t \in [0, T)$.

The parameters chosen are the same as given in Appendix A.2, and the simulation results are obtained by averaging the queue content for each period in 15 simulation runs of one hour each. It can be noted that the queue is much longer according to the simulation, than what the fluid model predicts. As described in previous sections, this is caused by the lack of stochastic behaviour. There are, however, several studies that have been conducted on queueing models [4, 1, 9], which attempt to better predict the average queue length. Even though these do not tell anything about the specific evolution of the queue, the concepts can be used to make a better fit. The first concept is the overflow queue, briefly introduced in the previous section. This so-called overflow queue is defined in [9] as "the mean queue length at the end of a green period". By using this extra queue, the entire queue length evolution shifts upwards.

One attempt to make a better fit, is to change the linear decrease in the queue into an exponential one. Assuming the start of the green phase is at $t = 0$, an expression for this decrease defined in $t \in [0, g]$ could be

$$x(t) = \alpha \cdot e^{-\beta t} + \gamma, \tag{4.7}$$

in which α, β, and γ are constants that need to be obtained by substituting constraints. The most obvious being the overflow queue

$$\gamma = \lambda \cdot \frac{\rho^2}{2 \cdot (1-\rho)}, \tag{4.8}$$

in this case based on an $M/D/1$ queue, in which λ is the mean arrival rate, and $\rho = \lambda/\mu$ with μ the maximal processing rate. The two other constants can be obtained by substituting

$$x'(0) = \lambda - \mu, \tag{4.9}$$

and

$$x(0) - x(g) = \lambda \cdot (T - g), \tag{4.10}$$

into (4.7). Both constraints representing that the derivative at the start of the green phase should equal the original decrease rate, and that the total decrease should equal the increase during the red phase, respectively. Unfortunately, this approach does not fit the queue evolution at all. The next step would be to extend (4.7) with a second exponent, as in

$$x(t) = \alpha \cdot e^{-\beta t} + \delta \cdot e^{-\epsilon t} + \gamma, \tag{4.11}$$

and an additional two constraints, to substitute for δ and ϵ, which could be

$$x''(t) > 0, \tag{4.12}$$

for $t \in [0, g]$, and

$$\int_0^g x(t)dt + \frac{1}{2} \cdot (T-g)^2 \cdot \lambda = T \cdot E[X]_{known}, \tag{4.13}$$

with $E[X]_{known}$ the expected average queue length according to any accepted prediction method, e.g., [1] or [9].

Unfortunately, this extra step towards a better fit, has not been solved in the course of this project. This could be an interesting option for further research.

The second point of improvement, is the queue increase during the red phase. Firstly, the fluid model assumes a point-queue, having no length. By making the delay Δ depending on the number of vehicles, the queue will increase faster and become a better match to the simulation results. Secondly, the phenomenon of lanes merging and dividing, is not taken into account. This event can be observed especially well in the case of the Kumar-Seidman problem, in which all vehicles are merging.

Thirdly, the method proposed in this chapter, considers fixed schedules and attempts to optimally use these within a network setting. The aspect being considered in this method to have most effect, is the platoon arrivals. A better way to consider these, might be to add a constraint between the connected green phases, forcing a flow between these signal groups to happen. This is similar to the recommendation done in Chapter 3, about the follow-up requirements to be considered in FTT.

Lastly, the obtained "synchronised" schedule, cannot be guaranteed optimal. One of the reasons being that the longest FTS is chosen as fixed, and the shorter one is "streched". The best solution would be to consider the network as a whole in the optimisation procedure in FTT, and take the platoon departures and arrivals into account, thereby obtaining an FTS for the combined network as a whole. Two other options arise while trying to make the obtained solutions better. One would be to check different ratios of period times. I.e., if the longest period is twice as long as the shorter one, it might be better to fit the shorter FTS twice in the longer one. The second option would be to check whether the addition of green phases to the preceding and succeeding signal groups is beneficial. By doing so, vehicles arrive or leave these queues in multiple smaller platoons, which can be of positive influence on the average queue lengths.

Chapter 5

Vehicle Actuated Control

As explained in Chapter 3, a Fixed-Time Schedule (FTS) is rarely implemented to control an intersection. These schedules are mostly used to check the capacity of the intersection and see whether a feasible schedule is possible. Also, it is used as a base to construct a Vehicle Actuated Control (VAC) strategy, which is how the majority of the Dutch Traffic Control Installations (TCIs) are controlled, based on the current traffic demand, usually detected using *induction-loop traffic detectors*. These sensors, located within the road surface, detect metal objects passing them. Most control strategies used in the Netherlands, use the information from these loops to extend or skip a green phase, to fit the current demand.

Although multiple control strategies exist, and different organisations, municipalities, as well as the department of waterways and public works, desire and use different ones, most of the strategies use common design parameters as a base for the eventual control implementation. First of all, most of them work with a maximal green time, imposing a constraint on the duration of the green phase. Secondly, a block structure is often used, which imposes a constraint on the order in which signal groups receive their green phases. Both elements are often inspired by an FTS.

Obviously, it is tedious and long work to produce good schedules for larger intersections. More so, if they should be vehicle dependant. Routine work at Grontmij Nederland B.V., consists of modelling parts of cities or important (commuting) arteries, considering different scenarios, and subsequently simulated. To this end, it is desired to have an easy option to obtain such strategies, while guaranteeing an acceptable performance. In this section, a foundation is made to produce VAC elements automatically, using an FTS.

5.1 Maximal Green Times

A VAC strategy aims to optimise throughput, and minimise waiting times by extending and reducing green phases according to the current traffic demand. To prevent that a very busy artery in an intersection remains green, a maximal green time is set. In a satisfactory FTS, the green phases are long enough to process the average arrivals during one period time, and the maximal green times can be based on those. A rule of thumb, among others, to determine the maximal green times is constructed as follows using the green time g_i from the FTS:

- $g_i^{\max} = g_i + g_i^*$
  - minimal 15s;
  - has a resolution of 5s;
- $g_i^* = 20\% \cdot g_i$
  - minimum of 10s, and a;
  - maximum of 20s.

This rule of thumb is intended to obtain the maximal green times for usage during the rush hours, and are slightly altered for usage in off-peak control. Using this strategy, it is possible to automatically obtain the times, necessary for most VAC.

5.2 Block Structure

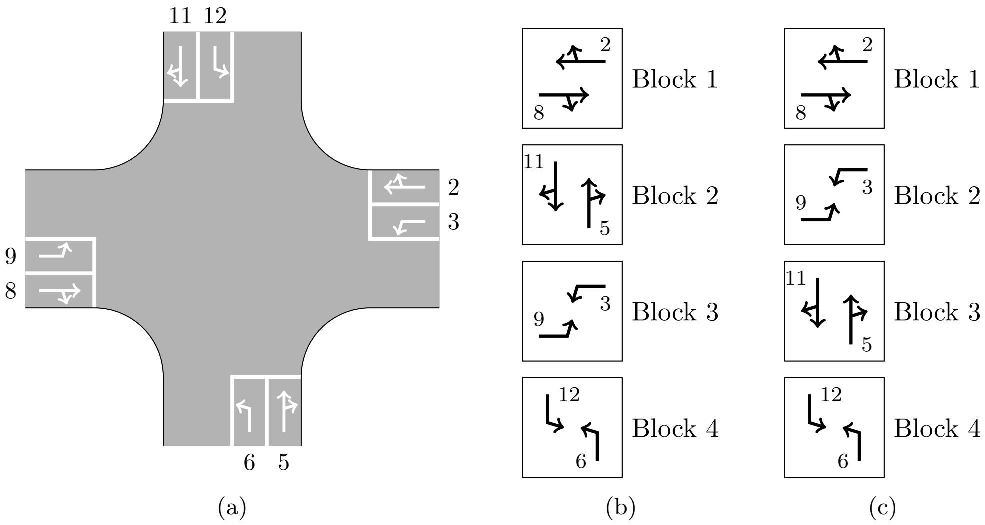


Figure 5.1: Example to illustrate the block structures. a shows the basic scenario, and b and c show two possible block structures for this example.

Another important aspect for most VAC strategies, is a block structure. This structure imposes the overall order in which the signal groups are given a green phase and an example of such a structure is shown in Figure 5.1. Each "block" consists of several non-conflicting signal groups, of which it is desired that their green phases overlap. Simply put, the basis of such strategies is to subsequently allow the clustered signal groups their green phases simultaneously. However, each signal group has a different load, which makes this basic control far from optimal, if the green phases are completely synchronised. Therefore, a VAC generally takes into account the arrivals, and vehicles "request" green phases by driving over an induction-loop. The strategy can extend or shorten the individual green phases to allow conflicting signal groups to receive a shortened or extended realisation, respectively.

Considering a single signal group within the control sequence, several scenarios can occur:

1. Its minimal green time has elapsed, and no vehicles are measured any more. In this case, the red phase can be reactivated to allow a busy signal group from the next block that has no conflicts with the rest of the current block, to have an *early realisation*.
2. Its minimal green time has elapsed, no vehicles are measured any more, and all other signal groups in its block coincide. This initiates the following block.
3. Its minimal green time has elapsed, there are still vehicles in its queue, but all other signal groups have emptied theirs. If the concerned signal group has any conflicts with one or more of the ones in the following block, it can receive an extended green phase, or *extended realisation*, but simultaneously the non-conflict ones from the following block can get an early realisation.
4. Considering the same scenario as in 3., but the concerned signal group does not have conflicts with the ones of the following block. In this case, its green period can also be extended until it reaches its maximal green time, but the following block can be initiated simultaneously. Note, that some signal groups from that block can have had an early realisation as well.

5.2.1 Flexibility Index

Looking back at the example in Figure 5.1, it can be observed that the block structure shown in Figure 5.1c allows more early and extended realisations than that of Figure 5.1b, as there are less conflicts between consecutive blocks. E.g., signal group 3 can receive an early realisation when 8 has very few arrivals, and similar interaction exists between signal groups 9 and 2, 6 and 11, and 12 and 5. Such a block structure is therefore flexible, and is preferred over the first one. To be able to make a comparison between different block structures, first $K(i)$ is defined as the number of conflicts between block i and its successor $i+1$. This value can be weighed using the average load of conflicting signal groups of consecutive blocks, using

$$W(i) = \frac{1}{K(i)} \sum_{j,k \in \text{conflicts}(i,i+1)} \frac{1}{2}(\rho_j + \rho_k). \tag{5.1}$$

Note that $0 \leq W(i) \leq 1$, and that it is higher when two consecutive blocks have heavily crowded, conflicting signal groups. Obviously, a block structure can be constructed with many blocks, each consisting of very few signal groups such that the consecutive blocks have low weighted conflicts. To this end, the total number of blocks M also needs to be taken into account, and is preferably low. For the comparison of multiple block structures of the same intersection, a Flexibility Index (FI) is introduced as

$$\text{FI} = \alpha \cdot \left[1 - \frac{1}{M} \sum_{i \in \text{blocks}} W(i)\right] + (1-\alpha) \cdot \frac{2}{M}, \tag{5.2}$$

with an arbitrary $0 \leq \alpha \leq 1$ representing the relative importance of the weighted conflicts with respect to the amount of blocks. This means that $0 \leq \text{FI} \leq 1$ as well, and that FI is higher for a more flexible block structure. Unless otherwise stated, $\alpha = 0.5$ is used.

5.2.2 Generating Block Structures

To obtain a block structure from an FTS, a heuristic is devised based on two main concepts. The first one being that a signal group can receive extended realisations within an indefinite amount of subsequent blocks, and the second being that it can receive an early realisation within an indefinite amount of preceding blocks. As long as that signal group has no conflicts in the intermediate blocks and its maximal green time is not exceeded, this assumption is well-founded. Let us consider the example of Chapter 2, which is again shown in Figure 5.2.

Figure 5.2: The FTS for the three-way junction used as an example in Chapter 2.

For both concepts, the first step is to obtain the starting times of each green phase, to find their order within the FTS. In this case the order is

$$\{3,4\},\{6\},\{7\},\{8\},\{2\}. \tag{5.3}$$

Let us consider the first concept. Assuming the VAC can extend the green phases indefinitely into the subsequent blocks, it would be most beneficial to place the signal group as early as possible in the block structure, so that it has as much room as possible to be extended. The signal group can only be pushed into an earlier block until it has a conflict with one in its preceding block. By checking whether each signal group in the given order, can be put in a block together with its predecessor(s), and removing the empty blocks, the block structure evolves as follows:

$$\begin{aligned}
&\{4\},\{6\},\{7\},\{8\},\{2,3\};\\
&\{6\},\{7\},\{8\},\{2,3,4\};\\
&\{6,7\},\{8\},\{2,3,4\};\\
&\{6,7\},\{2,8\},\{3,4\}.
\end{aligned} \tag{5.4}$$

In this case, the block structure cannot change any more, and this is its final form, with an FI $= 0.75901$. In some cases, checking each signal group once is not enough due to the change in composition of the blocks. Therefore, the signal groups are checked iteratively until the composition of the blocks does not change. Unfortunately, there are scenarios causing an endless cycle of changing block structures. The most simple example to show this is the so-called C5 problem (5 conflicts in a cycle), which consists of signal groups 2, 4, 6, 21 and 23, in which each signal group has conflicts with two adjacent ones in the sequence or cycle {2,6,23,4,21}. Fortunately, there is repetition over a finite number of possible structures, and the stopping criterion can be extended, considering the previously found structures.

Checking the FI of each of the obtained block structures within the repetition cycle, and choosing the one with the highest flexibility results in the desired block structure.

Obviously, the same approach can be used for the second concept. However, in that case, signal groups can be pushed into a later block until it has a conflict with one in its successive block. This results in the block structure

$$\{2,3\},\{4,6\},\{7,8\}, \tag{5.5}$$

which has an FI of 0.75901, equal to the one of the first approach. In this case, both concepts result in the same flexibility, however, in most cases one of the two is better, but none of the two methods has a guaranteed better result. Therefore, it is advised to check both methods to find the best for each intersection.

A short and interesting comparison is done on the case study of Appendix A.1.4, the *Hertekop*, in Appendix B. Because this case is actually a network of intersections, the proposed method of obtaining block structures does not result in a desired one. In this case, signal groups with predecessors or successors are sometimes put in the same block as their successors or predecessors, respectively, because they do not have any conflicts. Unfortunately, this would result in a lower performance, as by the time vehicles arrive at the successive signal group, the next block might already have been initiated, and the traffic flow therefore interrupted. As a consequence, the FI fails to representatively measure the systems performance. It is recommended to produce separate block structures for each individual intersection, and combine the obtained blocks in such a way to allow flows to be formed.

5.3 Further Research

Although generating maximal green times and block structures, allows an easy way to generate VAC, it is not yet substantially tested for these elements to be implemented into the control of intersections. Nevertheless, the method of obtaining these elements can facilitate the process of producing VAC strategies for large networks in prediction models. These models need to represent future scenarios, and therefore, many assumptions are made. Moreover, knowing the precise flows is not the aim of the models. Hence, decreasing the amount of manual work, by making the production of the necessary VAC strategies obsolete, is very much appreciated.

In the course of this project, a cooperation has been set up with a second graduation project, in which the generated elements of the VAC strategy have been used [5]. The results imply that the elements are performing well, and according to interviews with experts, the elements would only be chosen different as a consequence of certain standards or guidelines. This, of course, does not guarantee that the proposed method always generates well performing elements. To do so, a genuine comparison between the generated and the currently used elements should be carried out, consisting of simulations for both of them. Both manually choosing the parameters, as well as performing the simulations, are time consuming actions, and have been considered to be outside the scope of this project. Nevertheless, it could be an interesting research to conduct in the future. This will also show how the FI correlates to the actual performance of the intersection.

A complete different approach to VAC, would be to build new strategies and control algorithms. First, a stable algorithm is necessary, to guarantee processing of traffic with low waiting times, and to prevent backlash of congestions. After that, strategies implementing these control algorithms can be tested. During the course of my studies I have been introduced to supervisory control, which deserves to be looked into, to model and control this kind of dynamic system. Especially due to the modular nature of intersections, and networks of them, supervisory control might help to keep a clear overview of the problem, and easy addition of constraints between different elements of the systems. The last point being very hard to consider using the conventional methods.

Chapter 6

Conclusions and Recommendations

The goal set for this project is to study how Fixed-Time Schedules (FTSs) can be used in both networks of intersections and Vehicle Actuated Control (VAC) strategies. Moreover, an additional benefit of the cooperation with Grontmij Nederland B.V., is the possibility for the TU/e to improve the tool currently being developed. By comparing the performance of this tool, with that of the tool currently used, more insight has been gained in the way FTSs are used, and what aspects are desired for such a tool. First, the drawn conclusions are recapitulated, followed by a short summary of the recommendations for all parties involved in this project.

6.1 Conclusions

The first conclusion drawn during this thesis, is that the fixed-time optimisation method developed at the TU/e (FTT), performs at least as good as COCON, the tool currently used. Although the obtained FTSs are seldom used to control intersections as is, and the goal has not been to produce optimal schedules, a lot of time can be saved, as opposed to manually producing them. Nevertheless, drawbacks have been encountered while using FTT. Most of them, however, can be solved by implementing the recommendations intended for this tool, presented in the next section.

Then, considering networks of intersections, it is important to take platoon arrivals into account. The effect that platoons have on nearby intersections is not negligible, and is demonstrated using microsimulations. An optimisation method is proposed, to find the optimal relative timings of the FTSs, such to minimise the queue lengths within a network controlled by FTSs. It has to be noted, however, that this method fails to optimise each separate vehicle flow, but instead finds the average best solution. To improve the approach, several steps can be taken. Some improvement can be obtained by following the recommendations intended for FTT, but most of it, by following that for the network approach; both to be found in the next section.

Finally, for VAC, a lot of research still needs to be conducted. However, a method is proposed that generates both maximal green times, and a block structure for intersections.

This method can be used to facilitate the production of VAC strategies, especially within prediction simulation models. As these often cover extensive areas, and contain many intersections that do not require perfect control strategies, generating these elements can save a substantial amount of time. Also, the approach has been tested in cooperation with another graduation project, in which the generated elements have been used. The approach has been found to perform well, even though its performance has not been quantified.

6.2 Recommendations

As this project has explored several fields of traffic control, also the recommendations are split in several parts, starting with those for the method and tool being developed at the TU/e (FTT). While using FTT, several drawbacks have been encountered. The most obvious one is that multiple green phases are not taken into account yet. This is already being added, but at the time of writing it is still missing. Moreover, several other constraints are desired for FTT to be used as a design tool. Firstly, in general, the users are used to choose certain aspects of the schedule without considering the performance. Even though FTT returns near optimal schedules, certain choices are expected to be considered, e.g., a preferred order, or sequence of the green phases, as well as platoon arrivals at nearby intersections. Additions to the tool should at least consider both of them. An addition to the possibilities of COCON, would be to show the performance of the resulting FTS, and that of the near optimal solution, such to let the user weigh the importance of choosing certain constraints and to support the choice made.

Secondly, the method devised to consider networks of intersections achieves its aim, but allows for further improvements. To improve the queue length prediction, a model is proposed in Section 4.6 that brings the number of vehicles closer to reality. The model has not been finished, but can be considered in further research. A second improvement to the queue length model, consists of changing the way the build-up is modelled. Both the aspect that it is regarded as a point-queue and that it does not take into account capacity changes, can be improved. A third improvement would be to consider different constraints for the FTS, as this can result in better performance of the network setting. These constraints are explained in the recommendations for the tool. As a last improvement, it might be interesting to consider different ratios of period times, e.g., choosing to allow multiple fixed time cycles of one intersection to happen during one cycle of the other, or allowing multiple green periods for certain intersections within networks. This has not yet been looked into, mostly because the number of options is enormous, and an educated guess, which requires more research, is desired.

Thirdly and lastly, VAC has been looked into. Even though the generated elements have been used in a study, this does not guarantee overall good results. Further research can be conducted into the performance of the obtained VAC, comparing it with a manually produced one. Furthermore, additional strategies and control algorithms can be explored, one of them being supervisory control.

Bibliography

[1] R. Akçelik and Australian Road Research Board. *Traffic Signals: Capacity and Timing Analysis*. Australian Road Research Board, 1981.

[2] DTV Consultants. COCON. http://www.dtvconsultants.nl/Producten/-_COCON.aspx. Accessed: 20 Jul 2015.

[3] S. Fleuren. Fixed-time control at practically sized isolated intersections.

[4] J.H.T. Hendriks. Validation of traffic models using PTV VISSIM 7, April 2015.

[5] M. Kruijt. ITS scenario manager: De methode om voertuigafhankelijke regelingen te simuleren, June 2015.

[6] P.R. Kumar and T.I. Seidman. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *Automatic Control, IEEE Transactions on*, 35(3):289–298, March 1990.

[7] S. Lämmer and D. Helbing. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(04):P04019, 2008.

[8] N/A. Gestandaardiseerde nummering van signaalgroepen. http://www.wegenwiki.nl/Verkeersregeling#Gestandaardiseerde_nummering_van_signaalgroepen. Accessed: 10 Jun 2015.

[9] M.S. van den Broek, J.S.H. van Leeuwaarden, I.J.B.F. Adan, and O.J. Boxma. Bounds and approximations for the fixed-cycle traffic-light queue. *Transportation Science*, 40(4):484–496, November 2006.

[10] J.W.C.M. van der Vleuten. Networks of signalized traffic intersections. Master thesis, Eindhoven University of Technology (TU/e), July 2014.

[11] D. van Zwieten. *Fluid Flow Switching Servers*. PhD thesis, Eindhoven University of Technology (TU/e), 2014.

[12] CBS (Centraal Bureau voor de Statistiek). Verkeersintensiteit, verkeersindex alle wegen, werkdagen. http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=81435ned&D1=0&D2=a&D3=0&D4=a&HDR=G3&STB=G1,T,G2&CHARTTYPE=1&VW=T. Accessed: 01 Sep 2015.

[13] A. Wilson, R. de Groot, and CROW. *Handboek verkeerslichtenregelingen 2014*. CROW, 2014.

Appendices

Appendix A

Test Cases used

This Appendix contains all cases considered in the course of this report, with their used and assumed variables.

A.1 Cases Used for Comparison

In Chapter 3 the results of the comparison of the tools (COCON and FTT) is discussed. The following intersections were used:

A.1.1 Bleulandweg - Büchnerweg, Gouda

A.1.2 Larenstein, De Bilt

A.1.3 The Wall, Utrecht

A.1.4 Hertekop, Amersfoort

The obtained COCON databases consisted of already produced schedules. Basic information is obtained from these databases and these FTSs such to make an as fair as possible comparison.

Furthermore, the acronyms MR and ER are used to refer to the morning and evening rush hour situations, and all time variables are defined in seconds unless stated otherwise. The average delay, which is used to compare the schedules is computed using Akçelik's [1] expression, which is also used as objective function in the optimisation of FTT.

Lastly, the used maximal allowed saturations are applied to each signal group in that scenario, and are obtained from the signal group with the highest one in COCON. This has been done to obtain a fair comparison, as FTT would produce a tighter schedule otherwise, thus performing better on the average delay.

A.1.1 Bleulandweg - Büchnerweg, Gouda

The used information is represented in Table A.1.

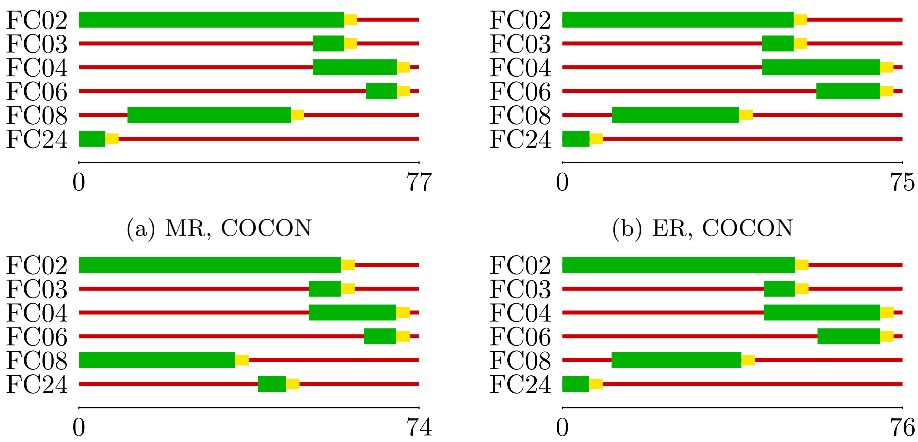
Table A.1: Signal group information: arrival rates for MR and ER, and processing rates in [PCE/h]; followed by minimal green and red, amber, and lost amber and green times in [s].

Signal Group i	λ_i^{MR}	λ_i^{ER}	μ	$g_{\min}$	$r_{\min}$	y	y^{lost}	g^{lost}
FC02	370	481	2000	7	2	3	3	1
FC03	80	164	1900	7	2	3	3	1
FC04	194	80	1700	7	2	3	3	1
FC06	167	304	1900	7	2	3	3	1
FC08	705	598	1940	7	2	3	3	1
FC24	200	200	5000	6	2	3	3	1

Furthermore, the conflict/clearance-time matrix is

$$\Sigma = \begin{bmatrix} \cdot & & & 2 & & \\ & \cdot & & 2 & 2 & 2 \\ & & \cdot & & 2 & 2 \\ 2 & 2 & & \cdot & 2 & 2 \\ & 2 & 2 & 2 & \cdot & 2 \\ & 2 & 2 & 2 & 2 & \cdot \end{bmatrix}, \tag{A.1}$$

and the maximum allowed saturations $\rho_{\max}^{MR} = 0.75$, and $\rho_{\max}^{ER} = 0.77$. The obtained FTSs are shown in Figure A.1 for both methods.



(a) MR, COCON (b) ER, COCON

(c) MR, FTT (d) ER, FTT

Figure A.1: FTS for both the morning and the evening rush hour, made in both tools

The reason FTT obtains a slightly longer schedule than was obtained in COCON (ER), is that the maximal occurring saturation in COCON is 0.77 is actually rounded from 0.771. Imposing a maximal allowed saturation of 0.771 in FTT results in the exact same schedule.

Table A.2: The delay experienced by the road users according to Akçelik definition in [s].

	MR	ER
COCON	16.8	19.6
FTT	16.6	19.7
FTT ($\rho_{\max}^{ER} = 0.771$)	-	19.6
improvement (%)	1.2	(-0.5) 0.0

A.1.2 Larenstein, De Bilt

Basic information is shown in Table A.3.

Table A.3: Signal group information: arrival rates for MR and ER, and processing rates in [PCE/h]; followed by minimal green and red, amber, and lost amber and green times in [s].

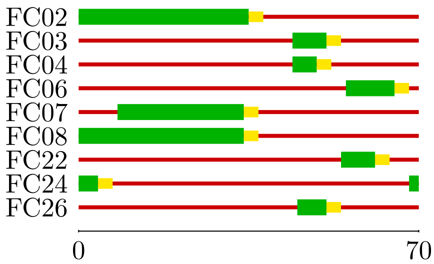
Signal Group i	λ_i^{MR}	λ_i^{ER}	μ	$g_{\min}$	$r_{\min}$	y	y^{lost}	g^{lost}
FC02	628	726	1800	5	2	3	2	1
FC03	165	175	1700	5	2	3	2	1
FC04	127	126	1700	5	2	3	2	1
FC06	216	308	1600	5	2	3	2	1
FC07	248	285	1700	5	2	3	2	1
FC08	789	670	1800	5	2	3	2	1
FC22	40	40	5000	6	2	3	2	1
FC24	40	40	5000	6	2	3	2	1
FC26	40	40	5000	6	2	3	2	1

Furthermore, the conflict/clearance-time matrix is

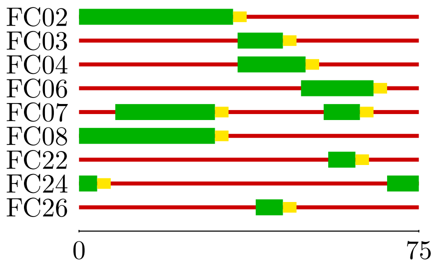
$$\Sigma = \begin{bmatrix} \cdot & & & 0 & & & 0 & & 2 \\ & \cdot & & 1 & 2 & 2 & 0 & 2 & \\ & & \cdot & & & 1 & 2 & 0 & \\ 2 & 2 & & \cdot & & 2 & & 0 & 3 \\ & 2 & & & \cdot & & & 3 & 0 \\ & 2 & 2 & 2 & & \cdot & 3 & & 0 \\ 6 & 6 & 1 & & & 1 & \cdot & & \\ & 1 & 6 & 6 & 1 & & & \cdot & \\ 1 & & & 1 & 6 & 6 & & & \cdot \end{bmatrix}, \tag{A.2}$$

and the maximum allowed saturations $\rho_{\max}^{MR} = 0.81$, and $\rho_{\max}^{ER} = 0.84$. The obtained FTSs are shown in Figure A.2 for both methods.

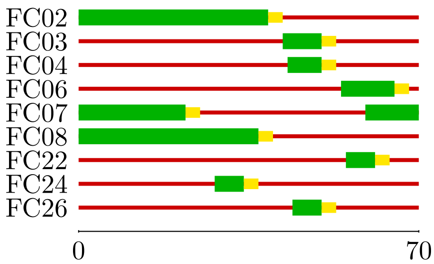
It has to be noted that in the original (COCON) schedule for the evening rush, one of the signal groups (FC07) is assigned green twice within the cycle. This has not been considered in FTT, because an acceptable schedule can be obtained with green only once per cycle, which is preferable over one with a second green interval. The main reason for this is that the road users often anticipate on their green times, with reference to the order of the other signal groups' green times. The reason why in COCON this realisation was added, is probably due to the desire to keep the order the same for both morning and evening control sequences, and having insufficient capacity to allow for only one green interval.



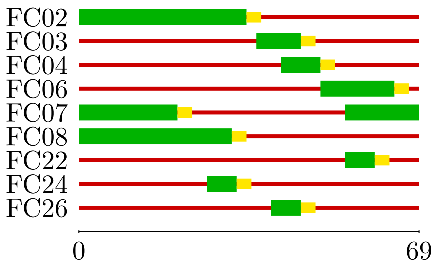
(a) MR, COCON



(b) ER, COCON



(c) MR, FTT



(d) ER, FTT

Figure A.2: FTS for both the morning and the evening rush hour, made in both tools

Table A.4: The delay experienced by the road users according to Akçelik definition in [s].

	MR	ER
COCON	20.2	21.4
FTT	18.7	21.4
improvement (%)	7.4	0.0

A.1.3 The Wall, Utrecht

This test case does not consist of morning and evening data, but of a current layout and a desired layout. Basic information is shown in Table A.5.

Table A.5: Signal group information: arrival rates, and processing rates for the current and desired situation in [PCE/h]; followed by minimal green and red, amber, and lost amber and green times in [s].

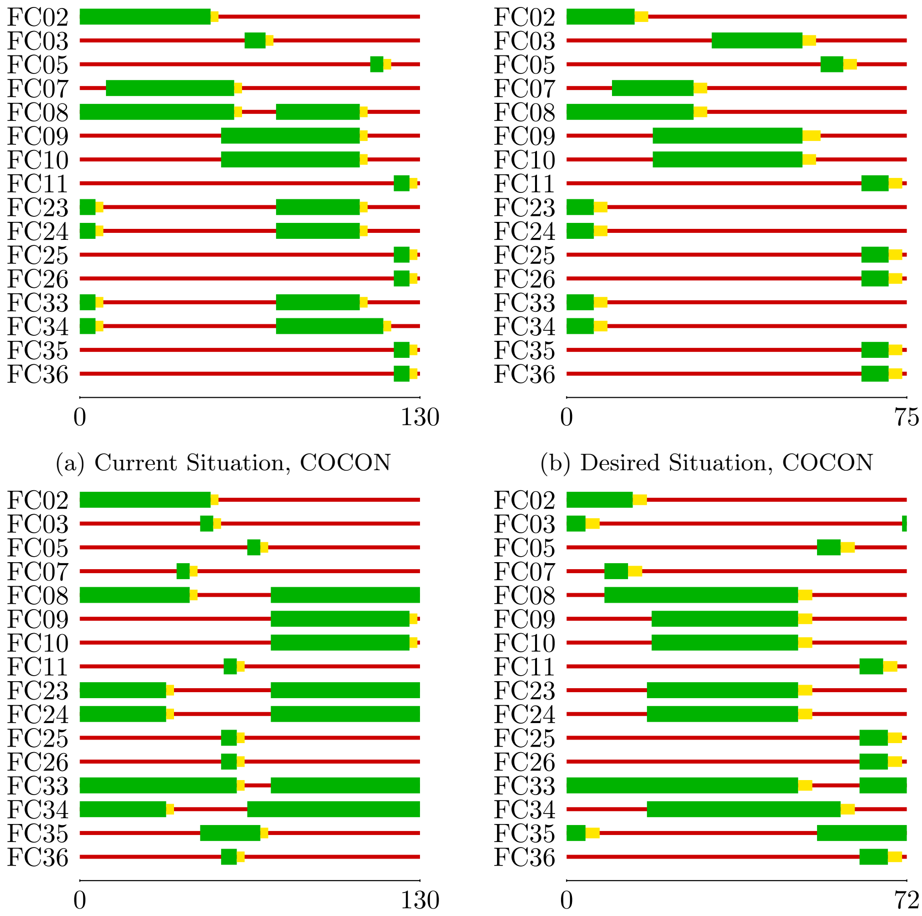
Signal Group i	λ_i	μ_i^{curr}	μ_i^{des}	$g_{\min}$	$r_{\min}$	y	y^{lost}	g^{lost}
FC02	630	1900	3800	5	2	3	2	1
FC03	5	1800	1800	5	2	3	2	1
FC05	30	1800	1800	5	2	3	2	1
FC07	20	1900	1900	5	2	3	2	1
FC08	625	2000	2000	5	2	3	2	1
FC09	635	1800	1800	5	2	3	2	1
FC10	635	1900	1900	5	2	3	2	1
FC11	10	1850	1850	5	2	3	2	1
FC23	40	5000	5000	6	2	3	2	1
FC24	40	5000	5000	6	2	3	2	1
FC25	40	5000	5000	6	2	3	2	1
FC26	40	5000	5000	6	2	3	2	1
FC33	100	9999	9999	6	2	3	2	1
FC34	100	9999	9999	6	2	3	2	1
FC35	100	9999	9999	6	2	3	2	1
FC36	100	9999	9999	6	2	3	2	1

Furthermore, the conflict/clearance-time matrix is

$$\Sigma = \begin{bmatrix}
\cdot & & 1 & & & 1 & 1 & 1 & & & 1 & 1 & & & & 1 \\
& \cdot & 1 & 1 & 1 & & & 1 & 1 & 1 & & & & 1 & & \\
1 & 1 & \cdot & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & 1 \\
& 1 & & \cdot & & & & 1 & 1 & 1 & 1 & 1 & & 1 & 1 & \\
& 1 & 1 & & \cdot & & & 1 & & & 1 & 1 & & & 1 & \\
1 & & 1 & & & \cdot & & 1 & & & 1 & 1 & & & 1 & \\
1 & & 1 & & & & \cdot & & & & 1 & 1 & & & & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & & \cdot & 1 & 1 & & & & 1 & & \\
& 1 & 1 & 1 & & & & 1 & \cdot & & & & & & & \\
& 1 & 1 & 1 & & & & 1 & & \cdot & & & & & & \\
1 & & 1 & 1 & 1 & 1 & 1 & & & & \cdot & & & & & \\
1 & & 1 & 1 & 1 & 1 & 1 & & & & & \cdot & & & & \\
& & 1 & & & & & & & & & & \cdot & & & \\
& 1 & & 1 & & & & 1 & & & & & & \cdot & & \\
& & & 1 & 1 & 1 & & & & & & & & & \cdot & \\
1 & & 1 & & & & 1 & & & & & & & & & \cdot
\end{bmatrix}, \tag{A.3}$$

and the maximum allowed saturations $\rho_{\max}^{Curr} = 0.85$, and $\rho_{\max}^{Want} = 0.80$. The obtained FTSs are shown in Figure A.3 for both methods.

Here as well, like in the ER case of A.1.2, the COCON schedule of the current situation compasses multiple green periods for some signal groups, and also in this case they have not been considered for the FTT. The reason is similar: the results are acceptable while comparing them to the one from COCON. I.e. the cycle time and the expected delay do



(a) Current Situation, COCON
(b) Desired Situation, COCON
(c) Current Situation, FTT
(d) Desired Situation, FTT

Figure A.3: FTS for both the current and the desired situation, made in both tools.

not differ a lot, while not using a second green period. Moreover in this example the desired situation is of importance, in which no second green period is used.

Table A.6: The delay experienced by the road users according to Akçelik definition in [s].

	Current	Desired
COCON	32.6	24.5
FTT	31.2	18.9
improvement (%)	4.3	22.9

A.1.4 Hertekop, Amersfoort

This intersection is special as well, as it consists of multiple junctions, controlled by one TCI. This is also the reason the information is available in a single file, which is shown in Table A.7. Because there are multiple junctions, designers consider follow-up lanes. To be able to take these into account, the schematic representation of the intersection is shown in Figure A.4.

Table A.7: Signal group information: arrival rates for MR and ER, and processing rates in [PCE/h]; followed by minimal green and red, amber, and lost amber and green times in [s].

Signal Group i	λ_i^{MR}	λ_i^{ER}	μ_i	$g_{\min}$	$r_{\min}$	y	y^{lost}	g^{lost}
FC101	201	208	1615	4	2	3	2	1
FC102	740	927	3800	5	2	4	2	1
FC104	312	274	1615	4	2	3	2	1
FC106	163	294	1805	4	2	3	2	1
FC108	380	335	1900	5	2	4	2	1
FC109	614	235	1805	4	2	3	2	1
FC123	50	50	5000	6	2	2	2	1
FC124	50	50	5000	6	2	2	2	1
FC201	54	13	1615	4	2	3	2	1
FC202	436	683	3800	5	2	4	2	1
FC203	413	525	1805	4	2	3	2	1
FC205	10	10	1786	4	2	3	2	1
FC207	422	265	1615	4	2	3	2	1
FC208	989	565	3800	5	2	4	2	1
FC209	127	12	1805	4	2	3	2	1
FC211	27	243	1900	4	2	3	2	1
FC212	471	836	3667	4	2	3	2	1
FC265	132	17	1900	4	2	3	2	1
FC270	222	678	3230	4	2	3	2	1
FC271	10	83	1900	4	2	3	2	1
FC272	244	318	1805	4	2	3	2	1
FC221	50	50	5000	6	2	2	2	1
FC222	50	50	5000	6	2	2	2	1
FC223	50	50	5000	6	2	2	2	1
FC224	50	50	5000	6	2	2	2	1
FC227	50	50	5000	6	2	2	2	1
FC228	50	50	5000	6	2	2	2	1
FC281	50	50	5000	6	2	2	2	1
FC282	50	50	5000	6	2	2	2	1

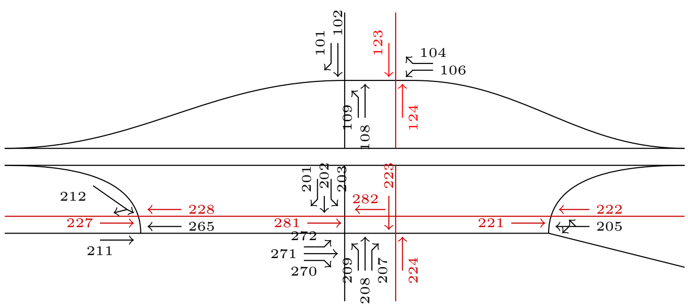


Figure A.4: Schematic representation of the "Hertekop" in Amersfoort

Furthermore, the conflict/clearance-time matrix is

$$
\Sigma = \begin{bmatrix}
\cdot \quad 0 \\
\cdot \quad 0 \quad 3 \\
\cdot \quad 0 \quad 1 \; 0 \\
2 \quad \cdot \; 3 \; 5 \; 0 \; 1 \\
0 \; 0 \; \cdot \\
2 \; 0 \quad 0 \quad \cdot \\
2 \; 4 \quad \cdot \\
4 \; 1 \quad \cdot \\
\cdot \quad 0 \quad 0 \quad 2 \; 0 \\
\cdot \quad 0 \quad 2 \quad 2 \; 1 \; 0 \quad 0 \; 0 \\
\cdot \; 8 \; 4 \; 2 \quad 1 \; 0 \; 11 \; 10 \; 4 \; 6 \quad 0 \; 1 \\
1 \; 3 \; 3 \; \cdot \; 0 \; 4 \; 6 \quad 7 \; 5 \; 3 \; 3 \; 3 \; 4 \; 3 \quad 4 \; 7 \\
0 \; 3 \; \cdot \quad 0 \quad 6 \quad 6 \; 0 \; 3 \\
0 \; 0 \quad \cdot \quad 0 \; 0 \quad 0 \; 0 \\
0 \; 0 \quad 0 \quad \cdot \quad 0 \; 2 \\
\cdot \; 0 \\
0 \; \cdot \; 1 \quad 1 \; 1 \\
0 \; \cdot \\
0 \quad 0 \quad \cdot \\
0 \; 0 \; 3 \; 1 \; 0 \; 0 \quad \cdot \quad 6 \quad 6 \; 1 \; 3 \\
0 \; 3 \; 0 \quad 0 \; 0 \quad \cdot \quad 3 \; 5 \\
0 \; 2 \; 0 \quad 0 \quad \cdot \\
0 \; 2 \; 0 \quad 0 \quad \cdot \\
0 \; 0 \; 5 \quad 1 \quad \cdot \\
0 \; 2 \; 1 \quad 0 \quad \cdot \\
5 \quad \cdot \\
5 \quad \cdot \\
0 \; 2 \; 3 \; 1 \quad 2 \quad 2 \quad \cdot \\
4 \; 3 \; 2 \; 0 \quad 0 \quad 0 \quad \cdot
\end{bmatrix}, \tag{A.4}
$$

and the maximum allowed saturations $\rho^{MR}_{\max} = \rho^{ER}_{\max} = 0.89$. The obtained FTSs are not visualised due to their large size and the lack of addition of information with respect to the delay and cycle time, which are shown in Table A.8.

Table A.8: The delays experienced by the road users and the cycle lengths in [s].

	MR		ER	
	$\mathbb{E}D_{\text{Akçelik}}$	T	$\mathbb{E}D_{\text{Akçelik}}$	T
COCON	47.7	131	43.2	119
FTT	28.0	93	24.9	91
improvement (%)	41.3	29.0	42.4	23.5

Although using FTT improves the produced schedules a lot, the results are not yet comparable, as FTT does not ensure good follow-up flows. E.g. signal group $FC211$ cannot leave the system without passing trough either $FC270$, $FC271$ or $FC272$. Currently, the tool does not provide for adding green after red constraints, necessary to intuitively add these to the optimisation. The only option to take these flows into account is to add more conflicts and clearance times (green after green) constraints. Firstly, it was attempted to remove all follow-up signal groups. This means $FC265$, $FC270$, $FC271$ and $FC272$ are removed from the system (Table A.7), and compensated for by adding their conflicts to the preceding signal groups. Moreover, the clearance times must be updated to take into account the distance between the original conflicts and the stop line. Doing so, results in the clearance time matrix

$$\Sigma_2 = \begin{bmatrix} \cdot & 0 \\ \cdot & 0 & 3 \\ \cdot & 0 & 1 & 0 \\ 2 & \cdot & 3 & 5 & 0 & 1 \\ 0 & 0 & \cdot \\ 2 & 0 & 0 & \cdot \\ 2 & 4 & \cdot \\ 4 & 1 & \cdot \\ \cdot & 0 & 0 & 8 & 2 & 0 \\ \cdot & 0 & 2 & -7 & -6 & 0 & 0 \\ \cdot & 8 & 4 & 2 & -9 & -7 & 11 & 10 & 4 & 6 & 0 & 1 \\ 1 & 3 & 3 & \cdot & 0 & 4 & 6 & -2 & 14 & 3 & 3 & 4 & 3 & 4 & 7 \\ 0 & 3 & \cdot & -10 & 6 & 6 & 0 & 3 \\ 0 & 0 & \cdot & -10 & -8 & 0 & 0 \\ 0 & 0 & 0 & \cdot & 9 \\ 9 & 10 & 12 & 11 & 10 & \cdot & 0 & 16 & 16 & 11 & 13 & 13 & 15 \\ -7 & 8 & 8 & 8 & 8 & 9 & 0 & \cdot & 1 & 1 & 11 & 13 \\ 0 & 2 & 0 & -15 & \cdot \\ 0 & 2 & 0 & -15 & \cdot \\ 0 & 0 & 5 & -9 & \cdot \\ 0 & 2 & 1 & -12 & \cdot \\ 5 & \cdot \\ 5 & \cdot \\ 0 & 2 & 3 & 1 & 2 & -10 & -8 & \cdot \\ 4 & 3 & 2 & 0 & 0 & -14 & -12 & \cdot \end{bmatrix}. \tag{A.5}$$

Unfortunately, this only yields a feasible result for the morning rush. The schedule for the evening rush can be obtained, with a cycle time of 256 seconds, which is much too high to be used on the TCI. To find a feasible, yet comparable schedule for the evening rush, only the follow-ups are considered that have the highest arrival rates, being $FC211$ and $FC212$ towards $FC272$. To do so, the follow-up signal groups cannot be removed, but the conflicts as well as the clearance times need to be copied to the preceding signal groups as done in Σ_2. This results in clearance time matrix

$$\Sigma_3 = \begin{bmatrix} \cdot & 0 \\ \cdot & 0 & 3 \\ \cdot & 0 & 1 & 0 \\ 2 & \cdot & 3 & 5 & 0 & 1 \\ 0 & 0 & \cdot \\ 2 & 0 & 0 & \cdot \\ 2 & 4 & \cdot \\ 4 & 1 & \cdot \\ \cdot & 0 & 0 & 2 & 0 \\ \cdot & 0 & 2 & -7 & -6 & 2 & 1 & 0 & 0 & 0 \\ \cdot & 8 & 4 & 2 & 1 & 0 & 11 & 10 & 4 & 6 & 0 & 1 \\ 1 & 3 & 3 & \cdot & 0 & 4 & 6 & -2 & 14 & 7 & 5 & 3 & 3 & 3 & 4 & 3 & 4 & 7 \\ 0 & 3 & \cdot & 0 & 6 & 6 & 0 & 3 \\ 0 & 0 & \cdot & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdot & 0 & 2 \\ 9 & 12 & \cdot & 0 \\ 8 & 8 & 0 & \cdot & 1 & 1 & 1 \\ 0 & \cdot \\ 0 & 0 & \cdot \\ 0 & 0 & 3 & 1 & 0 & 0 & \cdot & 6 & 6 & 1 & 3 \\ 0 & 3 & 0 & 0 & 0 & \cdot & 3 & 5 \\ 0 & 2 & 0 & 0 & \cdot \\ 0 & 2 & 0 & 0 & \cdot \\ 0 & 0 & 5 & 1 & \cdot \\ 0 & 2 & 1 & 0 & \cdot \\ 5 & \cdot \\ 5 & \cdot \\ 0 & 2 & 3 & 1 & 2 & 2 & \cdot \\ 4 & 3 & 2 & 0 & 0 & 0 & \cdot \end{bmatrix}. \tag{A.6}$$

Using Σ_2 and Σ_3 results in schedules that are comparable in execution as the original COCON schemes. The cycle times and computed delays for these new situations are still a great improvement to the original schedules, and are shown in Table A.9.

Table A.9: The delays experienced by the road users and the cycle lengths in [s].

	MR		ER	
	$\mathbb{E}D_{\text{Akçelik}}$	T	$\mathbb{E}D_{\text{Akçelik}}$	T
COCON	47.7	131	43.2	119
FTT (Σ_2)	33.4	105	-	-
FTT (Σ_3)	-	-	27.2	88
improvement (%)	30.0	19.8	37.0	26.1

A.2 Kumar-Seidman Problem

The Kumar-Seidman problem [6] is an example of a switching system, used to show that stable control policies can become unstable under certain conditions. It can be easily translated to a simple road network [7] as shown in Figure 4.8. This scenario consists of only two signal groups per intersection: 3 and 8 for the left one, and 2 and 9 for the other. Moreover, the successive signal group, curving off the main road, undergo a reduction in lanes, lowering their maximal processing rate.

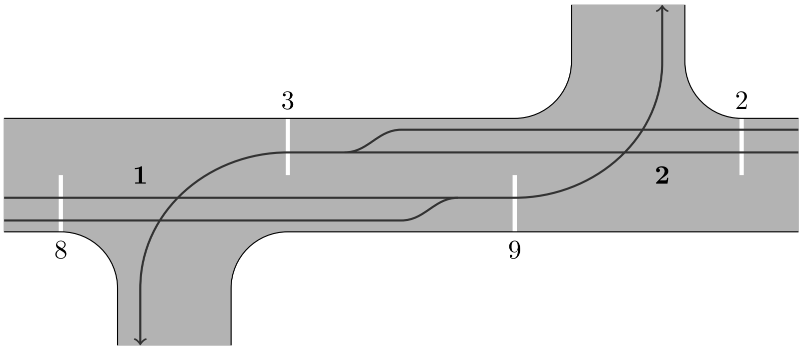


Figure A.5: The Kumar-Seidman problem.

The data used to come up with the FTSs is stated in Table A.10, and the clearance time matrices are given in (A.7).

Table A.10: Input values for the Kumar-Seidman problem.

Signal Group	Arrival Rate [PAE/h]	Processing Rate [PAE/h]
1-3	1100	1800
1-8	1100	3800
2-2	1100	3800
2-9	1100	1800

$$\Sigma_1 = \begin{bmatrix} \cdot & 3 \\ 0 & \cdot \end{bmatrix}, \Sigma_2 = \begin{bmatrix} \cdot & 0 \\ 3 & \cdot \end{bmatrix}. \tag{A.7}$$

Furthermore, by choice of the above rates, the maximal saturation rate is 0.95.

As for the network context, the distance between the two intersections is $x_d|_{1\to 2} = 200\text{m}$ and the average speed on this part is assumed $\overline{v} = 50\text{km/h}$.

A.3 De Uithof

Near the headquarters of Grontmij, in “De Bilt”, a group of two junctions called “De Uithof”, is located. This group of intersections is used as a test case because it is currently being looked into and very crowded during rush hours. It consists of two intersections, hereinafter considered 1 and 2, being a 4-way and a 3-way junction, respectively. The basic information for both intersections can be found in Tabel A.11.

Table A.11: Signal group information: arrival rates for MR and ER, and processing rates in [PCE/h], and; minimal green and red, amber, and lost amber and green times in [s].

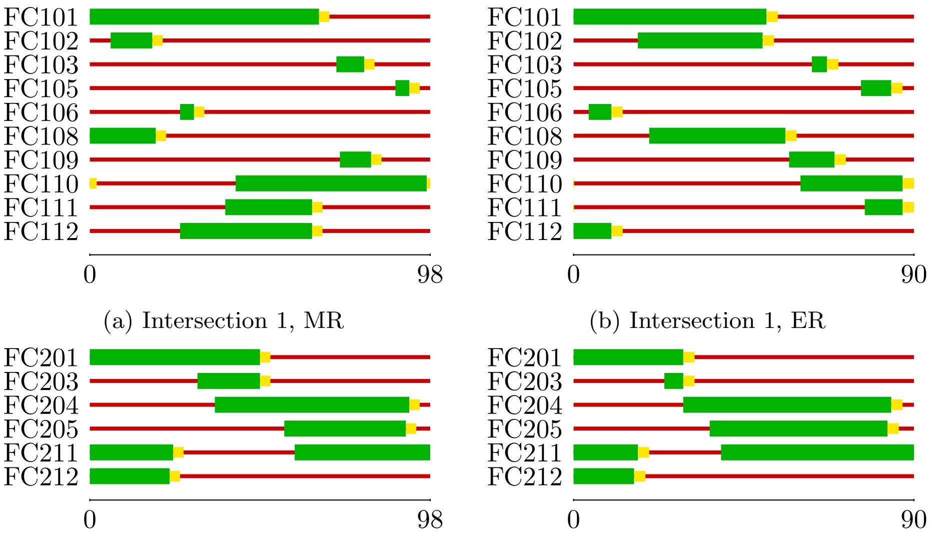
Signal Group i	λ_i^{MR}	λ_i^{ER}	μ_i	$g_{\min}$	$r_{\min}$	y	y^{lost}	g^{lost}
FC101	71	374	1615	4	5	3	1	2
FC102	296	1140	3800	4	5	3	1	2
FC103	13	1	1805	4	5	3	1	2
FC105	0	96	1615	4	5	3	1	2
FC106	4	124	3610	4	5	3	1	2
FC108	559	219	3800	4	5	3	1	2
FC109	111	168	1805	4	5	3	1	2
FC110	831	577	3230	4	5	3	1	2
FC111	148	2	1900	4	5	3	1	2
FC112	1127	270	3610	4	5	3	1	2
FC201	177	123	1615	4	5	3	1	2
FC203	439	97	3610	4	5	3	1	2
FC204	262	706	3230	4	5	3	1	2
FC205	862	1143	3800	4	5	3	1	2
FC211	345	303	1900	4	5	3	1	2
FC212	572	417	3610	4	5	3	1	2

Furthermore, the conflict/clearance-time matrices are

$$\Sigma_1 = \begin{bmatrix} \cdot & & & 4 & & & 3 & & & \\ & \cdot & & 4 & 4 & & 4 & 7 & 6 & 5 \\ & & \cdot & 6 & 8 & 10 & & & 7 & 6 \\ 3 & 3 & 3 & \cdot & & 3 & 3 & & & 3 \\ & 4 & 3 & & \cdot & 4 & 6 & 9 & 6 & \\ & & 4 & 4 & 4 & \cdot & & & 4 & 4 \\ 4 & 4 & & 4 & 3 & & \cdot & & 3 & 5 \\ & 4 & & & 4 & & & \cdot & & \\ & 4 & 4 & & 4 & 6 & 5 & & \cdot & \\ & 4 & 4 & 8 & & 7 & 5 & & & \cdot \end{bmatrix}, \Sigma_2 = \begin{bmatrix} \cdot & & & 4 & & \\ & \cdot & & 4 & 7 & 6 \\ & & \cdot & & & 3 \\ 4 & 4 & & \cdot & & 4 \\ & 4 & & & \cdot & \\ & 5 & 10 & 9 & & \cdot \end{bmatrix}, \tag{A.8}$$

and the maximal saturation rate is 0.9.

The resulting FTSs obtained using FTT are shown in Figure A.6.



(a) Intersection 1, MR

(b) Intersection 1, ER

(c) Intersection 2, MR

(d) Intersection 2, ER

Figure A.6: FTS for both intersections, for both the morning and the evening rush hour.

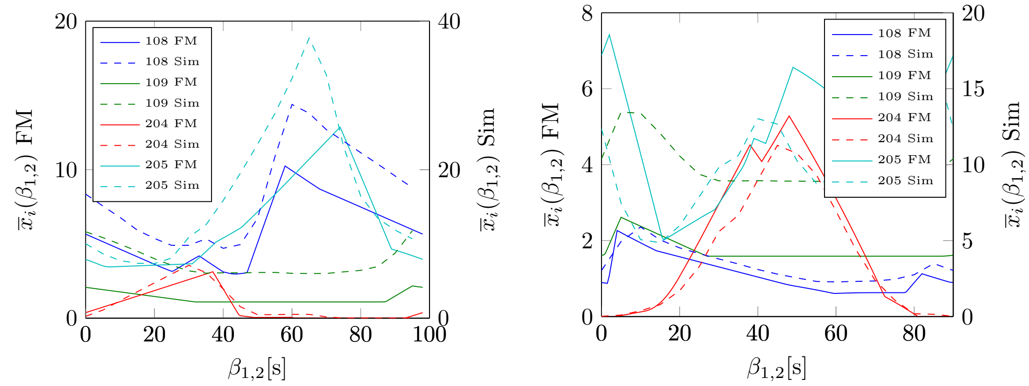
The used follow-up fractions are

$$f_{102\to204} = 0.8942; \tag{A.9a}$$
$$f_{102\to205} = 0.1058; \tag{A.9b}$$
$$f_{106\to204} = 1.0000; \tag{A.9c}$$
$$f_{110\to205} = 1.0000; \tag{A.9d}$$
$$f_{203\to108} = 0.8314; \tag{A.9e}$$
$$f_{211\to108} = 0.5623; \tag{A.9f}$$
$$f_{211\to109} = 0.3217, \tag{A.9g}$$

for the morning rush hour, and

$$f_{102\to204} = 0.5553; \tag{A.10a}$$
$$f_{102\to205} = 0.4447; \tag{A.10b}$$
$$f_{106\to204} = 0.5564; \tag{A.10c}$$
$$f_{106\to205} = 0.4436; \tag{A.10d}$$
$$f_{110\to205} = 1.0000; \tag{A.10e}$$
$$f_{203\to108} = 0.9897; \tag{A.10f}$$
$$f_{211\to108} = 0.4455; \tag{A.10g}$$
$$f_{211\to109} = 0.5545, \tag{A.10h}$$

for the evening rush hour. Moreover, the distance between the two intersections is $x_d|_{1\to2} = 312\text{m}$ and the average speed on this part is assumed $\overline{v} = 50\text{km/h}$. Using this information, the average queue lengths can be computed for multiple phase shifts $\beta_{1,2}$, of which the results are shown in Figure A.7.



(a) MR (b) ER

Figure A.7: Comparison of fluid model (FM) and microsimulations (Sim) of the average queue lengths over one period for: a the morning rush hour, and; b the evening rush hour.

Appendix B

Quick Comparison of Block Structures

As setting up simulations to genuinely compare block structures is a time consuming approach, a quick (visual) comparison is carried out on test case A.1.4. The block structures, obtained as explained in Section 5.2.2, for both the morning and the evening rush hours, generated from the FTSs obtained from both COCON and Fixed-Time Tool (FTT), are compared to the one that has been implemented on the actual intersection in 2007. This is shown in Table B.1 Note, that in practice, only one block structure is implemented for the entire control strategy, regardless of the time of execution.

Table B.1: Comparison of block structures of A.1.4

	Implemented	COCON, MR	COCON, ER	FTT, MR	FTT, ER
Blocks	6	5	5	5	5
FI	0.6294	0.65061	0.64766	0.65522	0.64818

The corresponding block structures in same order as in the table are

$$\begin{aligned}
&\{201, 202, 207, 208\};\\
&\{108, 109, 123, 124, 205\};\\
&\{203\};\\
&\{104, 106, 209, 211, 265, 221, 222, 223, 224, 227, 228, 281, 282\};\\
&\{212, 270, 271, 272\};\\
&\{101, 102\},
\end{aligned} \tag{B.1}$$

for the actual implemented one;

$$\begin{aligned}
&\{102, 205, 211, 265, 227, 228\};\\
&\{104, 106, 201, 202, 203\};\\
&\{108, 109, 123, 124, 208, 209, 270, 221, 222, 223, 224\};\\
&\{101, 207, 212, 281, 282\};\\
&\{271, 272\},
\end{aligned} \tag{B.2}$$

for the one generated from the FTS from COCON, for the morning rush;

$$\begin{aligned}
&\{104, 106, 205\};\\
&\{108, 109, 123, 124, 201, 202, 203\};\\
&\{101, 102, 208, 209, 212, 270, 221, 222, 223, 224\};\\
&\{207, 211, 265, 227, 228, 281, 282\};\\
&\{271, 272\},
\end{aligned} \tag{B.3}$$

for the one generated from the FTS from COCON, for the evening rush;

$$\begin{aligned}
&\{106, 212, 270, 271, 272\};\\
&\{108, 109, 123, 124, 207, 208, 209\};\\
&\{221, 222, 223, 224, 281, 282\};\\
&\{101, 102, 104, 205, 211, 265, 227, 228\};\\
&\{201, 202, 203\},
\end{aligned} \tag{B.4}$$

for the one generated from the FTS from FTT, for the morning rush;

$$\begin{aligned}
&\{101, 102, 211, 270, 271, 272\};\\
&\{104, 207, 208, 209, 212\};\\
&\{106, 265, 221, 222, 223, 224, 227, 228, 281, 282\};\\
&\{108, 109, 123, 124, 205\};\\
&\{201, 202, 203\},
\end{aligned} \tag{B.5}$$

for the one generated from the FTS from FTT, for the evening rush.

The first thing that comes to mind is that all the generated block structures seem better than the one actually implemented. Unfortunately, this example shows that for combined systems (networks), the proposed algorithm does not produce reliable block structures. The follow-up lanes do not have any conflicts with each other, and therefore, could be put in the same block. However, this means that the green periods will most probably overlap, even though for a good follow-up lane, the green periods should better be subsequent. By putting them in the same block, the green phase of the consecutive flow will most likely have a negative influence on how the blocks are processed. This aspect is indeed taken into account in the implemented block structure, e.g. 201 and 202 are put in the block after 102 and the same holds for 108 and 109 with respect to 208, and 270, 271 and 272 with respect to 211. Unfortunately, this results in a lower FI, which in this case fails to give a representative measure.

The best thing to do in this case, is to produce block structures for each individual intersection, and synchronise them such to optimise the flows. This has not been carried out due to the limited amount of time. Moreover, performing simulations on VAC strategies with different block structures, could show the way the FI correlates to the actual performance of the system.