An Evaluation of the Efficiency of Self-Organising versus Fixed Traffic Signalling Paradigms

Mark David Einhorn

Thesis presented in partial fulfilment of the requirements for the degree
MSc (Operations Research)
Department of Logistics, Stellenbosch University

Supervisor: Prof JH van Vuuren
Co-supervisor: Dr AP Burger

March 2012

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2012

Abstract

Traffic congestion is a major problem in cities around the world, resulting in significant costs in terms of man hours lost and excessive fuel consumption, thereby also having an adverse environmental effect through unnecessary carbon emissions. At present, urban traffic flows are largely controlled by centralised traffic signals. However, these traffic signals are typically operated according to a fixed red-green time schedule which varies for different periods of the day (as a result of observed traffic densities), with the incorporation of adaptive traffic control systems on some South African roads. The *Split Cycle Offset Optimisation Technique* (SCOOT) utilises online traffic control together with vehicle detection equipment in an attempt to optimise network performance and has achieved a certain degree of success in South Africa, and other countries around the world. SCOOT attempts to calculate optimal signal timings and green time offsets between traffic signals at adjacent intersections in such a way that vehicles may travel unimpeded along an arterial, receiving a green signal upon their arrival at each intersection.

An alternative to the typical global optimisation approach of traffic control regimes, such as SCOOT, is the implementation of a decentralised self-organising system of traffic signals, which allows the system to "discover for itself" the most effective local traffic signal timings as a function of the current traffic situation and to adjust itself accordingly. A consequence of each intersection in a traffic network being optimised locally in terms of throughput is that a ripple-effect occurs resulting in a natural traffic signal synchronisation among intersections. Electro-magnetic induction loops are the most widely used form of vehicle detection equipment in traffic control. Recent developments in technology, however, have seen the introduction of radar systems which, when mounted on a traffic light, effectively allows the traffic light to "see" a certain distance down a stretch of roadway, enabling the controlling algorithm to perceive the number of vehicles approaching the intersection (and their respective velocities).

In this thesis two self-organising traffic control algorithms are investigated in terms of their ability to minimise vehicle waiting times and reduce traffic congestion. The data provided by the aforementioned radar detection equipment are assumed to be available to the traffic control algorithms. The performances of these two algorithms are compared to those of an optimised fixed-time cycle-based traffic control regime for a variety of road network topologies. A number of experiments are performed in a simulated environment using a traffic simulation model built specifically for the purpose of this thesis. The experimental results show that the self-organising traffic control algorithms are capable of improvements of up to 26%, 19% and 25% in terms of minimising the mean waiting times, commute times and queue lengths of the system, respectively, over the fixed control regime for the various traffic network topologies.

Uittreksel

Verkeersopeenhoping is wêreldwyd 'n ernstige problem wat beduidende kostes in terme van verlore man-ure en oormatige brandstofverbruik meebring en as gevolg van onnodige koolstofvrylating ook 'n nadelige invloed op die omgewing het. Stedelike verkeersvloeie word tans oorwegend deur middel van gesentraliseerde verkeersligte beheer. Hierdie verkeersligte funksioneer tipies volgens vaste rooi-groen faseskedules wat (op grond van waargenome verkeersdigthede) vir verskillende tye van die dag varieer, met die inkorporering van aanpasbare verkeersbeheerstelsels op sommige paaie in Suid-Afrika. Die sogenaamde *Split Cycle Offset Optimisation Technique* (SCOOT) gebruik aanlyn verkeersregulering tesame met voertuigwaarnemingstoerusting in 'n poging om die werkverrigting van vervoernetwerke te optimeer en het in Suid-Afrika en internasionaal beperkte sukses behaal. Die werking van SCOOT is gebaseer op pogings om optimale faselengtes en tydvertragings van groen fases tussen opeenvolgende straatkruisings langs hoofweë te bereken om sodoende voertuie daartoe in staat te stel om onverhinderd langs sulke hoofweë af te beweeg en tydens 'n groen fase by elke opeenvolgende kruising aan te kom.

'n Alternatief tot die tipiese globale optimeringsbenadering in die regulasie van verkeersvloei in beheerstelsels soos SCOOT is die gebruik van 'n gedesentraliseerde self-organiserende stelsel van verkeersligte, waar die stelsel die mees doeltreffende rooi-groen fases as 'n funksie van die huidige verkeersituasie "self kan ontdek" en dienooreenkomstig kan aanpas. Die gevolg van die lokale optimering van die deurvloei van verkeer deur elke straatkruising van 'n vervoernetwerk is dat 'n rippeleffek in die netwerk ontstaan waartydens die fases van verkeersligte op 'n natuurlike wyse met mekaar sinkroniseer. Elektromagnetiese induksielusse is tans die mees algemene meganisme vir voertuigwaarnemings. Onlangse tegnologiese ontwikkeling het egter gelei na die ontwerp van radar-stelsels wat op verkeersligte gemonteer kan word, en die ligte sodoende in staat kan stel om 'n sekere afstand langs die strate van 'n kruising af te "sien." Op hierdie wyse kan die beheeralgoritme van 'n stel verkeersligte die getal voertuie wat 'n kruising nader (asook hul onderskeie snelhede), waarneem.

In hierdie tesis word twee self-organiseringsalgoritmes ondersoek in terme van hul vermoë om voertuigwagtye en verkeersopeenhopings te minimeer. Daar word aangeneem dat data wat deur toerusting soos die bogenoemde radar-toestel verskaf word, aan die algoritmes beskikbaar is. Die werkverrigting van hierdie twee algoritmes word in verskillende vervoernetwerk-topologieë vergelyk met dié van 'n regime waarin vaste rooi-groen faseskedules gevolg word. 'n Aantal eksperimente word in 'n gesimuleerde omgewing wat spesiaal vir die doel van hierdie tesis ontwikkel is, uitgevoer. Die eksperimentele resultate dui daarop dat die self-organiseringsalgoritmes kan lei na verbeterings van tot 26%, 19% en 25% in terme van onderskeidelik die minimering van wagtye, pendeltye en toulengtes oor verskeie vervoernetwerk-topologieë in vergelyking met regimes waarin vaste rooi-groen faseskedules gevolg word.

Acknowledgements

The author wishes to acknowledge the following people for their various contributions towards the completion of this work:

- I wish to extend my sincerest thanks to my supervisor, Prof Jan van Vuuren, for acting not just as a supervisor over the past two years, but as a mentor and a friend. I appreciate all the time and patience he has afforded me as well as the enthusiasm he has shown towards the project and the guidance he has provided in ensuring that the work delivered is of a high standard. Above all though, it is the belief he has shown in my abilities, even when it was lacking on my part, which has been invaluable.
- I would also like to extend deep thanks to my co-supervisor, Dr Alewyn Burger, for all of his programming support and inspired ideas with respect to the topic.
- I would like to thank the Department of Logistics for the use of their excellent computing facilities as well as all the members of the Operations Research staff for making my past two years some of the best I have spent at Stellenbosch University.
- Funding in the form of two Rated Researcher Incentive Funding Bursaries (GUNs 70593 and 77248) is hereby acknowledged with gratitude.
- I would like to thank all of the other masters and doctoral students whom I have had the pleasure of working with over the past two years. After starting out as colleagues, I am pleased to be able to call each of them my friend. Their hard work has been both an inspiration and a motivation to me. In particular, I would like to thank Jacques du Toit for his tireless assistance with any IT related problems I encountered.
- Finally, I would like to extend my deepest gratitude to my family, in particular, my mother, Claire, my father, Peter, and my sister, Paula, as well as all of my close friends for their unwavering support and encouragement. None of my achievements would have been possible had it not been for those I love.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

CDF Cumulative distribution function

CIVACR Currently implemented vehicle actuated control regime

CONTRAM Continuous traffic assignment model

IBM International Business Machines

NETSIM Network simulation model

OPS I Optimising prioritisation strategy I

OPS II Optimising prioritisation strategy II

PDF Probability distribution function

SCOOT Split cycle offset optimisation technique

SD System dynamics

SOTCA I Self-organising traffic control algorithm I

SOTCA II Self-organising traffic control algorithm II

SOTL Self-organising traffic lights

SS Stabilisation strategy

TRANSYT Traffic network study tool

VISSIM Traffic in towns: Simulation (translation)

CHAPTER 1

Introduction

Contents

1.1 Background

In a recent survey carried out by IBM [21], it was found that in the United States of America alone, as population grew by nearly 20% during the period 1982–2001, traffic volumes increased by 236%. In the same report, this increase in traffic volumes is cited as one of the main causes of the annual loss of 3.7 billion man-hours spent in congested traffic. More than 2.3 billion gallons of fuel are burnt needlessly every year in the United States of America alone, due to people being delayed by traffic. These losses equate to a cost to the American economy of $78 billion per annum [21].

However, the debilitating consequences of traffic congestion are not experienced in the United States of America alone, but indeed the world over. In a second survey by IBM, entitled *Frustration Rising: IBM 2011 Commuter Pain Survey* [22], over 8 000 motorists from 20 different cities (approximately 400 per city) around the world were surveyed to investigate the effects of traffic on their everyday lives in terms of factors such as stress, anger, health and performance at work or school. The cities selected for the survey were ranked among the 65 top cities in the world in terms of their size and economic activity, and included Bangalore, Beijing, Buenos Aires, Chicago, Johannesburg, London, Los Angeles, Madrid, Mexico City, Milan, Montreal, Moscow, Nairobi, New Delhi, New York City, Paris, Shenzhen, Singapore, Stockholm and Toronto [22]. From this survey, it was found that of all the respondents, 91% had been held up in traffic over the preceding 3 years, with an average maximum delay reported to be 1.3 hours, while 42% admitted that their stress levels had increased due to the adverse consequences of traffic. More specifically, it was found that within the last month before the time of completing the survey, 47% of the respondents admitted that they had at least once forgone a planned trip due to inclement traffic conditions. Of this 47%, 24% had been destined for work, 21% for shopping, 17% for recreation, 11% for entertainment, and 11% for eating out.

Based on the above facts, it is clear that improving the flow of traffic along urban roads is expected to yield significant economic, environmental and social benefits by reducing the amount of time commuters are required to spend in traffic.

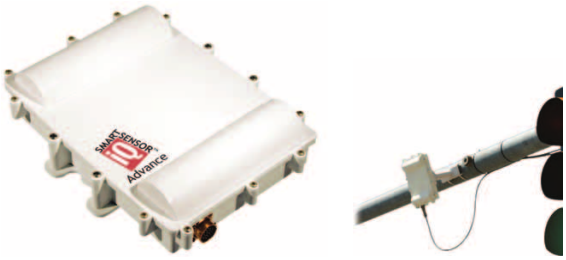
1.2 Informal problem description

One approach towards reducing driver commute times and easing congestion is the physical alteration of the roadways along which motorists travel, such as, for example, the widening of highways through the addition of more traffic lanes, or the construction of fly-overs or bridges at particularly congested intersections. A draw-back of such alterations is that they require extensive planning and their implementation requires considerable expenditure of both time and capital. Moreover, these alterations often result in worsened traffic conditions during their construction and in some cases even after completion. Physical alterations may in any case be seen as a temporary solution, as traffic volumes are typically expected to increase over time, resulting in the alterations becoming more ineffective as time goes by.

An alternative to separating conflicting traffic flows in space, is to separate them in time. This is the concept behind the implementation of traffic signals at street intersections: stopping one traffic flow for a period of time, while a conflicting traffic flow proceeds through the intersection [32]. Interestingly, in the aforementioned survey carried out by IBM [22], drivers in Los Angeles, Mexico City, India, China Singapore and Johannesburg listed stop-and-go traffic as the most painful experience of their daily commutes. For decades, researches have been proposing various mathematical models in an attempt to optimise traffic flow in networks and at signalised intersections [15]. These models include vehicle following models, as well as fluid dynamic traffic models [25].

The optimisation of the cycle timings of traffic signals responsible for controlling traffic flow at intersections is of critical importance when attempting to optimise the overall traffic flow through a network. A typical goal associated with the optimisation of traffic signals is to find an optimal signal cycle time, based on the spatio-temporal patterns of traffic flow through the intersection [9]. In addition to optimising the cycle times of traffic signals, attempts are often made to synchronise the green times of signals at adjacent intersections through an offset in the commencement time of the green signal at adjacent intersections [35], which results in so-called *green-waves*, (groups or platoons of vehicles which are able to proceed along a road through several intersections, all indicating a green signal at their arrival). Helbing [20] describes these synchronisation methods as being completely coercive in that they force the traffic flow to comply with pre-calculated patterns in an attempt to optimise certain criteria associated with travel through a network, such as the total travel time. Due to the fact that traffic demand varies, however, it is widely believed that further improvement of traffic flow requires a more flexible approach to the control of traffic at adjacent signalised intersections [13].

One such example of a more flexible approach to the control of traffic signals is the notion of *self-organisation*. Self-organisation is an optimisation technique inspired by numerous processes which occur in nature. It is a process in which the global level of coordination of a system emerges naturally and solely from interactions among the system's lower-level components. These interactions are governed by sets of rules which are executed using only local information, without any reference to the global pattern [10]. One such example in nature is the organisational abilities of bees. Bees possess the ability to successfully organise complex social interactions without any form of centralised command and control [46]. The brain of a honey-bee is minuscule when compared to that of a human, comprising only approximately one million neurons [10], and



(a) (b)

Figure 1.1: (a) A Wavetronix SmartSensor Adavance^{TM} radar detection unit, (b) mounted on a traffic light at an intersection.

yet bees are able to facilitate community defence, environmental control, food production and manufacture, reproduction and the rearing of their young [46]. Bees achieve this organisation according to a swarm-like response to social interactions and environmental triggers such as predators, rather than being governed by a higher, centralised authority [46]. Other examples of self-organising systems are encountered in many scientific areas, including biology, chemistry, geology, sociology and information technology [40].

Lämmer and Helbing [25] propose an application of the notion of self-organisation to the field of traffic flow optimisation. They describe a decentralised control algorithm, based on short term traffic forecasts, which allows the system to discover for itself the most effective local traffic signal timings at each intersection in the network as a function of the current local traffic situation and to adjust itself accordingly. A consequence of allowing each intersection to achieve local optimisation with respect to traffic through flow is that a "ripple-effect" occurs in the network in such a way that synchronisation among intersections results naturally, rather than attempting to optimise an entire system of traffic signals externally for assumed traffic conditions which are never exactly met. This approach stands in contrast with the optimisation techniques of global coordination which forces traffic in the network to adjust to a set of predefined cycles and timings.

Advances in technology have led to the improvement in traffic control through the introduction of techniques such as vehicle detection, which provides information on the number of vehicles approaching an intersection and, depending on the type of vehicle detection technology used, may provide information concerned with certain characteristics of the approaching vehicles, such as their speeds and distances from the intersection.

The most common type of detector used in South Africa is the inductive loop detector [39], which employs a wire sensor loop embedded within the road pavement. When a vehicle enters into the detection zone of the sensor, it causes a disturbance within the magnetic field of the loop by decreasing its inductance. If the magnitude of this decrease in inductance is above a certain predetermined threshold, it is detected by a loop detector unit which is responsible for monitoring and energising the loop. This loop detector then sends an output signal to the controller unit which is responsible for the implementation of the logic which determines the switching of the traffic signals [32]. It receives signals from the detectors and interfaces with the traffic signals to provide the sequencing and timings of the traffic signal displays.

Another form of vehicle detection technology, is that of radar detection. An example of radar detection equipment may be seen in Figure 1.1. The detection unit in Figure 1.1(a) has been designed and built by a South African company based in Cape Town, called Traffic Management Technologies [41], which provides information on the specifications and performance abilities of

the equipment as summarised in Table 1.1.

When mounted at an intersection, as shown in Figure 1.1(b), this radar detection unit effectively enables the traffic signalling equipment controlling traffic flow through the intersection to observe the traffic approaching the intersection, in terms of the number of vehicles approaching along each lane, as well as their travel speeds and their estimated times of arrival at the intersection. This information may be used as input by the algorithm responsible for the control of the signal timings and phase sequence at the intersection.

Measured quantities	Vehicle presence, speed, range and arrival time
Detection zones	Up to 8 traffic lanes simultaneously
Detection range	30 metres to 152.4 metres (100 feet to 500 feet)
Zone resolution	1.524 metres (5 feet)
Time resolution	2.5 milliseconds
Ambient operating temperature	–40 C° to 75 C°
Humidity	Up to 95% RH
Shock	10 g 10ms half sine wave
Physical dimensions (H × W × D)	32 cm × 23 cm × 7.6 cm (12.6 in × 9.0 in × 3.0 in)
Mass	less than 2.27 kg or 5 lbs

Table 1.1: Wavetronix SmartSensor Adavance^{TM} radar detection equipment specifications [41].

The problem considered in this thesis may be described as attempting to find an answer to the following research question and to motivate this answer scientifically: Are there more efficient traffic control paradigms (using, for example, available radar technology) within the context of a local self-organising and self-coordinating set of traffic light signals, and if so, how effective are these alternative techniques compared to those that are currently implemented?

1.3 Scope and objectives

In order to investigate the effectiveness of self-organising traffic control algorithms in terms of their propensity to minimise driver waiting times and to ease congestion in a traffic network, the following objectives are pursued in this thesis:

1. To *perform* a comprehensive survey of the literature pertaining to the various fields incorporated in this study, including
   (a) the basic theory associated with the dynamics of traffic flow in road traffic networks and at signalised intersections,
   (b) mathematical models which describe the flow of traffic along a road section, both in free-flowing and congested conditions,
   (c) methods of self-organisation and their application to traffic flow optimisation at signalised intersections, and
   (d) various computer simulation modelling techniques and approaches.
2. To *investigate* the efficiency of a number of different traffic control algorithms in a simulated environment for various road network topologies.

This requires

(a) the building of a simulation model capable of replicating real-world traffic flows,

(b) the verification of the simulation model in (a) above to ensure that the model functions correctly and performs to expected standards,

(c) the validation of the simulation model in (a) above to ensure that the model sufficiently mimics real-world traffic flow at signalised intersections, and

(d) the implementation of various traffic signal control algorithms to be tested.

3. To *perform* a real case study in which the performance of proposed self-organising traffic signal control techniques may be compared to currently implemented real-world regimes.

4. To *present* the findings of the simulation study together with an in-depth analysis and interpretation of the results and their consequences.

The traffic simulation model built for the purpose of this study attempts to replicate real-world traffic flows as closely as possible. For this reason, vehicle accelerations and decelerations were incorporated into the model. However, only constant accelerations and decelerations are considered. It is also assumed that the cruising speeds of all vehicles in the system are equal. Finally, all vehicles present in the system are assumed to be of the same dimensions.

1.4 Thesis organisation

Apart from this introductory chapter, this thesis contains five additional chapters. Chapter 2 provides information obtained from the literature in terms of the dynamics of flows through traffic networks and at signalised intersections. It also provides the reader with information about currently implemented traffic control techniques (both fixed-time and demand actuated) at signalised intersections. The chapter closes with a description of an anticipative self-organising traffic control algorithm found in the literature as well as several adaptive self-organising control algorithms.

Chapter 3 serves to provide the reader with information on good practices in computer simulation modelling. The chapter opens with an overview of various types of computer simulation models, as well as the various approaches to computer simulation modelling typically found in the literature. The composition of a simulation model is discussed, together with the steps that are to be followed in a sound simulation study. Both the benefits and disadvantages of using simulation in a research study are touched upon. The chapter closes with a discussion on the applicability of computer simulation modelling to the investigation of traffic flow control in a road network.

The details of the simulation model built for the purpose of this study are described in Chapter 4. The chapter opens with a description of the approaches taken to model the road sections upon which the vehicles in the model travel, the traffic signals responsible for controlling traffic flow at intersections, and the vehicle characteristics, such as speed, acceleration and deceleration. This is followed by a section on the implementation of the model, as well as the performance measures which were considered when evaluating the effectiveness of the various traffic signal control algorithms tested. An investigation into the effects of incorporating vehicle accelerations and decelerations into the model follows. The chapter closes with sections describing the verification and validation processes followed, showing that the simulation model has been built correctly and that the correct model has been built.

Chapter 5 provides information pertaining to the various traffic control algorithms implemented and tested as well as the simulation experimental design followed for each simulation run. The simulation results are presented for each network topology investigated, which includes a single, isolated intersection, a two-by-two traffic signalised street grid structure, a three-by-three traffic signalised street grid structure, and a realistic case study performed on an intersection found along the Adam Tas Road in Stellenbosch, South Africa. Each set of results is followed by an analysis and interpretation of their significance and meaning.

The final chapter of the thesis, Chapter 6, provides a brief summary of the work contained in the thesis, together with recommendations and suggestions as to the applicability and effectiveness of self-organising heuristics with respect to traffic flow control at signalised intersections, based on the simulation study of Chapters 4 and 5. It closes with suggestions towards possible future work related to the application of self-organisation to traffic control.

CHAPTER 2

Traffic Flow Theory and Self-Organising Traffic Control

Contents

Some of the basic principles behind traffic control at signalised intersections, as well as proposed methods for effectively modelling traffic flow along signalised links, are reviewed in this chapter. This is followed by a presentation of currently employed methods of signal control at intersections along South African roads. The chapter closes with a section introducing self-organisation, particularly with application to traffic light control.

2.1 Traffic flow through a network and intersection control

A traffic network is generally made up of roads of different types in terms of their physical attributes and the traffic flow patterns along them. This section contains a review of the different road types and intersections of a traffic network, as well as a mathematical description of vehicle flows along and through them.

2.1.1 Traffic flow on signalised networks

According to a report published by the National Transport Commission [11], urban roads may be divided into three categories. The first category comprises local or residential streets, which provide access to residential areas, and are typically associated with relatively low traffic volumes, with vehicle movement through intersections being controlled by stop and yield signs. The second category includes collector streets and arterials. Collector streets are responsible for collecting and distributing traffic to and from residential areas, while arterials link residential areas with the busier roadways of business districts. Movement through intersections on collector streets and arterials is commonly controlled by stop and yield signs, and automated traffic signals, respectively. The third category comprises business district streets. Traffic moving along business district streets generally maintain lower speeds due to intensive land use patterns in the area as well as larger traffic volumes. Most intersections of streets in a business district are controlled by automated traffic signals.

As traffic assembles on collector streets from local streets en route to more centralised locations, it is likely that all vehicles comprising the traffic flows which merge on arterials have only encountered intersections controlled by stop or yield signs, and not automated traffic signals. Because of this phenomenon, it is unlikely that any noticeable platooning[1] of vehicles will occur, assuming that roads are not congested as an arterial is approached. Hence arrivals to the first intersection controlled by automated traffic signals will typically occur in a (close to) random manner. These arrivals are best approximated by a Poisson distribution [11].

Any vehicles that were stopped, or queued at a signalised intersection depart the intersection as a platoon upon the commencement of a green signal. As this platoon proceeds downstream, it will tend to gradually disperse. If the next intersection encountered by the traffic flow is substantially further than 800 metres downstream from the previous intersection, then vehicle arrivals at the intersection tends away from being uniform and are once again better approximated by a random process [11].

In business districts, intersections are generally closely spaced, which affords the opportunity for coordinating the signal timings of adjacent intersections. Taking into account traffic movement and platoon progression along an arterial, a traffic flow relationship may typically be established among adjacent intersections such that vehicles may be afforded a green band[2] whilst proceeding along an arterial, thus minimising the need for stopping.

2.1.2 Traffic flow at signalised intersections

The most common interruption of vehicle flow along a roadway occurs at an intersection where conflicts with respect to the right of way the right of way arise due to the fact that a common space is shared by several traffic streams [11, 32]. These conflicts may be classified into crossing movements and converging movements. An illustration of these areas of conflict for the various types of movements is shown in Figure 2.1 which depicts an intersection of two two-way streets. In an attempt to reduce the conflict, traffic streams may either be separated in space (*i.e.* by construction of an overpass) or in time (*i.e.* by interrupting each traffic stream via automated traffic signal controls) [32].

[1] A *platoon* is a collection of vehicles which are all travelling at the same speed, with approximately equal following distances between consecutive vehicles in the platoon [34].

[2] A *green band* is a term used to describe the event in which a vehicle receives consecutive green signals at adjacent intersections while travelling along an arterial.

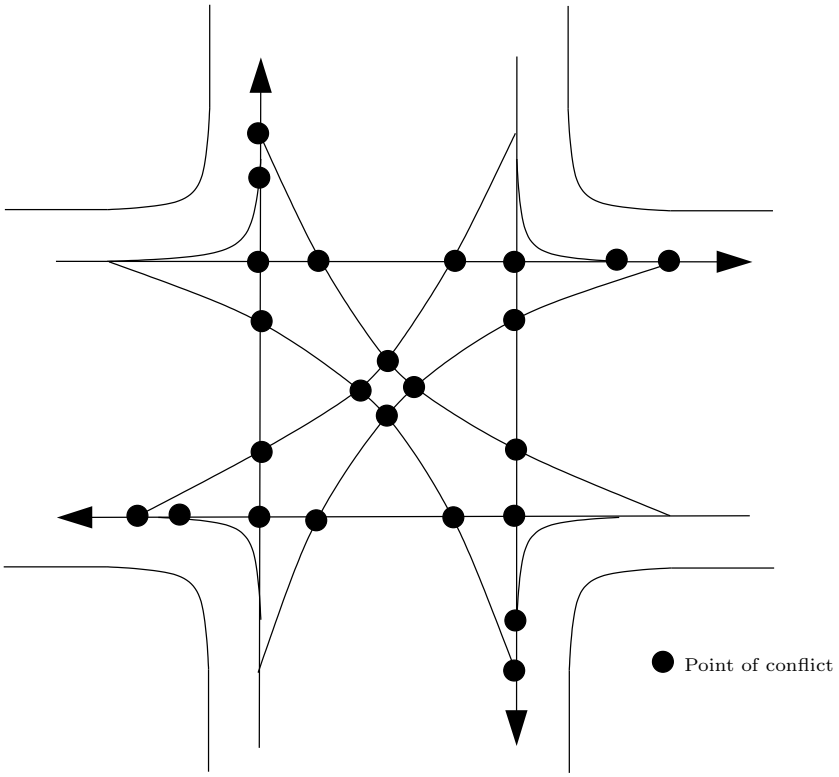


Figure 2.1: Potential conflicts between vehicles at an intersection of two two-lane streets [11].

In the case of controlling traffic flows at an intersection via automated control signals, the most important aspects to consider with respect to the control strategy implemented are the signal timings, which include the *cycle length*, the individual *signal indication* times and the *signal phasing*.

A *signal indication* is defined as a green, red, or amber (and possibly other special indications, such as single, or combined turning arrows) indication which is emitted through the signal lens of a traffic control signal [11, 32]. A *cycle length* is the time period required for one complete sequence of signal indications, and is equal to the sum of its components [11, 32].

Signal phasing is the sequence by which the various movements of vehicles (and pedestrians) are served at an intersection [11, 32] where a *phase* is a part of the cycle during which a green indication is displayed to a particular movement [11].

The main objectives of signal phasing, according to the National Transport Commission [11], is that the right of way of the various phases in the cycle should alternate so as to provide for an orderly and efficient movement of traffic, to minimise average vehicle and pedestrian delays, to reduce accident risk and to maximise the capacity of each intersection approach. Apart from these main objectives, some secondary objectives include coordinating traffic where suitable signal spacing exists, controlling of traffic lane use, providing ramp control at freeway entrances and being able to interrupt traffic for emergency vehicles. These desirable attributes are, however, not always compatible. For example, in order to minimise delay, it is recommended that fewer phases and shorter cycle lengths be employed [11], while increasing the number of phases and their respective lengths promotes safety, but hinders efficiency as it results in greater delays [11, 32]. Thus, a trade-off between risk and delay minimisation is required.

The National Transport Commission [11] suggests that the number of phases required in a signal

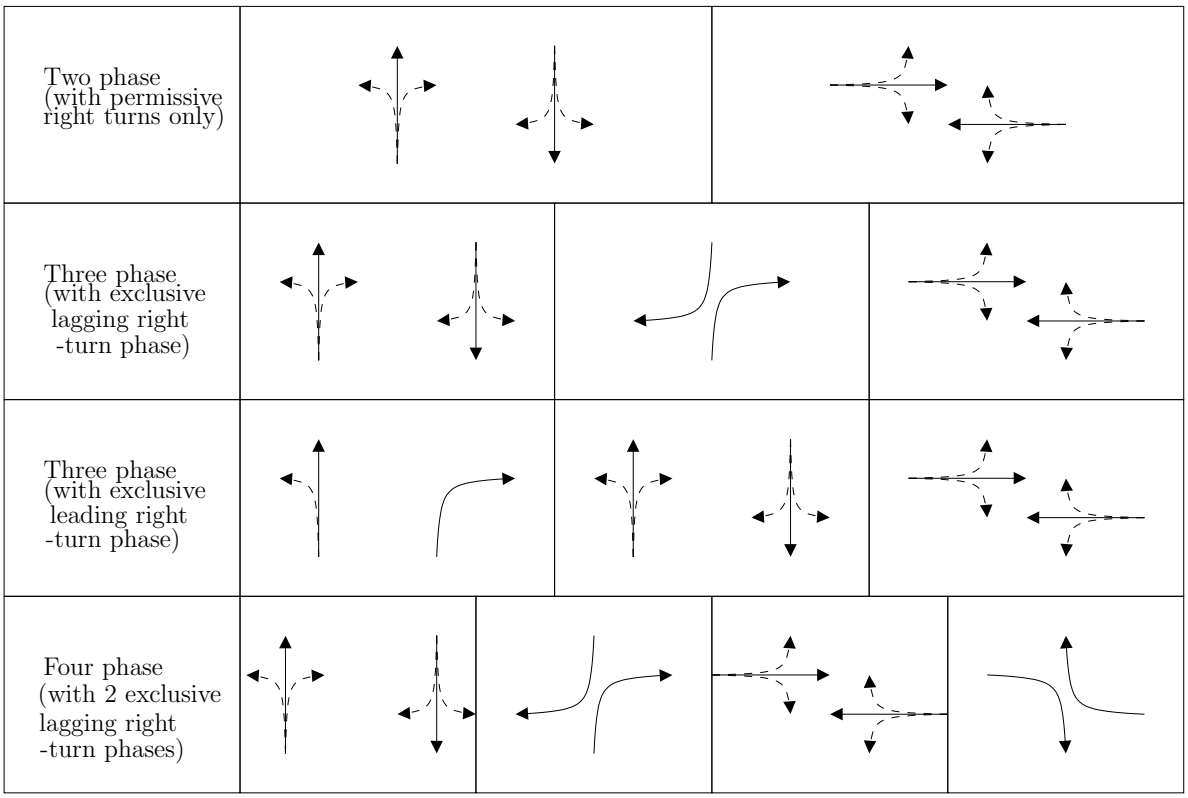


Figure 2.2: Possible phase plans for traffic signal control [11].

plan should primarily depend on the right-turn (left-turn in some countries) demand of vehicles approaching the intersection. Their report goes on to state that there are no limitations on the number of phases that may be utilised, but that it is desirable to use the minimum number which will meet all the objectives required. Some examples of control regimes comprising different phase numbers are shown in Figure 2.2.

Right-hand turning at an intersection may either be permissive, *i.e.* vehicles turning right at the intersection must yield to oncoming traffic, or it may be protected, in which an extra phase is added to the cycle which allows for exclusive right of way for vehicles turning right by stopping all other conflicting traffic flows. The decision as to whether to include an extra exclusive right turning phase in the cycle of a traffic signal control system requires careful consideration. This is because, although the inclusion of an exclusive right turning phase may lessen the risk associated with vehicles turning right across conflicting traffic flows, it also results in greater vehicle delays, due to an increase in cycle lengths and a decrease in the amount of green time available for conflicting traffic flows. Intersection efficiency is further reduced due to increased change intervals (amber and all-red times) between phases. Some examples of existing warrants which should be considered when deciding whether to incorporate an exclusive right turning phase include:

- the cross product of right-turn and opposing through-traffic volumes [23, 43],
- right-turn traffic volumes [1, 43],
- opposing through-traffic volumes [11],
- the right-turn delay [1, 23],

- the right-turn volume to capacity ratio [38],
- the number of right-turn related accidents [37, 38],
- the number of right-turn conflicts [1, 23, 38],
- the speed of the opposing through-traffic, [43],
- pedestrian interference [37],
- intersection geometry [37], and
- sight distance [11].

2.1.3 Vehicle delays at traffic signals

As was mentioned earlier in the chapter, the main source of interruptions and hence delay to traffic flows along road sections is signalised intersections. This is because each traffic stream approaching a signalised intersection only receives service for a fraction of the signal's control cycle, during which vehicles belonging to that particular traffic stream may proceed through the intersection. For the rest of the control cycle the vehicles are required to wait at the intersection for service. However, apart from the delay that vehicles experience while waiting for service, there is also a delay associated with a queue of vehicles as it discharges from a queue upon receiving service. An analysis of vehicle actions and movements, as well as their associated delay times, upon the commencement of a green signal is presented below, according the National Transport Commission's report [11].

Consider Figure 2.3, which shows a stationary queue of vehicles awaiting service at a signalised intersection. Upon receiving service, the vehicles will begin to depart from the intersection. To investigate the delays incurred by vehicles as they depart from rest, the headways between consecutive vehicles as they cross the stop line of an intersection are considered.[3] When observing vehicles departing from rest at a signalised intersection, Greenshields *et al.* [18] observed that the headways of the first several vehicles in the queue are relatively longer than those of the vehicles that follow them. These longer headways may be motivated as follows. Upon commencement of the green signal, the driver of the first vehicle in the queue must observe the signal change, react to it, and then accelerate through the intersection. This results in a relatively long headway. The second vehicle in the queue follows similarly, although its driver's reaction and acceleration period partially overlaps with that of the first vehicle in front of it. It will also be travelling at a greater speed than the first vehicle as it crosses the stop line as it has had an extra vehicle length over which to accelerate. Thus, the resulting headway of the second vehicle will still be comparatively long, but it will be shorter than that of the first vehicle. This observation holds for the next vehicle in the queue, and so on. However, each consecutive vehicle achieves a shorter headway than the previous one until, after some number of vehicles, N (say), the effect of driver reaction and acceleration on vehicle headways has dissipated, as may be seen in Figure 2.4.

In Figure 2.4, the average headway achieved by vehicles after the N^{th} vehicle in the queue is denoted by hw. The headways of the first N vehicles, on average, exceed hw, and are expressed

[3]The *headway* between two vehicles is defined as the time elapsed between the crossing of the stop line of the rear of the first vehicle, and the crossing of the rear of the vehicle following it. In the case of the first vehicle in a queue, the headway is given by the time elapsed between the commencement of the green signal and the crossing of the stop line of the rear of the vehicle.

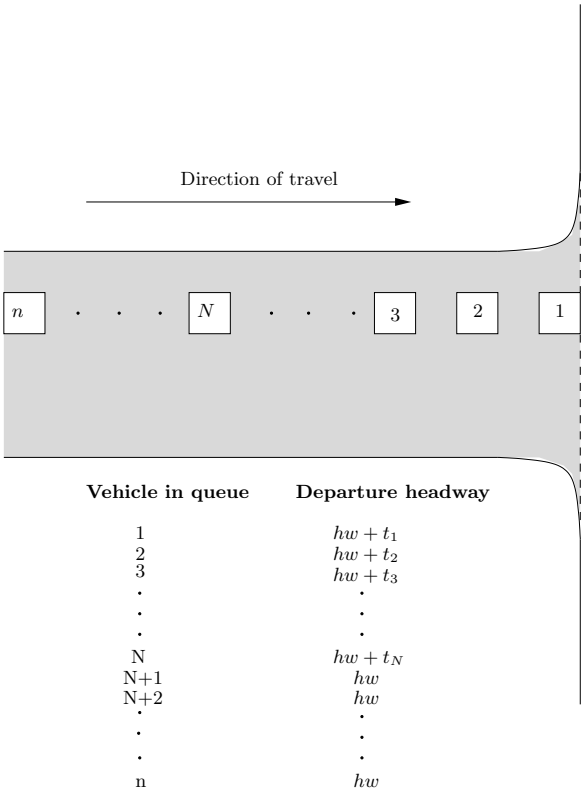


Figure 2.3: Queue departure conditions at a signalised intersection.

as $hw + t_i$, where t_i is the marginal headway associated with vehicle i. The value of t_i decreases as the value of i increases from 1 to N. The quantity hw is called the *saturation headway* [42], and is used in calculating the *saturation flow rate* s of a road section, where s represents the number of vehicles per hour that may pass through an intersection when the saturation headway hw occurs between all vehicles, and is given by

$$s = \frac{3\,600}{hw}.$$

In Figure 2.3, the marginal headway values, $t_1, \ldots, t_N$ are also called *start-up lost times* [42], and their sum represents the total start-up lost time L_1 of vehicle 1 to vehicle N in the queue, *i.e.*

$$L_1 = \sum_{i=1}^{N} t_i.$$

Thus, each time a queue of N or more queued vehicles receives a green signal, the total amount of time lost, is a sum of hw seconds per vehicle and L_1.

A second type of lost time is experienced by all vehicles on a road section between traffic signal phases. This occurs because safety requirements call for a clearance period between all phase changes during which no vehicle is supposed to enter the intersection. This time gap between phases is called as the *clearance interval* [11] and generally consists of a period of amber and all red signal indications. However, drivers do not always observe the entire clearance interval, and may pass through the intersection during the early stages of the amber signal indication. Thus, according to [11], the clearance interval consists of two parts, namely an amber effective green part, during which vehicles may proceed through the intersection, and the part where vehicles

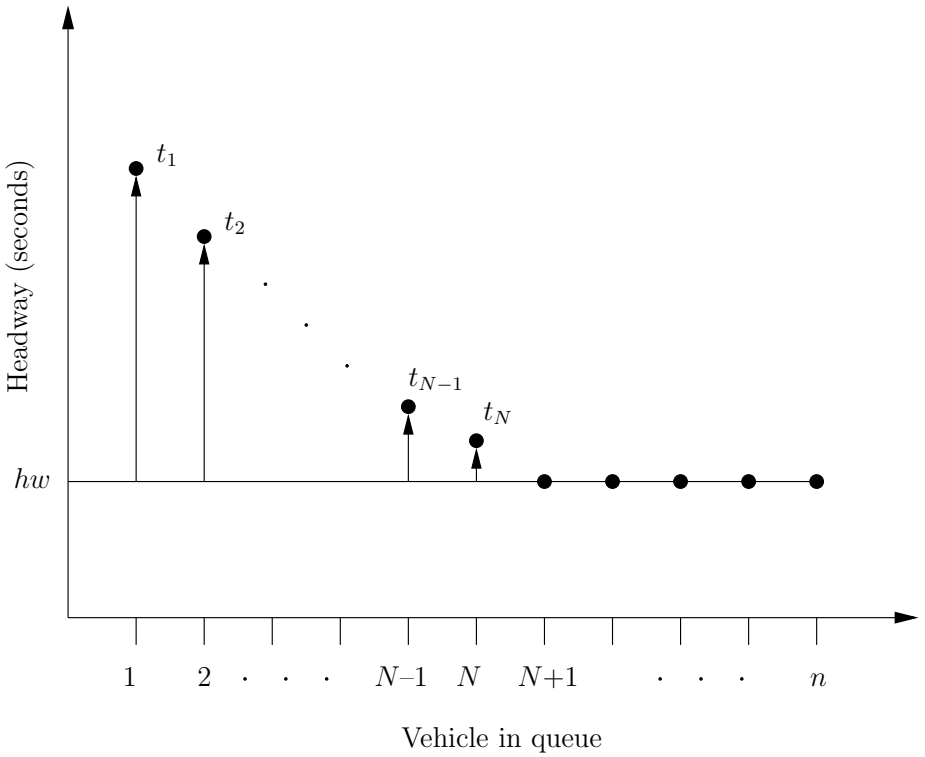


Figure 2.4: Saturation headway and start-up lost times at a signalised intersection.

are required to stop. The time associated with this second part is denoted by L_2, and is called the *clearance lost time* [11].

Approaches towards modelling vehicle delays at intersections analytically are presented by Allsop [2], and are discussed further later in this thesis.

2.2 Currently employed traffic control techniques

An automated traffic signal responsible for controlling traffic at an intersection may be classified as either pre- or fixed-timed, or demand actuated [32]. Fixed-timed signals typically repeat a predetermined constant signalling cycle, the length and composition of which may vary for different periods of the day, depending on traffic demand. Demand actuated signals, on the other hand, have the ability to respond to current traffic conditions through the use of vehicle detectors. Examples of both instances are provided in this section and discussed below.

2.2.1 Fixed-timed traffic signal control

Vermeulen [45] lists the following techniques which may be employed in determining the amount of green time made available to each approach of an intersection under fixed-time control:

- Attempt to provide the minimum required green time to side streets along an arterial, determined by pedestrian constraints, with the remainder of the available green time being allocated to the arterial approaches.

- Divide the available green time among the various approaches proportionally according to the critical traffic volumes along each approach.
- Divide the green time among the various approaches proportionally according to the critical lane traffic volumes along each approach.
- Distribute the available green time in such a way that the volume to saturation flow ratio on all the approaches is equal. This is also known as the *British technique.*
- Distribute the available green time proportionally according to the total traffic volumes on the approaches, weighted by a function of the saturation flow, so as to attempt to minimise the total delay at the intersection.

2.2.2 Demand actuated traffic signal control

Demand actuated control may be divided into two further subcategories, namely semi-actuated control and full-actuated control. Semi-actuated control is typically implemented at the intersection of a major and a minor street, where a green signal is presented to traffic along the major streets unless a vehicle is detected along the minor streets, or when a pedestrian needs to cross the major street [32]. Full-actuated control employs vehicle detectors along all lanes of all the approaches of an intersection carrying varying, but approximately equal traffic volumes [32].

An example of a full-actuated control system which has been implemented widely around the world, and is used in South Africa, is SCOOT. The *split-cycle-offset-optimisation technique*, or SCOOT, was conceived in the early 1970s, and as of the late 1990s, more than 170 implementations of SCOOT have taken place internationally [36]. SCOOT is an adaptive traffic control system which "coordinates the operation of all the traffic signals in an area to give good progression to vehicles through the network. Whilst coordinating all the signals, it responds intelligently and continuously as traffic flow changes and fluctuates throughout the day" [39].

The operation of SCOOT is based on cyclic flow profiles which are typically measured by inductor loops or other sensory devices [36]. Information from the detectors is used as input to the SCOOT model, which is responsible for modelling vehicle progression between a sensor, which is usually positioned upstream along a road link, and the stop line of the road link. As vehicles pass the detector, they generate data that are received by the SCOOT model which converts the data into internal units and uses them to construct the aforementioned cyclic flow profiles for each link approaching the intersection. These data are then utilised by the SCOOT model in the form of three optimisers, which continuously adapt three traffic control parameters of the intersection signals. These are the amount of green time afforded to each approach at each intersection (known as the *split*), the time between the indication of a green signal between adjacent intersections (known as the *offset*), and finally, the cycle length of the traffic signals at each signalised intersection. The split optimiser is run at every stage change, while the offset optimiser is called for each signal cycle for every intersection, and the cycle time for each region is optimised once every five minutes or once every two and a half minutes, when required, so as to respond to rapid flow changes.

One of the main advantages of SCOOT is that the signal timings evolve as the traffic situation changes without any of the disruptions generally associated with changing fixed time control plans of more traditional urban traffic control systems.

2.3 Modelling traffic flow on signalised links

The dynamics of traffic flow, both along road sections and at the intersections they link, are considered in this section. A traffic network may be modelled by a directed graph in which the edges represent homogeneous road sections and the vertices represent intersections. This section is devoted to a description of the work by Lämmer and Helbing [25], Lämmer, Donner and Helbing [24], and Helbing, Lämmer and Lebacque [20] on the mathematical modelling of traffic flow along homogeneous road sections. Although a macroscopic approach is adopted with respect to modelling traffic flow, the ideas and philosophies behind them are also applicable to a microscopic perspective when modelling traffic flows.

2.3.1 Homogeneous Road Sections

The literature describes a traffic flow network as comprising links, representing homogeneous road sections, and nodes, representing intersections. Associated with each homogeneous road section i, is a time invariant length, L_i, an arrival rate Q_i^{arr}, a departure rate, Q_i^{dep}, and a saturation flow rate $Q_i^{\max}$. The arrival rate represents the number of vehicles entering the road section per unit time along all of its lanes and is such that $Q_i^{\text{arr}} \leq Q_i^{\max}$. Similarly, the departure rate represents the number of vehicles leaving the road section per unit time along all of its lanes, where $Q_i^{\text{dep}} \leq Q_i^{\max}$.

The dynamics along a link, between its entry and exit points, in a network are described by a section-based theoretical queueing model due to Helbing [19], which is directly related to the equation of vehicle conservation proposed by Lighthill and Whitham [28]. The average *velocity* of vehicles at some position x along the road section at time t is given by $\boldsymbol{V}_i(x,t) = V_i(x,t)\boldsymbol{i}$, where $\boldsymbol{i}$ is a unit vector in the positive direction of traffic flow. If the traffic *density* (the number of vehicles per unit of road length) at the same point and time is given by $k_i(x,t)$, then the traffic *flow rate* is the product $Q_i(x,t) = k_i(x,t)\,V_i(x,t)$.

Following the philosophies proposed by Helbing *et al.* [20] with respect to the spacing between vehicles along a homogeneous road section, it is assumed that vehicles follow each other in a manner such that the lower their velocity, the smaller their following distance and *vice versa*. Following distances are smallest when vehicles are stationary, *i.e.* when $V_i(x,t) = 0$, in which case they are a minimum safety front-bumper-to-rear-bumper distance apart. The space occupied by a vehicle (*i.e.* the vehicle length together with the minimum safety front-bumper-to-rear-bumper distance) in this stationary situation is given by $1/k^{\text{jam}}$, where k^{jam} denotes the maximum traffic density (*i.e.* the largest number of stationary vehicles per space unit on the road section. The following distance of a vehicle is assumed to grow linearly with the speed of the vehicle in front of it. A safe following distance is assumed to be $TV_i(x,t)$ where T is a safe time gap or reaction time maintained between consecutive vehicles. The space occupied per vehicle may therefore be represented by an *effective vehicle length*, $\ell^{\text{eff}} = 1/k_i(x,t) = 1/k^{\text{jam}} + TV_i(x,t)$. Vehicles are assumed to move as fast as possible without violating the safe time gap or the speed limit $V_i^0 = ||\boldsymbol{V}_i^0||$ associated with road section i.

Traffic is considered to be *free-flowing* if the vehicles are travelling at the speed limit, *i.e.* when $V_i(x,t) = V_i^0$, or in terms of the average space occupied per vehicle, if $1/k_i(x,t) \geq 1/k^{\text{jam}} + TV_i^0$. Traffic is considered to be *congested* otherwise. The flow rate of traffic may be approximated

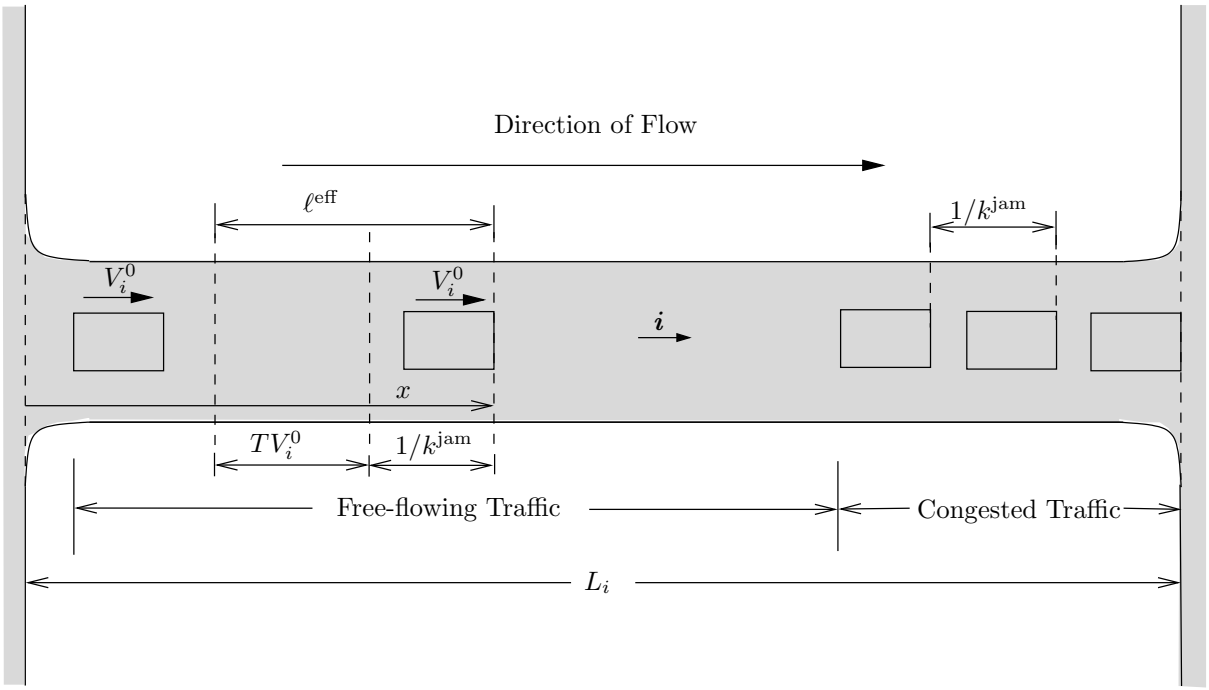


Figure 2.5: Graphical illustration of homogeneous road section traffic flow model.

by the flow-density relationship

$$Q_i(x,t) = \begin{cases} k_i(x,t)V_i^0 & \text{in free-flowing traffic,} \\ \frac{1}{T}[1 - k_i(x,t)/k^{\text{jam}}] & \text{in congested traffic.} \end{cases} \tag{2.1}$$

The road section model of traffic flow described above is illustrated graphically in Figure 2.5 and depends on three intuitive parameters: the maximum jam density k^{jam}, the speed limit V_i^0 or speed of free-flowing traffic on road section i, and the following distance coefficient of proportionality or safe time gap between vehicles in congested traffic, T.

2.3.2 Influences on the congestion front

The section-based model of traffic flow in §2.3.1 is based on two characteristic velocities [20]. The first is the velocity $\boldsymbol{V}_i^0$ at which free-flowing traffic propagates downstream, and the second is the velocity $\boldsymbol{c} = \frac{-1}{Tk^{\text{jam}}}\boldsymbol{i}$ at which the congestion front (the boundary between congested and free-flowing traffic) propagates upstream.

All relevant quantities of the model may be determined from its boundary flow rates. For example, the traffic dynamics in the interior of road section i at time t may be derived from the arrival flow rate $Q_i^{\text{arr}}(t)$ at the upstream end of the road section and the departure flow rate $Q_i^{\text{dep}}(t)$ at the downstream end of the road section together with the two characteristic velocities $\boldsymbol{V}_i^0$ and $\boldsymbol{c}$.

The interior traffic flow rate at time t and position x is given by

$$Q_i(x,t) = \begin{cases} Q_i^{\text{arr}}\left(t - \frac{x}{V_i^0}\right) & \text{if } 0 \le x < L_i - p_i(t) \text{ (in free-flowing traffic)} \\ Q_i^{\text{dep}}\left(t - \frac{L_i - x}{|\boldsymbol{c}|}\right) & \text{if } L_i - p_i(t) \le x \le L_i \text{ (in congested traffic),} \end{cases}$$

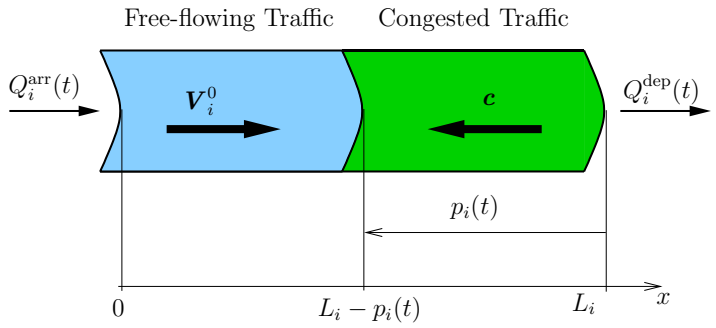


Figure 2.6: A road section i of length L_i with a subsection of length p_i of congested traffic at the downstream end.

where p_i is the length of the subsection of L_i for which traffic is congested as illustrated in Figure 2.6, while the traffic density at time t and position x is given by

$$k_i(x,t) = \begin{cases} Q_i(x,t)/V_i^0 & \text{if } x < L_i - p_i(t) \text{ (in free-flowing traffic)} \\ \left[1 - TQ_i(x,t)\right] k^{\text{jam}} & \text{if } 0 \leq L_i - p_i(t) \leq x \leq L_i \text{ (in congested traffic).} \end{cases}$$

Finally, the average traffic velocity at time t and position x may be calculated via the formula

$$\boldsymbol{V}_i(x,t) = \frac{Q_i(x,t)\boldsymbol{i}}{k_i(x,t)},$$

if $k_i(x,t) > 0$.

2.3.3 Movement of the congestion front

The downstream end of the congestion front remains at $x = L_i$ while the upstream end lies at $x = L_i - p_i(t)$, where discontinuities of magnitudes Δk_i and ΔQ_i occur in the density and flow, respectively [20]. In order to ensure the conservation of flow of vehicles, the condition $\Delta Q_i = -\Delta k_i \cdot \frac{\mathrm{d}p_i}{\mathrm{d}t}$ must hold. Thus, from the work of Helbing [19], the boundary between congested and free flowing traffic may be seen to move with a velocity of

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{Q_i^{\text{arr}}(t - [L_i - p_i(t)]/V_i^0) - Q_i^{\text{dep}}(t - p_i(t)/|\boldsymbol{c}|)}{k_i^{\text{arr}}(t - [L_i - p_i(t)]/V_i^0) - k_i^{\text{dep}}(t - p_i(t)/|\boldsymbol{c}|)}. \tag{2.2}$$

Note that in the congested region of length $p_i(t)$ the maximum possible flow rate $Q_i^{\max}$ is

$$Q_i^{\max} = \left(T + \frac{1}{V_i^0 k^{\text{jam}}}\right)^{-1}, \tag{2.3}$$

which corresponds to a vehicle accelerating out of the congested region every T seconds. It should be noted that this rate of $1/T$ is never actually achieved due to the fact that each subsequent vehicle has to travel an additional distance $\ell^{\text{eff}} = 1/k^{\text{jam}}$ in order to reach the stop line of the road section, corresponding to an additional travel time of ℓ^{eff}/V_i^0.

If the departure rate is zero due to a red signal, then (2.2) becomes

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = \left[\frac{k^{\mathrm{jam}}}{Q_i^{\mathrm{arr}}(t - [L_i - p_i(t)]/V_i^0)} - \frac{1}{V_i^0}\right]^{-1} \approx \frac{Q_i^{\mathrm{arr}}(t - [L_i - p_i(t)]/V_i^0)}{k^{\mathrm{jam}}}. \tag{2.4}$$

If a green signal is displayed at time t_0', the boundary of the congested region still propagates upstream at the speed described by (2.4) due to newly arriving vehicles. However, a section of quasi-free flowing traffic moving at the maximum flow rate $Q_i^{\max}$ propagates upstream at a speed of $|\boldsymbol{c}|$ due to vehicles leaving road section i at the downstream boundary. The effective length $p_i^{\mathrm{eff}}(t)$ of the vehicle queue is thus given by

$$p_i^{\mathrm{eff}}(t) = p_i(t) - |\boldsymbol{c}|(t - t_0').$$

If this effective queue has been resolved at time t^*, *i.e.* $p_i^{\mathrm{eff}}(t^*) = 0$, it will take an additional time of $p_i(t^*)/V_i^0$ for the last vehicle of that queue to leave the road section. Therefore $p_i(t) = 0$ at time $t^* + p_i(t^*)/V_i^0$.

Lämmer and Helbing [25] showed that, when considering the actual number of vehicles present on road section i, it may be seen that in free flowing traffic the cumulative number $N_i^{\mathrm{exp}}(t)$ of vehicles expected to reach the congested region boundary of road section i until time t after being detected by an upstream detection device may be represented by the integral of the arrival rate $Q_i^{\mathrm{arr}}(t - [L_i - p_i(t)]/V_i^0)$ over time, where the time shift $L_i - p_i(t)/V_i^0$ represents the time required to traverse the region of free-flowing traffic along road section i. Thus, $N_i^{\mathrm{exp}}(t)$ is defined as

$$N_i^{\mathrm{exp}}(t) = N_i^{\mathrm{exp}}(t_0) + \int_{t_0}^{t} Q_i^{\mathrm{arr}}\left(t' - [L_i - p_i(t)]/V_i^0\right) \mathrm{d}t'. \tag{2.5}$$

In the case of congestion, however, the number of vehicles that have actually left the road section at its downstream end is given by the integral of the departure rate,

$$N_i^{\mathrm{dep}}(t) = N_i^{\mathrm{dep}}(t_0) + \int_{t_0}^{t} Q_i^{\mathrm{dep}}(t')\,\mathrm{d}t'. \tag{2.6}$$

The difference between $N_i^{\mathrm{exp}}(t)$ in (2.5) and $N_i^{\mathrm{dep}}(t)$ in (2.6) directly corresponds to the number of delayed vehicles, or the queue length, $n_i(t)$ [25]. Consequently, the accumulated waiting time $w_i(t)$ of all vehicles on road section i up to time t increases at the rate

$$\frac{\mathrm{d}w_i}{\mathrm{d}t} = n_i(t) = N_i^{\mathrm{exp}}(t) - N_i^{\mathrm{dep}}(t). \tag{2.7}$$

Three different influencing states are identified in terms of the dynamics of the departure of vehicles from intersections controlled by traffic lights:

1. If the signal is red, the vehicle flow rate at the intersection is zero.

2. When the signal has switched to green, the vehicle queue discharges at the saturation flow rate $Q_i^{\max}$.

3. If the signal remains green once the queue has been dissipated, then vehicles leave the downstream end of road section i at the same rate at which they entered the upstream end of the road section, *i.e.* at a rate of $Q_i^{\mathrm{exp}}(t) = Q_i^{\mathrm{arr}}\left(t - [L_i - p_i(t)]/V_i^0\right)$.

With the introduction of a permeability factor $\rho_i(t)$ [25], which alternates between 0 and 1 corresponding to a red and green traffic signal respectively, it is possible to derive a state-dependent expression for the departure rate, given by

$$Q_i^{\text{dep}}(t) = \begin{cases} 0 & \text{if } \rho_i(t) = 0 \\ Q_i^{\max} & \text{if } \rho_i(t) = 1 \text{ and } n_i(t) > 0 \\ Q_i^{\text{exp}}(t) & \text{if } \rho_i(t) = 1 \text{ and } n_i(t) = 0. \end{cases} \tag{2.8}$$

Analogously, the temporal evolution of the number of vehicles in the queue is given by

$$\frac{\mathrm{d}n_i}{\mathrm{d}t} = \begin{cases} Q_i^{\text{exp}}(t) & \text{if } \rho_i(t) = 0 \\ Q_i^{\text{exp}}(t) - Q_i^{\max}(t) & \text{if } \rho_i(t) = 1 \text{ and } n_i(t) > 0 \\ 0 & \text{if } \rho_i(t) = 1 \text{ and } n_i(t) = 0. \end{cases} \tag{2.9}$$

2.4 Self-Organising Traffic Light Control

Serugendo *et al.* [40] define a self-organising system as follows:

> "A self-organising system functions without central control, and through contextual local interactions. Components achieve a simple task individually, but a complex collective behaviour emerges from their mutual interactions. Such a system modifies its structure and functionality to adapt to changes to requirements and to the environment based on previous experience."

For the application of self-organisation to traffic control, the work of Lämmer and Helbing [25], and Lämmer, Donner and Helbing [24] was researched. In these papers an anticipative approach to self-organising traffic control is suggested. Gershenson [15], on the other hand, suggests an adaptive approach towards self-organising traffic control.

2.4.1 Anticipative self-organising traffic control

The approach adopted by Lämmer and Helbing [25] with respect to self-organising traffic control assumes a priority-based control of the traffic signals at an intersection, determined by the vehicle flows approaching the intersection, taking into account anticipated vehicle flows. The heuristics on which their work is based were inspired by observations of self-organised oscillations of pedestrian flows at bottlenecks.

The service process and set up times

The anticipation of vehicle flows and their associated waiting times, as well as the green time required for their clearance, are essential to the efficiency of flexible traffic light control [25]. Before the methods employed to anticipate vehicle waiting times and required green times are reviewed, the notions of a *service process* and *setup time* associated with each traffic flow approaching an intersection is introduced.

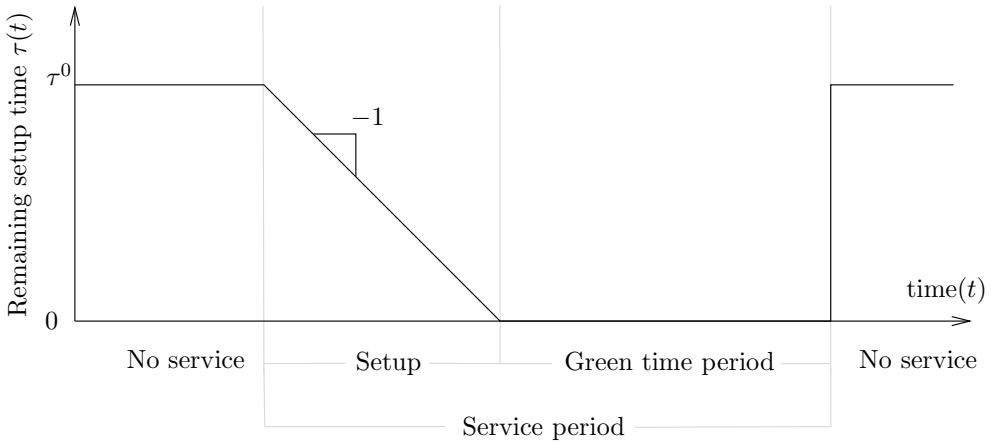


Figure 2.7: The remaining setup time $\tau(t)$ during one traffic signal cycle.

The safe operation of traffic signal control requires that between each phase of service in the cycle of a traffic control system, there exists a period in which all conflicting traffic flows are stopped, allowing for the intersection area to be cleared of any vehicles which may be present in it. This concept is referred to in the literature [24, 25] as a *setup time.* Thus, if some traffic flow i is to receive service at an intersection, a green signal is not displayed before the corresponding setup (or intergreen) time, denoted by τ_i^0, has elapsed. The value of τ_i^0 is chosen according to various safety considerations, and usually lies within the range of 3 to 8 seconds [25]. In the literature, τ_i^0 is responsible for reflecting any time losses due to driver reactions and finite accelerations associated with discharging a queue from rest, as described in §2.1.3. The entire service process may then be seen as comprising three successive states: the setup, the clearing of the queue, and the green time extension (*i.e.* the amount of time the signal remains green once the queue has been cleared). The value of τ_i^0 is shown graphically in Figure 2.7 for each of the states of the service process.

The green time required to clear a queue

For a queue to be completely cleared, the green time allocated to an approach must be sufficient so as to clear both the vehicles that are currently queued along it at the start of its respective service process, as well as the vehicles that are expected to join the queue during the setup phase and while the queue is discharging. This required green time is denoted by $\hat{g}$. Thus, if the service process commences at time t for approach i with remaining setup time $\tau_i(t)$ and anticipated green time $\hat{g}_i(t)$, the queue of vehicles along it will be cleared at time $t+\tau_i(t)+\hat{g}_i(t)$. During the clearing phase of the service process, which has length $\hat{g}_i$, vehicles leave the road section at the maximum flow rate, *i.e.* at a rate of $Q_i^{\max}$. The number of vehicles that are expected to be served during the clearing of the queue at time t is denoted by $\hat{n}_i(t)$ and is equal to $\hat{g}_i(t)Q_i^{\max}$, while the total number of vehicles that would have departed the intersection at the time $t+\tau_i(t)+\hat{g}_i(t)$ is equal to $N_i^{\text{dep}}(t)+\hat{g}_i(t)Q_i^{\max}$. In the literature [24, 25], a queue is defined to be cleared if the number of vehicles that have departed the road section are equal to the number that have arrived at the stop line. Thus, the amount of green time required to clear a queue must satisfy

$$N_i^{\text{dep}}(t)+\hat{g}_i(t)Q_i^{\max}=N_i^{\text{exp}}(t+\tau_i(t)+\hat{g}_i(t)), \tag{2.10}$$

where $\hat{g}_i(t)$ is the largest possible solution to (2.10). The value of $\hat{n}_i(t)$, captures all those vehicles already waiting in the queue, joining the queue during the setup or clearing phases, and arriving as a platoon immediately after the queue is cleared.

Lämmer *et al.* [24], provide a detailed analysis of the temporal evolution of the required green time of an approach, as a function of the current state of the vehicle queue and the expected arrivals to the queue, $Q_i^{\text{exp}}(t)$. As an initial step, (2.5) and (2.6) are substituted into (2.10), resulting in

$$N_i^{\text{dep}}(t_0) + \int_{t_0}^{t} Q_i^{\text{dep}}\left(t'\right) \mathrm{d}t' + \hat{g}_i(t) Q_i^{\max} = N_i^{\text{exp}}(t_0) + \int_{t_0}^{t+\tau_i(t)+\hat{g}_i(t)} Q_i^{\text{exp}}\left(t'\right) \mathrm{d}t'. \tag{2.11}$$

Applying the time derivative $\mathrm{d}/\mathrm{d}t$ to both sides of (2.11), it follows by the fundamental theorem of the calculus that

$$Q_i^{\text{dep}}(t) + \frac{\mathrm{d}\hat{g}_i}{\mathrm{d}t} Q_i^{\max} = Q_i^{\text{exp}}(t + \tau_i(t) + \hat{g}_i(t)) \left(1 + \frac{\mathrm{d}\tau_i}{\mathrm{d}t} + \frac{\mathrm{d}\hat{g}_i}{\mathrm{d}t}\right). \tag{2.12}$$

Finally, the term $\mathrm{d}\hat{g}_i/\mathrm{d}t$ may be separated in (2.12) (replacing $\mathrm{d}\tau_i/\mathrm{d}t$ by $\dot{\tau}_i$), leaving

$$\frac{\mathrm{d}\hat{g}_i}{\mathrm{d}t} = \frac{(1+\dot{\tau}_i) Q_i^{\text{exp}}(t^*) - Q_i^{\text{dep}}(t)}{Q_i^{\max} - Q_i^{\text{exp}}(t^*)}, \tag{2.13}$$

where $t^* = t + \tau_i(t) + \hat{g}_i(t)$, which provides a general expression for the temporal evolution of the predicted green time required to dissolve a queue. It should be noted, however, that, in addition to the departure rate Q_i^{dep}, the remaining setup time $\tau_i(t)$ is also state dependent, as may be seen in Figure 2.7. The state-dependent evolution of the green time $\hat{g}_i(t)$ required for clearing a given queue may be modelled as

$$\frac{\mathrm{d}\hat{g}_i}{\mathrm{d}t} = \begin{cases} \frac{Q_i^{\text{exp}}(t^*)}{Q_i^{\max} - Q_i^{\text{exp}}(t^*)}, & \text{if “no service”, since } Q_i^{\text{dep}}(t) = 0 \text{ and } \dot{\tau} = 0 \\ 0, & \text{if “setup”, since } Q_i^{\text{dep}}(t) = 0 \text{ and } \dot{\tau} = -1 \\ -1, & \text{if “clearing the queue”, since } Q_i^{\text{dep}}(t) = Q_i^{\max} \text{ and } \dot{\tau} = 0 \\ 0, & \text{if “extension of service”, since } Q_i^{\text{dep}}(t) = Q_i^{\text{exp}} \text{ and } \dot{\tau} = 0. \end{cases} \tag{2.14}$$

From (2.13) above, it may be seen that as long as service has not yet commenced, the green time required to clear a queue $\hat{g}_i(t)$ increases continuously, and may even jump to a higher value if $Q_i^{\text{exp}}(t^*) = Q_i^{\max}$. This corresponds to the situation where the arrival of a platoon of vehicles is expected at time t^*. In this situation, the magnitude of the jump is directly proportional to the size of the platoon. Also, between the jump at time t and the arrival of the first vehicle of the platoon responsible for the jump at time t^*, there is exactly as much time left as required to perform the setup, to serve the currently queued vehicles, and to serve all other vehicles arriving before the platoon. This means that if service is initiated by a jump in $\hat{g}_i(t)$, the corresponding platoon will be served in a *green-wave*[4] manner without any delay.

During the setup time, the nominator of (2.13) disappears, resulting in $\mathrm{d}\hat{g}_i/\mathrm{d}t = 0$. This means that the value of $\hat{g}_i(t)$ computed above does not change during the setup. While the queue is being cleared, $\mathrm{d}\hat{g}_i/\mathrm{d}t = -1$ Hence, for each time unit elapsed while the queue is being cleared,

[4] A green-wave is a term used to describe the event in which a platoon of vehicles is able to pass through an intersection indicating a green signal without having to reduce their speed.

the corresponding remaining green time required to clear the queue decreases by one unit per time unit. Thus, for both the setup and clearing phases it may be seen that the time t^* at which the queue was predicted to be cleared remains constant.

Finally, during the period of green time extension, the nominator of (2.13) disappears. However, when a platoon arrives at maximum flow rate $Q_i^{\exp}(t^*) = Q_i^{\max}$, the denominator may also become zero, resulting in a jump in $\hat{g}_i(t)$, causing a transition to the "clearing the queue" state, indicating that the process of serving a platoon and the process of clearing a queue both result in the same temporal evolution of the anticipated green time.

Anticipation of vehicle waiting times

Waiting time is perhaps the most important quantity to consider when attempting to optimise traffic control. The waiting time $w(t)$ of all vehicles up to time t is defined in the literature [24, 25] as the integral of the queue length $n(t)$, that is

$$w(t) = w(t_0) + \int_{t_0}^{t} n(t')\ \mathrm{d}t'. \tag{2.15}$$

Lämmer *et al.* [24] provide a method for determining the expected waiting time of vehicles along an approach which is similar to that described earlier for the calculation of the expected required green time. This method is presented below and attempts to predict the total waiting time $\hat{w}_i(t)$ of all vehicles up to the end of the following service period.

Assuming that the service process commences (or continues) at the current time t, then $\hat{w}_i(t)$ may be defined as the sum of the waiting time experienced by vehicles along road section i up to time t, given by (2.15), the waiting time experienced during the setup process, given by $A(t)$, and the waiting time experienced while the queue is cleared, given by $B(t)$, where

$$A(t) = \int_{t}^{t+\tau_i(t)} N_i^{\exp}(t') - N_i^{\mathrm{dep}}(t)\ \mathrm{d}t'^{5}$$

and

$$B(t) = \int_{t+\tau_i(t)}^{t+\tau_i(t)+\hat{g}_i(t)} N_i^{\exp}(t') - \left[N_i^{\mathrm{dep}}(t) + (t' - (t+\tau_i(t)))Q_i^{\max}\right] \mathrm{d}t'.$$

As was the case when determining the temporal evolution of the required green time, the time derivative of $\hat{w}_i(t)$ is the sum of the time derivatives of its terms,

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \frac{\mathrm{d}w_i}{\mathrm{d}t} + \frac{\mathrm{d}A}{\mathrm{d}t} + \frac{\mathrm{d}B}{\mathrm{d}t},$$

which results in the expression

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \hat{g}_i(t)Q_i^{\max}(1 + \dot{\tau}_i(t)) - Q_i^{\mathrm{dep}}(t)(\tau_i(t) + \hat{g}_i(t)). \tag{2.16}$$

The rate at which the predicted waiting time increases during each state of the queue may be found by substituting the specific values of $Q_i^{\mathrm{dep}}(t)$, $\tau_i(t)$, and $\dot{\tau}_i(t)$ into (2.16).

[5] The outflow is zero during the setup process, which causes $N_i^{\mathrm{dep}}(t)$ to remain constant with respect to t'.

In the no service state ($Q_i^{\text{dep}}(t) = 0$, $\tau_i(t) = \tau_i^0$, and $\dot{\tau}_i = 0$), equation (2.16) simplifies to

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \hat{g}_i(t) Q_i^{\max}. \tag{2.17}$$

Here, $\hat{n}_i(t)$ represents the number of vehicles that will depart from the intersection at the maximum rate $Q_i^{\max}$ during the "clearing" state of duration $\hat{g}_i(t)$. In all other states, the right hand side of (2.16) becomes zero. In summary,

$$\frac{\mathrm{d}\hat{w}_i}{\mathrm{d}t} = \begin{cases} \hat{n}_i(t), & \text{if there is no service,} \\ 0, & \text{during the entire service period.} \end{cases} \tag{2.18}$$

The above analysis shows that the number of vehicles to be served in order to dissolve a queue, $\hat{n}_i(t)$, is a direct measure for the cost of delaying the service of a queue.

The reason why $\hat{w}_i(t)$ does not change during the service process is because the corresponding value has already been anticipated. However it will change again as soon as the service process is terminated. At the same point in time, the anticipated waiting time $\hat{w}_i(t)$ will increase by the additional amount $\Delta\hat{w}_i(t)$ due to the fact that the next green time cannot start before performing a new setup, which requires a time period of length τ_i^0. This additional setup waiting time is given by

$$\Delta\hat{w}_i(t) = Q_i^{\max} \int_{\tau_i(t)}^{\tau_i^0} \hat{g}_i(t, \tau')\ \mathrm{d}\tau', \tag{2.19}$$

where $\hat{g}_i(t, \tau)$ corresponds to the solution of (2.10), given a remaining setup time of τ'. Equation (2.18) and the expression in (2.19) allow for the anticipation of the costs of delaying or terminating a service process in terms of expected future waiting times. There is an important relationship between $n_i(t)$ and $\hat{n}_i(t)$: While $n_i(t)$ is the growth rate of the *current* waiting time $w_i(t)$ according to (2.7), $\hat{n}_i(t)$ is the growth rate of the *expected* future waiting time $\hat{w}_i(t)$ for a traffic flow i that is not being served.

The effect of approaching vehicles on the anticipated green time and waiting time is shown in Figure 2.8 for the corresponding states of the service process. In Figure 2.8 it may be seen that the service process starts early enough to serve a platoon of five vehicles in a green-wave manner, because the start of the service process is initiated by a platoon-related jump in the expected number of vehicles arriving along road section i, $\hat{n}_i(t)$, thus causing the sudden increase in the effective range.

2.4.2 The self-organising control method of Lämmer and Helbing

Given the values of anticipated required green times and predicted waiting times of vehicles along a road section, Lämmer and Helbing [25], propose a heuristic approach towards traffic control based on a self-organised prioritisation strategy. As was mentioned earlier, their heuristic is based on the observation of self-organised oscillations of the passing direction of pedestrians on either side of a bottleneck, where the passing direction changes when the pressure on one side of the bottleneck exceeds that on the other side by a sufficient amount. In terms of traffic control, the vehicles may be viewed as pedestrians, and the bottleneck may be seen to represent an intersection. The heuristic combines two strategies, an *optimisation strategy* and a *stabilising strategy*, both discussed below.

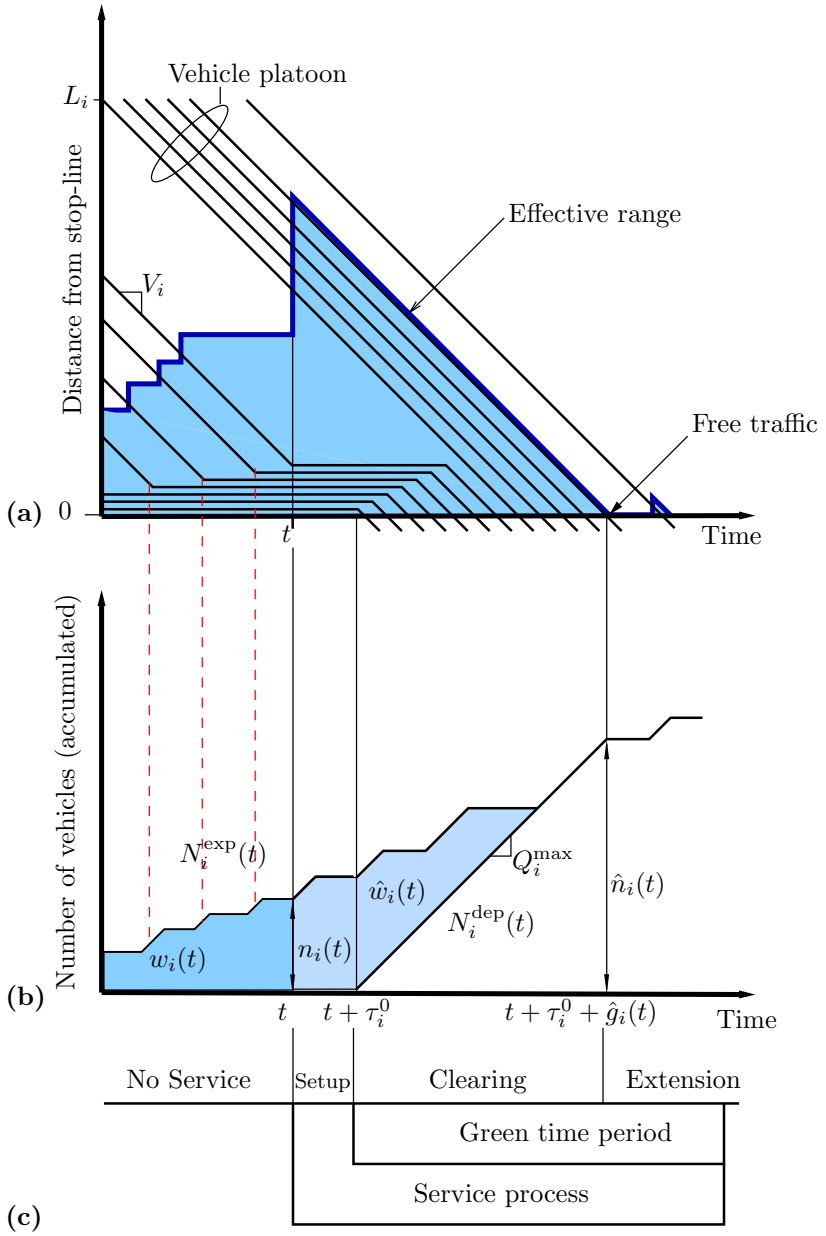


Figure 2.8: (a) Trajectories and (b) cumulated number of vehicles on a road section i, and (c) the different states of the service process [25].

The optimisation strategy

Lämmer and Helbing [25] use dynamic priority indices to define the aforementioned *pressures* associated with each approach to an intersection. The dynamic priority index of approach i at time t is denoted by $\pi_i(t)$, and service is provided to the traffic flow having the greatest priority. To represent this prioritisation rule mathematically the argument i of the traffic flow having the greatest priority at time t is stored in a decision variable $\sigma(t)$, in other words $\sigma(t) = \arg\max_i \pi_i(t)$.

The switching procedure may be explained by considering a pair of competing traffic flows, simply labelled 1 and 2. It is assumed that at the current time t, the values of the remaining setup times of both traffic flows, $\tau_1(t)$ and $\tau_2(t)$, are known, as well as the required green times, $\hat{g}_1(t)$ and $\hat{g}_2(t)$, and the number of vehicles expected to be served during the anticipated service periods, $\hat{n}_1(t)$ and $\hat{n}_2(t)$. It is also assumed, without loss of generality, that at time t, traffic flow 1 is currently receiving service, *i.e.* $\sigma(t) = 1$. In this scenario, the traffic flow controller has two options:

1. to complete servicing flow 1 before switching to flow 2 or
2. to switch to flow 2 immediately, at the cost of an additional setup for switching back to flow 1 later on.

In order to enable the controller to make the optimal decision in terms of minimising the overall waiting time of vehicles in the system, the expected waiting time associated with each decision is considered. In the case of the first option, traffic flow 1 will continue to receive service for a time period of $\tau_1(t) + \hat{g}_1(t)$ seconds. From (2.18) it may be seen that the waiting time of the vehicles along traffic flow 2 increases at a rate of $\hat{n}_2(t)$, while it remains constant for traffic flow 1 during its service process. The total increase of the anticipated waiting time associated with the first option is therefore

$$(\tau_1(t) + \hat{g}_1(t))\hat{n}_2(t). \tag{2.20}$$

For the case where the second option is selected, the termination of service to traffic flow 1 will result in the anticipated waiting time increasing by the amount $\Delta\hat{w}_1(t)$, according to (2.19). The value $\Delta\hat{w}_1(t)$ represents the additional waiting time associated with the setup time required when switching back to traffic flow 1 at a later stage. Once the change of service has taken place, traffic flow 2 will receive service for $\tau_2(t) + \hat{g}_2(t)$ seconds during which the anticipated waiting time of vehicles along traffic flow 1 will increase at a rate of $\hat{n}_2(t)$. The total increase in anticipated waiting time associated with option 2 is therefore given by

$$\Delta\hat{w}_1(t) + (\tau_2(t) + \hat{g}_2(t))\hat{n}_1(t). \tag{2.21}$$

It may be seen in (2.20) and (2.21) that, in terms of minimising anticipated waiting time, it is optimal to continue serving traffic flow 1 if

$$(\tau_1(t) + \hat{g}_1(t))\hat{n}_2(t) < \Delta\hat{w}_1(t) + (\tau_2(t) + \hat{g}_2(t))\hat{n}_1(t). \tag{2.22}$$

From (2.22), the individual priority indices $\pi_1(t)$ and $\pi_2(t)$ may be defined by separating their respective variables. The priorities $\pi_1(t)$ and $\pi_2(t)$ are given by

$$\pi_1(t) = \frac{\hat{n}_1(t)}{\tau_1(t) + \hat{g}_1(t)}, \quad \text{and} \tag{2.23}$$

$$\pi_2(t) = \frac{\hat{n}_2(t)}{\Delta\hat{w}_1(t)/\hat{n}_1(t) + \tau_2(t) + \hat{g}_2(t)}. \tag{2.24}$$

Unlike the definition of $\pi_1(t)$, which is a function of its own variables only, the definition of $\pi_2(t)$ includes the extra term $\Delta\hat{w}_1(t)/\hat{n}_1(t)$. In general the term $\Delta\hat{w}_\sigma(t)/\hat{n}_\sigma(t)$ reflects the penalty associated with terminating the current service process, where σ represents the traffic flow currently being served at time t. It follows by (2.19), that the value of $\Delta\hat{w}_\sigma(t)/\hat{n}_\sigma(t)$ ranges from 0 to τ_σ^0 and thus represents the additional waiting time $\Delta\hat{w}_\sigma(t)$ due to the additional setup for switching back, averaged over all corresponding vehicles $\hat{n}_i(t)$ that are due to receive service. Since the penalty for switching from σ to i applies only to those traffic flows $i \neq \sigma$ not being served, the penalty term is given by

$$\tau_{i,\sigma}^{\text{pen}}(t) = \begin{cases} \Delta\hat{w}_\sigma(t)/\hat{n}_\sigma(t) & \text{if } i \neq \sigma \\ 0 & \text{if } i = \sigma \end{cases}$$

in general. The priority index $\pi_i(t)$ is therefore in general defined as

$$\pi_i(t) = \frac{\hat{n}_i(t)}{\tau_{i,\sigma}^{\text{pen}}(t) + \tau_i(t) + \hat{g}_i(t)}. \tag{2.25}$$

Lämmer and Helbing [25] provide an interpretation of (2.25), by describing the priority index $\pi_i(t)$ as a representation of the anticipated average service rate, or, more specifically, the anticipated number of vehicles expected to receive service, $\hat{n}_i(t)$, during a time period of length $\tau_i(t)+\hat{g}_i(t)$. This definition of $\pi_i(t)$ depends on the anticipation of vehicle arrivals and the green time required to clear them, and takes into account the time losses associated with switching service from one traffic flow to another, as well as switching service back.

The stabilisation strategy

Lämmer and Helbing [25] define a traffic light control system to be *stable* if queue lengths will always remain finite, which requires that traffic demand does not exceed the intersection capacity. To ensure that stability is maintained, they propose implementation of a stabilisation strategy to complement the prioritisation rule in (2.25). An ordered priority set Ω is introduced to which the argument of any traffic flow i is added such that traffic flow i may be serviced as soon as possible in order to maintain stability along its length. To determine whether a traffic flow i requires immediate service at time t, a critical queue length value $n_i^{\text{crit}}(t)$ is introduced. The argument of a traffic flow i is added to Ω as soon as the anticipated number of vehicles $\hat{n}_i(t)$ requiring service along it, exceeds $n_i^{\text{crit}}(t)$. Once an argument of a traffic flow i has been added to Ω, it is removed at time t if the queue along it has been cleared, *i.e.* $n_i(t) = 0$, or else if it has received a green signal for a length of time equal to some maximum allowable green time $g_i^{\max}(t)$. The traffic flows associated with the arguments added to Ω are served on a first-come-first-served basis, with the first element, or head of Ω receiving service first. If Ω is empty, the control of the traffic signals is implemented by the prioritisation rule (2.25).

The combined strategy

The overall control strategy proposed by Lämmer and Helbing [25] may therefore be summarised as

$$\sigma = \begin{cases} \text{head}\,\Omega & \text{if } \Omega \neq \emptyset \\ \arg\max_i \pi_i & \text{otherwise.} \end{cases} \tag{2.26}$$

It is therefore a combination of two complementary regimes, the first being an optimising prioritisation strategy which attempts to minimise the waiting of vehicles along all the approaches

to the intersection by serving all incoming traffic as quickly as possible, while the stabilisation strategy intervenes only if the prioritisation strategy fails to keep the anticipated vehicle queues below some threshold value.

Service intervals

In order to implement the control strategy proposed by Lämmer and Helbing [25] it is necessary to specify the critical threshold value n_i^{crit} and the maximum allowable green time $g_i^{\max}$ for each traffic flow i, such that the stabilisation strategy alone fulfils the following two safety criteria. Each traffic flow shall be served

1. once, on average, within a desired service interval $U > 0$ and
2. at least once within a maximum service interval $U^{\max} \geq U$.

The two parameters U and $U^{\max}$ are the only adjustable parameters of the control algorithm. Lämmer and Helbing [25] define the length z_i of a service interval to be the time interval between the commencement of two successive service processes for traffic flow i. Thus, z_i may be seen as the sum of the red time r_i preceding the service process, the setup time τ_i^0 and the anticipated green time $\hat{g}_i$, *i.e.*

$$z_i = r_i + \tau_i^0 + \hat{g}_i. \tag{2.27}$$

Since the service interval length z_i may be anticipated, the critical threshold n_i^{crit} may be defined as a function $n_i^{\text{crit}}(z_i)$ of the anticipated service interval length z_i. One possibility presented by Lämmer and Helbing is

$$n_i^{\text{crit}}(z_i) = Q_i^{\text{exp}} U \frac{U^{\max} - z_i}{U^{\max} - U}. \tag{2.28}$$

The expression in (2.28) ensures that $n_i^{\text{crit}}(z_i) \leq 0$ for all $z_i \geq U^{\max}$, since within a desired service interval U, the number of vehicles that are expected to arrive is given by $Q_i^{\text{exp}} U$. This number, however, is equal to the critical threshold $n_i^{\text{crit}}(z_i)$ for an anticipated service interval of length $z_i = U$. Thus, a service process commences as soon as there are as many vehicles expected to require service as there are that are expected to arrive during the service interval U.

When determining the value of the maximum allowable green time $g_i^{\max}$ along traffic flow i, two constraints have to be considered. Firstly, $g_i^{\max}$ must be large enough to serve the number of vehicles expected to arrive during the service interval, given by $Q_i^{\text{exp}} U$, which requires a green time of at least $Q_i^{\text{exp}} U / Q_i^{\max}$, and secondly serving each traffic flow i one after the other for a period of $\tau_i^0 + g_i^{\max}$ should not take longer than U seconds in total. That is, $g_i^{\max}$ must meet the constraints

$$g_i^{\max} \geq U Q_i^{\text{exp}} / Q_i^{\max} \tag{2.29}$$

for all i and

$$\sum_i \left(\tau_i^0 + g_i^{\max} \right) \leq U. \tag{2.30}$$

Substituting (2.29) into (2.30) yields

$$\sum_i \tau_i^0 \leq \left(1 - \sum_i Q_i \text{exp} / \sum_i Q_i^{\max} \right) U. \tag{2.31}$$

An interpretation of (2.31) is that the sum of all service times must be smaller than the fraction of the service period U required for serving arriving vehicles. In order to minimise the total waiting time over an interval of length U, it is necessary to maximise the overall throughput of the intersection, given by $\sum_i Q_i^{\max} g_i^{\max}$. Lämmer and Helbing attempt to solve this optimisation problem by setting

$$g_i^{\max} = \frac{Q_i^{\exp}}{Q_i^{\max}} U + \frac{Q_i^{\max}}{\sum_{i'} Q_{i'}^{\max}} U^{\text{res}}, \tag{2.32}$$

which satisfies both constraints (2.29) and (2.30). The first term on the right hand side of (2.32) represents the minimum amount of time required to serve arriving vehicles during the service process, according to (2.29), while the second term adds a fraction of the *residual time*, U^{res}, proportional to the corresponding saturation flow rate $Q_i^{\max}$, where U^{res} is defined as

$$U^{\text{res}} = U(1 - \sum_i Q_i^{\exp}/Q_i^{\max}) - \sum_i \tau_i^0.$$

The value of U^{res} may viewed as the fraction of the service interval U which is not necessarily required for clearing vehicles. That is, the queues along each traffic flow i would remain stable even if no traffic flow was served for a period of U^{res} within the service interval U. Thus, U^{res} relates to the free intersection capacity, which is utilised in the literature to provide the maximum possible green times if they are required by the stabilisation strategy.

2.4.3 Adaptive self-organising traffic control

In a paper by Gershenson [15] the problem of *self-organising traffic lights* (SOTL) is approached as an adaption problem rather than as an optimisation problem. Gershenson proposes that an adaptive mechanism is better suited to real traffic situations, rather than a mechanism that is optimal sometimes, but has the potential to create havoc at other times. Three simple traffic-responsive methods for traffic light control that are adaptive through self-organisation are briefly reviewed in this section.

SOTL-request control

The SOTL-request control method forms the basis for all three of the aforementioned strategies, which function according to the same basic principles. When a traffic signal is red, a count κ_i is kept of the number of vehicles along approach i (regardless of the state or speed of the vehicles, *i.e.* the vehicles may be moving or stationary) multiplied by the number of time steps. When κ_i reaches a specified threshold value θ, the signals that are currently indicating green switch to indicate amber, and during the following time step the red signals that were responsible for keeping count, switch to indicate a green signal and the amber signals switch so that a red signal is displayed with $\kappa_i = 0$. Vehicles will therefore wait at a red signal until a sufficient number of vehicles have arrived at or are approaching the intersection along approach i at which time the signal will switch so that a green signal is displayed. When a sufficient number of vehicles are approaching a red signal, the signal will switch to green without the vehicles having to stop, allowing for the formation of a green wave or green band. Having platoons or convoys of vehicles moving together improves traffic flow compared to a homogeneous distribution of vehicles, since there are typically large areas between platoons which may be used by crossing platoons with little interference. It must be noted, however, that high traffic densities may trigger signals switching too frequently, resulting in a loss of service and obstructions to traffic flow.

SOTL-phase control

The SOTL-phase control method adds to that of the SOTL-request method by adding the constraint that a traffic signal will not switch if the number ϕ_i of time steps during which it has been displayed is less than some minimum value $\phi_{\min}$, *i.e.* $\phi_i < \phi_{\min}$. Once $\phi_i \geq \rho_{\min}$, the signal switches when $\kappa_i \geq \theta$, preventing the premature and excessive switching of traffic signals.

SOTL-platoon control

The SOTL-platoon control method imposes two further restrictions upon the SOTL-request method. Before changing a red signal to green, the method ascertains whether a platoon is crossing the intersection so as not to break it. If there is at least one vehicle within a specified distance ω from the intersection, a green signal will continue to be displayed. This ensures that platoons remain intact. This restriction may, however, cause havoc at high traffic densities. Therefore a second restriction ensures that the first restriction is not implemented if there are more than μ vehicles approaching the intersection.

2.5 Summary

The dynamics of traffic flow in a network and at signalised intersections, as well as a number of proposed methods of self-organised traffic control, were considered in this chapter. The chapter contains a review of currently employed traffic control techniques, and provides information on proposed methods of macroscopically modelling traffic flow along homogeneous road sections and at signalised intersections. The chapter closed with a thorough description of an anticipative self-organising traffic control algorithm based on the work of Lämmer and Helbing [25] and Lämmer *et al* [24], as well as three adaptive self-organising control algorithms proposed by Gershenson [15].

CHAPTER 3

Computer Simulation Modelling

Contents

This chapter contains a description of what simulation is, as well as the elements that comprise a simulation model and the processes involved when developing a simulation model. The advantages and disadvantages of using simulation as a modelling approach and its applicability to traffic network design and testing are also considered.

3.1 Simulation modelling

Simulation is defined by Banks *et al.* [5] as the imitation of a real world process or system over time such that the behaviour of the system can be studied by developing a simulation model, and it is from this model that data may be collected as if it were the real system being studied. Such simulation-generated data may then be used to investigate and analyse the performance of the system. According to Banks [3], simulation is an important problem-solving tool for the solution of numerous real-world problems in that it may be used to model both existing and conceptual systems, as well as being able to analyse and ask "what if" questions of these systems.

When developing a simulation model it is important to have an understanding of the problem at hand or the goal that is to be achieved. Different aspects and concerns of the model need

to be identified and analysed before the model building process begins. According to Borschev and Filipov [8] these include the availability of information regarding the system to be modelled as well as the ease with which the data may be obtained, *i.e.* whether the data are readily available or whether data collection would be necessary and if so, how complicated are the collection procedures involved. It is also necessary to identify important, fundamental system characteristics, and being able to distinguish them from insignificant, negligible system attributes. This is achieved by incorporating necessary assumptions and mathematical relationships that reduce modelling complexities. Above all, the validity and integrity of the model needs to be considered in the sense of ascertaining whether it is able to produce an acceptable account of the relevant system while producing viable and accurate results.

All the above factors have an influence on the selection of an appropriate level of model detail, which is described by Law [26] to be an art in itself. According to Borschev and Filipov [7], this selection process related to the extent of model detail has a direct influence on the modelling method chosen as well as how well the system will be modelled and simulated. In [48] the different levels of model abstraction are listed as a strategic level, an operational level and a physical level, as described in more detail below.

The aim of a simulation study on a strategic level is to identify and analyse strategic organisational issues, with relatively less detail being required for modelling at this level. Typical examples include the simulation of the effectiveness of an advertising campaign, or the spread of a disease among a large population of possible carriers.

The next level of model abstraction occurs an operational level. More detail is required when building a simulation model at an operational level, as this kind of model will typically be used to make tactical decisions about the system it is replicating. Examples of where an operational level of detail may be required include investigating optimal inventory levels of a company, balancing production lines in a factory and identifying problems in a system through analysis of relative performance measures.

The final and most detailed level of abstraction is modelling at a physical level. This level of detail is required when it is necessary to observe and analyse the behaviour of each individual entity of the system being studied, in terms of their speeds, positions and timings. This is typical when studying the dynamics of a transportation network or the movement of passengers through a bus or subway terminal, or when investigating evacuation procedures and their effectiveness.

3.1.1 Types of simulation models

Simulation models are classified as being static or dynamic, deterministic or stochastic, and discrete or continuous [4].

In the case of a static, or analytical model, the model represents the system at a particular point in time, and the result is a model which is functionally dependent on the model input parameters [8]. In a dynamic model, on the other hand, a system is represented as it changes over time, typically by making use of a set of rules that define how the system will change in future given its present state.

A deterministic model is one in which all the model inputs are known values, as opposed to a stochastic model in which there is a certain degree of randomness associated with the input parameters of the system, and as a result the same degree of randomness or uncertainty is associated with the model outputs. For this reason stochastic models may be considered as approximations or estimates of the true characteristics of the system being modelled.

A discrete model is one in which system state changes occur instantaneously at specific points in time. The queue in a bank may be considered as an example as its state (*i.e.* the number of people present in the queue) only changes when a person joins the queue or when a person leaves the queue to commence service at a bank teller. A continuous model, on the other hand, is one in which the state of the system changes constantly with respect to time. For example, the velocity and position of a vehicle along a road are characteristics that are either constantly changing, or have the potential to change constantly.

It must be noted, however, that a system is very seldom wholly discrete or wholly continuous — it is often a combination of both modelling approaches [26]. Banks [3] describes the process of simulating a combined discrete-continuous model as one in which the values of the system state variables are computed at small time steps, while the values of attributes of entities and global variables are calculated at discrete event times.

3.1.2 Modelling approaches

There are various different modelling approaches available when attempting to replicate a real-world system, depending on the level of abstraction and detail required in building the model, as dictated by the problem at hand. When simulating there are four main distinguishable approaches, namely system dynamics modelling, discrete event modelling, agent-based modelling and dynamic systems modelling [48].

System dynamics modelling

System dynamics (SD) modelling is most commonly used as a tool to observe the effects of interaction among organisational structures and other influencing factors on the performance of an enterprise as a whole. In SD modelling, real-world processes are replicated and implemented through the use of causal loops and stock-flow diagrams in which stocks represent the system commodities, *e.g.* people, money, or materials. SD modelling is generally employed in long-term strategic models and assumes a high level of abstraction and aggregation as objects being modelled are represented by their quantities and therefore lose their individual properties and dynamics.

Discrete event modelling

In discrete event modelling, system state changes occur at a countable number of discrete points in time, with the goal being to portray the activities in which the components, or entities of the system engage and in so doing, gaining insight into the system's dynamic behaviour. This is achieved by defining the various states of the system as well as the activities that are responsible for moving it from state to state, with the beginning and end of these activities being known as events. The state of the model remains constant between consecutive events; the dynamic workings of the system are made evident as the simulation time advances from one event to the next [3].

Agent-based modelling

Agent-based modelling is most simply defined as a decentralised, individual-centric approach to model design [48]. The fundamental difference between agent-based modelling and SD modelling

is that in agent-based modelling the behaviour of the global system as a whole is not defined, but instead the behaviour of the individual entities, the so-called agents of the system, which can be anything from people or vehicles to companies, is defined, and it is from interactions among these agents that the global behaviour of the system results or is derived [8].

Dynamic systems modelling

Borschev and Filipov [8] present the model of a dynamic system to rely on various state variables as well as algebraic differential equations of different forms pertaining to these variables. It is also noted how the integrated variables in a dynamic system model have direct physical attributes (such as location, velocity or acceleration) in contrast to the variables concerned with system dynamics modelling. Dynamic system tools have been developed to be embedded in the engineering design cycle and to make use of graphical modelling computer languages.

The above-mentioned modelling approaches are suited to different levels of detail and abstraction, as required. However, it is not uncommon for integration among the processes to occur in order for a model to achieve the desired result [8].

3.1.3 The composition of a simulation model

There are a number of underlying concepts and components to any simulation model, and these vary according to the system being modelled. These include the *entities* of the model and their individual *attributes*, as well as the *activities* of the various entities in the system, and finally the *events* which may result in a change in the *state variables* of a system [4]. Descriptions of each of these model components follow.

An entity is an item of interest to the system requiring specific definition. An entity may be dynamic in that it moves through the system, or it may be static, in which case it will typically interact with or serve other entities in the system. An attribute is a property associated with each individual entity, and characterises that individual entity alone [3].

An activity is a process undertaken by an entity for which the duration is known prior to the commencement of the activity. The duration of an activity may be a constant or a random value drawn from a probability distribution. Unlike an activity, the duration of a delay is indefinite and is a result of some combination of system conditions. Discrete-event simulations contain activities that cause time to advance as well as delays where entities wait for some service [3]. The commencement and termination of an activity or delay are events.

An event may be described as an instantaneous occurrence which may alter the state of the system being modelled [4]. An event may be *endogenous*, in which case it occurs within the system, or *exogenous*, in which case it occurs in the external environment of the system, but still has an effect on the system.

The system state variables are a collection of all information required to define what is happening within a system to a sufficient level of detail at a given point in time, relative to the objectives of the study [3, 4]. The system state variables allow for a distinction to be made between discrete-event and continuous models. In the case of a discrete-event model, the system state variables remain constant over intervals of time, changing only upon the occurrence of specific events, whereas in a continuous model the system state variables may change continuously over time [3].

3.1.4 Steps in a simulation study

There are a number of steps that need to be taken, and sometimes repeated, when working towards a successful simulation study. These steps are presented below and are summarised graphically in Figure 3.1.

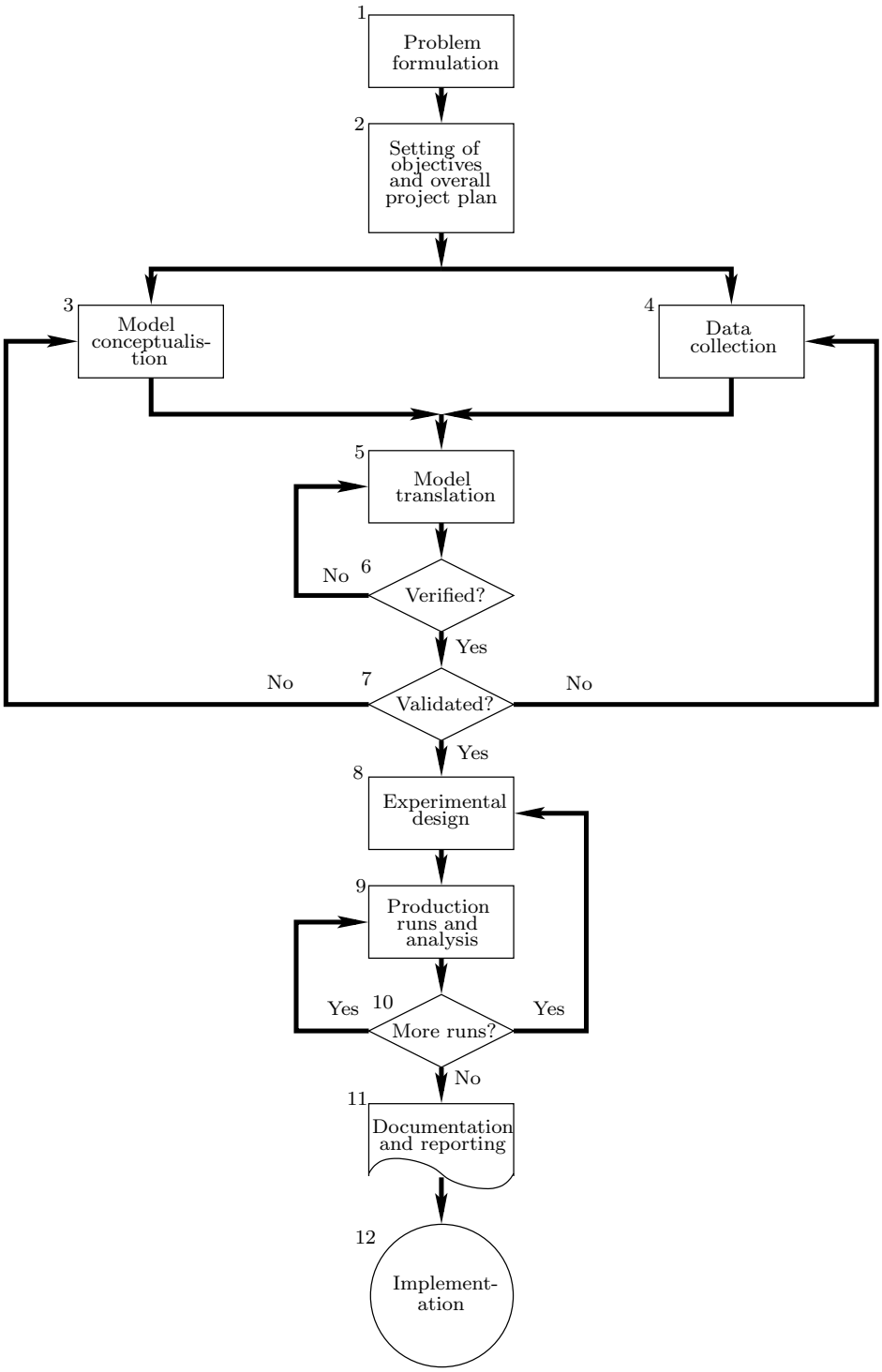


Figure 3.1: Steps in a simulation study [3].

1. *Problem formulation.* Every simulation study begins with a statement of the problem at hand. This statement must account for the overall objectives of the study, as well as the specific questions which are to be answered in the study, such that the required level of model detail may be decided upon [26].

2. *Project planning.* The project plan should include a statement on the scope of the project together with the various scenarios that will be investigated. It should also include information about the performance measures that will be used to evaluate the efficiency of the different system configurations investigated. The plan for the study should be indicated in terms of the time frame required for the completion of the study, the hardware and software requirements, as well as the various stages of the study and the output expected after each stage.

3. *Model conceptualisation.* The real world system under investigation is abstracted to a conceptual model, which is a series of mathematical and logical relationships concerning the components and structure of the system. The art of modelling lies in an ability to abstract the *essential* features of a problem that characterise the system, and then to elaborate upon the model until a useful approximation to the real world system under investigation results [4]. When deciding upon the appropriate level of model detail, the project objectives should again be considered as well as the performance measures of the model and the availability of data, as well as the performance constraints of the computer system in the model will be implemented [26].

4. *Data collection.* There is a constant interplay between the construction of a model and the collection of the required input data, with the objectives of the study dictating, to a large extent, the data to be collected. If the complexity of a model changes to better suit the objectives of a study, the required input data may change as well [4]. In addition to collecting the required input data, it is also necessary to collect data pertaining to the performance of the current system for validation purposes [26].

5. *Model translation.* A suitable programming language or simulation software package is selected to convert the conceptual model developed in Step 3 above into an operational model. The benefits of using a known programming language to code and implement the model is that it offers the user greater control over the model, while using a specialised simulation software package greatly reduces programming time [26].

6. *Model verification.* Verification pertains to the computer model built when implementing the conceptual simulation model and seeks to answer the question "Is the computer program performing properly?" [4]. Banks [5] advises that verification should take place continuously throughout the construction of a model, and that use is made of interactive run controllers and debuggers to aid in the verification process.

7. *Model validation.* Validation is the process of determining whether the conceptual model is an adequate representation of the real system. Validation is achieved by comparing the simulation model to the real world system it is replicating and using any discrepancies between the two, as well as input from subject matter experts to improve upon the accuracy of the model. This process is repeated until model accuracy is judged to be acceptable [4].

8. *Experimental design.* The alternative system designs that are to be investigated are decided upon, and for each scenario, decisions are made concerning the length of the simulation run, the number of runs, or replications to be implemented, and the manner of the system state initialisation.

9. *Production runs and analysis.* Production runs, and their subsequent analysis, are used to estimate measures of performance for the scenarios that are being simulated, allowing for a comparison between alternative system configurations in a relative sense.

10. *More production runs.* Based on the output of the initial production runs, it is decided whether further production runs are required. If required, the design of these additional experiments should follow.

11. *Documentation and reporting.* There are two types of documentation, namely documentation of the program used to implement the simulation model and documentation of the progress of the simulation study. Program documentation is important if the program is to be used by other analysts, so that they will be able to understand how the program operates. If in the future it is felt that the program needs to be altered or modified in any way, sufficient documentation will greatly facilitate this process. Progress documentation includes information on model specifications, prototype demonstrations, model animations, and progress reports, all delivered before the presentation of a final report [4].

12. *Implementation.* The final implementation of the processes suggested by the simulation study depends on how successfully the previous eleven steps have been implemented, and how convincing the final report and the simulation study's findings are to the final decision maker.

3.2 Reasons for using simulation

Competition in the information technology sector has led to technological breakthroughs which have resulted in computer hardware becoming faster, more powerful and easier to use. These advancements have had a positive effect in the simulation software industry too, as specialised simulation software programs can now make use of the improved hardware performance, resulting in more accurate execution with quicker completion times, allowing for the expansion of the application of simulation in industry. Many companies have adopted simulation after realising the benefits it provides when studying complex systems [3, 26]. The advantages of using simulation are mentioned by many authors (see, for example, Banks [3], Banks *et al.* [4] and Law [26]):

- Simulation allows for control of a system in its entirety, which means that new policies may be investigated without committing any additional resources to their real world implementation. It also allows for modifications or additions to currently employed process to be made and the results analysed without disrupting ongoing operations of the real world system. As a result various proposed system designs may be compared via simulation to ascertain which one best meets the specified requirements.

- One of the main advantages of simulation is that it provides the ability to compress and expand time, such that an entire time shift of days, or even weeks may be observed in a matter of minutes. Time may also be slowed down in order to investigate events which occur too rapidly to analyse sufficiently in the real-world. This control over the model time allows for phenomena to be studied in an attempt to better understand the causes for their occurrence.

- Using simulation, insight may be gained into the interaction of variables which make up a complex system. This allows for problems to be diagnosed when they arise, and also provides information on the effect of certain variables on the system as a whole.
- Simulation is an efficient tool for performing bottleneck analysis as it allows for the cause of bottlenecks in a system to be identified, and methods for alleviating these bottlenecks to be investigated.
- Many simulation software packages allow for an animation to accompany the simulation model, enabling a visualisation of the system and how its various components operate and interact.
- Simulation is often a valuable tool in building consensus towards the implementation of a proposed policy. This is because there is less risk on behalf of a decision maker in accepting reliable simulation results, which have been modelled, tested, validated and visually represented, instead of personal opinions of the results that will occur from a proposed design.

3.3 Disadvantages of using simulation

Although there are many advantages to using simulation, it is not without its disadvantages. Some of these disadvantages of adopting a simulation modelling approach, as documented in the literature, are described below.

- Simulation model building is considered an art as much as it is a science. It is a skill learned over time and through experience, and it requires specialised training. This drawback has, however, been offset somewhat through the development of software packages that contain complete or partially complete models which require only input data from the user for their operation.
- Simulation results may be difficult to interpret, especially in cases where the system inputs are random, as it may be hard to determine whether the observed results are due to system interactions or randomness. To counter this, many simulation software packages provide output analysis functionalities or tools of some sort.
- The building and running of a simulation model can be very time consuming and many of the commercially available simulation packages, particularly the more professional targeted ones, are expensive. Due to advances in technology, however, running times of various simulation packages are being reduced, and many software vendors are releasing various copies of software packages ranging from basic to professional versions, which reduces the required capital outlay if a professional package is not entirely necessary.

3.4 Simulation in the context of traffic flow and control

According a special report by the Transportation Research Board [14], traffic simulations were run on computers in the United States of America as early as 1954. According to the report it became accepted that traffic simulation was possible and feasible by about 1960, at which time efforts were increasingly directed to the development, validation and use of large-scale simulation programs.

When investigating the role of simulation in the context of traffic flow and control, the work of Barceló [6] and Papacostas and Prevedouros [33, 35] is central. Simulation is considered to be an important tool for the analysis of highways and urban street networks, as it is through simulation that transport experts are able to study the formation and dissipation of congestion on roadways, assess the impact and effectiveness of various traffic control strategies and compare different geometric roadway designs [33]. A variety of commercial traffic simulation models are available which vary according to the specifications of the systems for which they were designed. Gibson [16] classifies traffic simulation models as those concerned with modelling intersections, arterials, urban networks, freeways, and freeway corridors.

In this section, three different types of traffic simulation modelling approaches are described. These approaches are prevalent in the literature, and are microscopic traffic simulation, macroscopic traffic simulation and mesoscopic traffic simulation.

3.4.1 Microscopic traffic simulation

Microscopic modelling of traffic flows is concerned with the motion characteristics, *i.e.* acceleration, deceleration and lane changes, of each individual vehicle comprising the traffic stream [6]. To model driver reactions to the surrounding traffic, a *vehicle following model* is typically employed. A number of vehicle following models are proposed in the literature, each of varying degree of complexity. The classical vehicle following approach is relatively straightforward in that each vehicle attempts to travel at its desired speed while maintaining a safe following distance from the vehicle in front of it [35]. Barceló [6] describes the pioneering work carried out on vehicle following models in the 1950s, some of which were based on the California Motor Vehicle Code, as follows: "A good rule for following another vehicle at a safe distance is to allow yourself at least the length of a car between your vehicle and the vehicle ahead for every ten miles per hour of speed at which you are travelling." Most microscopic traffic simulation models are stochastic in nature, employing Monte Carlo processes to generate random numbers used in generating vehicle arrival times as well as driver and vehicle behaviour in the system.

A microscopic simulation approach was followed in the development of the traffic simulation model presented later in this thesis. The entities of the model include the vehicles, the traffic lights themselves and the road sections on which the vehicles travel. The attributes of each individual vehicle include the velocity at which the vehicle travels, its position along a road section, its position in the queue along the road section, and its colour. The attributes associated with the traffic lights are the timings of each phase in the cycle, and the attributes of the road sections are the positions of their entry, turning, stopping and destination points. The exogenous events of the model include the arrival of a new vehicle to the system, while the endogenous events include the commencement of a vehicle's deceleration, acceleration or changing of lanes, as well as a phase change in the cycle of the traffic lights. The service that vehicles compete for is the green signal provided by the traffic lights, and all vehicles experience a delay as they wait for service.

Microscopic traffic simulation software

Traffic simulation has become an indispensable instrument for transport planners and traffic engineers. Barceló [12] presents information on one such microscopic traffic flow simulator, VISSIM. VISSIM is a microscopic, behaviour-based multi-purpose traffic simulation model which may be used to analyse and optimise traffic flows. It is a commercial software tool, and is

used worldwide within consultancies and industry, public agencies and academic institutions. The software is primarily suited to traffic engineers. However, as the need for greater detail in intelligent transport systems increases, so too does the number of transport planners who use VISSIM. Some common areas of application of VISSIM include, but are not limited to the development and analysis of management strategies on motorways (including impacts during phases of construction), corridor studies on arterials with signalised and non-signalised intersections, analysis of alternative actuated and adaptive signal control strategies in traffic networks, and investigations with respect to so-called traffic calming schemes.

Papacostas and Prevedouros [35] describe a second microscopic simulation model called NETSIM. NETSIM, or the NETwork SIMulation model, was originally called UTCS-1, because its development was supported by the Office Of Research of the U.S. Federal Highway Administration as part of its Urban Traffic Control System program. NETSIM is a microscopic, interval scanning simulation model capable of representing complex urban networks, traffic control systems and vehicular performance characteristics. The motion of each vehicle in the system is governed by a vehicle following model, including several vehicle characteristics such as vehicle turning.

3.4.2 Macroscopic traffic simulation

Macroscopic traffic flows are typically modelled from an aggregated point of view, based on a hydrodynamic analogy, by regarding traffic flows as a particular fluid process whose state is characterised by aggregate macroscopic variables, such as density, volume and speed [6]. A continuum model is usually employed when modelling traffic macroscopically. This continuum model consists of a continuity equation which represents the relationship between the speed, density and volume of the traffic flow. However, the simple continuum model does not consider acceleration and inertia effects and cannot describe non-equilibrium traffic flow dynamics with precision. A higher order continuum model accounts for this lack of accuracy, by incorporating acceleration and inertia effects by means of a momentum equation which describes the dynamic speed-density relationships observed in real traffic flow, together with a continuity equation [35].

Macroscopic traffic simulation software

A macroscopic traffic simulation package called TRANSYT is described by Papacostas and Prevedouros [35]. The TRAffic Network StudY Tool, or TRANSYT, was originally developed by Dennis Robertson at the Transport Road Researach Laboratories (UK) in 1967. Individual vehicles and their properties are not represented in TRANSYT; instead all calculations are based on average vehicle flow rates, turning movements and queue lengths. TRANSYT-7F may be used as both a simulation tool and an optimisation tool. Implementing it as a simulation tool, the performance of the existing model without any alterations is generated as output. When run as an optimisation tool, it manipulates the cycle lengths, green splits and offsets of intersections in an attempt to minimise delays and improve progression.

3.4.3 Mesoscopic traffic simulation

Mesoscopic traffic simulation, in essence, combines aspects of both microscopic traffic simulation and macroscopic traffic simulation, in that it has the ability to account for individual vehicles in the system, but is still primarily concerned with the traffic dynamics of the vehicles as a whole

and does not explicitly consider the details of the vehicle lane changing and vehicle following behaviour [6, 35]. Because of this modification, mesoscopic models are usually less demanding of data and computationally more efficient when compared to microscopic traffic simulation models. There are two variations of mesoscopic models, those in which individual vehicles are not considered and vehicles are instead grouped into packages or platoons which move along road links, and those in which flow dynamics are determined from simplified dynamics of the individual vehicles [6].

Mesoscopic traffic simulation software

The COntinuous TRaffic Assignment Model, or CONTRAM, is an example of a traffic assignment and simulation model that treats groups or packets of vehicles as single entities [35]. It was developed primarily for traffic assignment purposes and can be used for simulating vehicle routing in complex traffic systems.

3.5 Summary

In this chapter, various types of simulation approaches and models were described. The various components which make up a simulation model were also described, and the steps that are typically followed in a sound simulation study elaborated upon in §3.1. The advantages and disadvantages associated with simulation studies were stated in §3.2 and §3.3, respectively. The chapter closed with a brief description of the applicability of simulation with respect to modelling and investigating traffic flows along road networks as well as commercially available software packages for the various types of traffic simulation modelling.

CHAPTER 4

Traffic Simulation Model

Contents

A simulation model was designed and implemented in order to investigate the effectiveness of various traffic light control regimes. The concepts employed and assumptions made during the design of this simulation model, as well as the ideas and philosophies behind the various aspects of the model, are described in this chapter. The actual implementation of the logic behind the workings of the model is discussed, and various verification and validation techniques are employed.

4.1 Concepts and assumptions of the simulation model

The model was implemented in the software suite AnyLogic University 6.5.0 [48], a Java based simulation environment. The most fundamental building blocks of the model were individual road sections linking intersections, the vehicles that travel along these road sections, and the traffic lights that control the flow of conflicting vehicle streams at intersections. Through replication and interaction of these elements, various different road network layouts and traffic situations could be simulated, observed, and analysed. Each of these three elements are considered in closer detail in this section.

4.1.1 Modelling the road sections

In the context of this model, a road section describes a single lane along a (possibly multi-lane) segment of road which acts as a pathway between intersections for the vehicles which travel along it. Each road section has at least three main points of interest along its length. These are an entry point (a point at the beginning of the road section where vehicles enter into the road section), a stopping point (a point at which the vehicle closest to the intersection along the road section will stop when it is not able to enter the intersection), and a destination point (the point towards which all the vehicles along the road section travel at the end of the road section). In addition to these three points, a road section may contain certain other points. For example, if a road section runs parallel to a right turning only lane, then it will possess a lane changing point, where a vehicle may change lanes in preparation to turn. Similarly, if a vehicle can turn left or right from a road section in an intersection, there will be specific turning points along the road section for them to do so. An ordered set of vehicles is associated with each road section. A vehicle is added to the set when it enters the road section at the entry point, and it is deleted from the set when it leaves the road section, either upon its arrival at the destination point or at a lane changing point.

4.1.2 Modelling the traffic signal controls

There are numerous different phases that the signals of a set of traffic lights may step through at an intersection. For example, besides the well known, green, amber, red, and all red phases, traffic lights may also contain exclusive left and right turning phases as well their associated amber phases.

The traffic lights were modelled making use of state charts, where each state represents a specific phase in the traffic signal control cycle, *i.e.* when the active state of the traffic light's state chart is the green state, the traffic light in question displays a green signal. The crux of modelling the traffic lights, however, is to effectively model the requirements for the transitions from state to state. The model allows for various user controlled transition techniques and parameters, from simple timed switching sequences, to more complex dynamic switching criteria.

An in-depth discussion of the various switching control techniques implemented and the methodologies behind them are presented in the next chapter.

4.1.3 Modelling vehicle accelerations

A fixed speed limit is assumed for all road sections considered in the model, and it is further assumed that all vehicles in the system attempt to achieve this speed limit subject to the constraint of allowing for safe following distances, with no vehicles exceeding the speed limit. Thus, any vehicle can be in one of four states at any given time, namely, cruising (at the speed limit), decelerating, accelerating, or stationary. A discussion of each of these four states follows, focusing on the conditions necessary for a vehicle to enter each state, as well as changes that are affected to the vehicle's motion dynamics in each state.

The cruising state

Any vehicle that enters the system through one of the model entry points is assumed to do so at the speed limit, and thus in the *cruising* state. It is assumed that a vehicle will continue in this

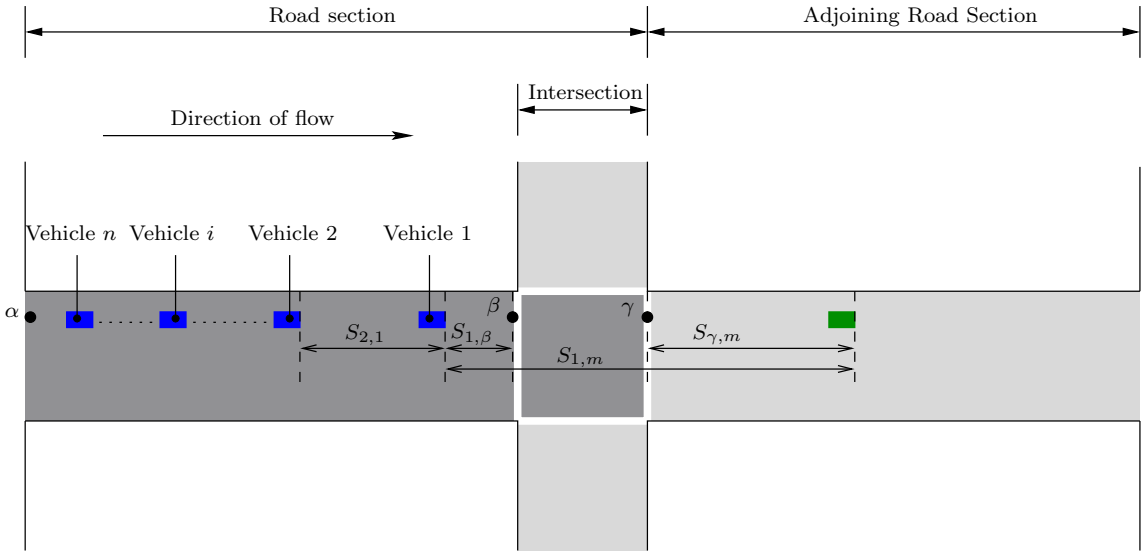


Figure 4.1: Graphical illustration of vehicles along a road section. The road section considered is shown in dark grey, while the adjoining road sections towards which vehicles may be travelling are shown in light grey. The points marked α, β and γ represent the entry point to the road section, the stopping point of the road section, and the destination point of the road section, respectively. Point γ is the entry point of the adjoining road section.

cruising state, travelling at the speed limit, provided that it is sufficiently far from the vehicle in front of it and from the next signalised intersection where a traffic signal indicates that it should slow down in preparation to stop. In the case that a road section is congested to a point where its entire length is occupied by vehicles, no new vehicle will enter the road section until there is sufficient space for it to do so.

The State of Deceleration

All decelerations were modelled according to equations which assume a vehicle to travel with *constant* acceleration in a straight line, and which were derived from Newton's laws of linear motion [49]. There are various circumstances under which a vehicle may enter the *decelerating* state, and these are primarily concerned with whether or not the vehicle is the front vehicle along the road section, or not. The concept of the *front* or *first* vehicle is explained by Figure 4.1, where it is shown as the vehicle closest to the destination point of the road section. For a set of n vehicles along a road section, an index i is employed denoting the order of vehicles along the road section, with i taken as 1 for the front vehicle, 2 for the second vehicle and so on. The conditions under which the first vehicle decelerates, are described in the following paragraphs. When it is necessary for a vehicle to decelerate, the appropriate rate of deceleration for the vehicle is calculated on a per-second basis.

If, at time t the vehicle is approaching the stopping point and its distance to the stopping point is less than a particular decelerating distance $Tv^{\max}$, where T is the safe time gap between the vehicle and the stopping point and $v^{\max}$ is the speed limit along the road section, and a red traffic signal is indicated, the vehicle decelerates at a rate of

$$d_1(t) = \frac{-v_1(t)^2}{2S_{1,\beta}(t)} \quad m/s^2, \tag{4.1}$$

where $v_1(t)$ denotes the velocity of the first vehicle at time t, and $S_{1,\beta}(t)$ denotes the distance from the first vehicle to the stopping point of the road section at time t.

Otherwise, if an amber signal is indicated, and the vehicle is not sufficiently close to the intersection to make it across the intersection to the destination point of the road section at its current speed before a red traffic signal is indicated, nor is it sufficiently far from the stopping point of the road section to safely continue travelling at its current speed, the vehicle decelerates linearly to stop at the stopping point of the road section at the rate in (4.1). That is to say, if at time t the first vehicle's distance to the stopping point $S_{1,\beta}(t)$ of the road section lies within the range

$$\frac{A^{\text{rem}}(t)v^{\max}(t)}{2} - S_{\beta,\gamma} \leq S_{1,\beta}(t) \leq 2Tv^{\max}, \tag{4.2}$$

where $A^{\text{rem}}(t)$ denotes the remaining amber time at time t, and $S_{\beta,\gamma}$ denotes the distance between the stopping point and the destination point of the road section which corresponds to the length of the intersection which needs to be crossed when travelling from the stopping point to the destination point, then the vehicle will decelerate at the rate indicated in 4.1. If the front vehicle's distance to the stopping point of the road section is outside this range, then it either travels at the speed limit, or it accelerates until its speed reaches the speed limit.

In addition to the status of the traffic signal in front of it, the front vehicle on a road section is also modelled so that it considers its speed relative to the last vehicle on the adjoining road section, *i.e.* the first vehicle it will encounter upon crossing the intersection. Vehicles do not enter an intersection without being able to safely clear it, thereby preventing congestion within an intersection. Therefore, regardless of the traffic signal indicated, the front vehicle does not cross an intersection unless there is sufficient space to accommodate it along the adjoining road section, and if there does happen to be sufficient space, the front vehicle adjusts its speed according to its distance to the last vehicle on the adjoining road section, as it approaches the destination point of the road section. That is, if the last vehicle on the adjoining road section is not more than $1/k^{\text{jam}} + Tv^{\max}$ metres from the adjoining road section's entry point and has a speed less than $v^{\max}/2$, then the front vehicle decelerates at the rate in (4.1) in order to stop at the stopping point of the road section. Here, as in Chapter 2, $1/k^{\text{jam}}$ denotes the vehicle length plus the minimum distance between vehicles when queued. If, however, the front vehicle has crossed the stopping point of the road section into the intersection at time t, and it comes within a distance of $1/k^{\text{jam}} + Tv_1(t)$ metres to the last vehicle of the adjoining road section, then it decelerates at a rate of

$$d_1^1(t) = \frac{-v_1(t)^2}{2\left(S_{1,m}(t) - \frac{1}{k^{\text{jam}}} - Tv_1(t)\right)} \quad m/s^2 \tag{4.3}$$

where $S_{1,m}(t)$ denotes the distance between the nose of the front vehicle on the road section and the nose of the last vehicle of a set of m vehicles along the adjoining road section at time t. It may be seen from (4.3) that when calculating the deceleration of the first vehicle along the road section relative to the last vehicle along the adjoining road section, the last vehicle is taken to be stationary. This implies that even if the last vehicle on the adjoining road section were to come to a complete stop, the vehicle following it would safely come to stop $1/k^{\text{jam}}$ metres behind it.

For the case where a vehicle is not the front vehicle along a road section, *i.e.* for all $i = 2, \ldots, n$, vehicle i decelerates if it is too close to the vehicle in front of it, *i.e.* vehicle $i-1$, for the speed at which it is travelling. Vehicle i is considered to be too close to vehicle $i-1$ at time t if, for a speed $v_i(t)$ of vehicle i, the nose-to-nose distance between the two vehicles is less than the safe

following distance $1/k^{\text{jam}} + Tv_i(t)$ metres, in which case vehicle i decelerates at the rate

$$d_i(t) = \frac{-v_i(t)}{2\left(S_{i,i-1}(t) - \frac{1}{k^{\text{jam}}} - Tv_i(t)\right)} \quad m/s^2. \tag{4.4}$$

Again, it may be seen that when calculating the deceleration of vehicle i relative to vehicle $i-1$, vehicle $i-1$ is taken to be stationary for the same reason as mentioned earlier regarding safe stopping distances. Also, if vehicle $i-1$ has crossed the stopping point of the road section and has entered the intersection, but has not yet reached the destination point of the road section, then vehicle i decelerates according to the same rules as were described for the front vehicle in response to the traffic signal indicated.

The state of acceleration

When modelling acceleration it may be assumed that if vehicles accelerate at all, they do so at a constant rate so as to achieve the speed limit. A vehicle is assumed to continue accelerating until either it has to decelerate or until its speed equals the speed limit, at which point it enters the cruising state, *i.e.* continues to travel at the speed limit. As with deceleration, when it is necessary for a vehicle to accelerate, the appropriate rate of acceleration is calculated on a per-second basis.

The conditions under which the state of acceleration is triggered for the front vehicle are different than those for the other vehicles along a road section. The front vehicle is assumed to accelerate if its speed is less than the speed limit, and both the traffic signal ahead and the position and speed of the last vehicle on the adjoining road section permits it to do so. For instance, if the front vehicle is stationary at the stopping point of the road section due to the indication of a red traffic signal, then upon commencement of a green signal at time t, and assuming that the last vehicle on the adjoining road section is further than a distance of $1/k^{\text{jam}} + Tv^{\max}$ metres from the adjoining road section's entry point, the front vehicle accelerates at a rate of

$$a_1(t) = \frac{{v^{\max}}^2 - v_1(t)^2}{2Tv^{\max}} \quad m/s^2. \tag{4.5}$$

For the case of vehicles other than the front vehicle in relatively free flowing traffic, at time t vehicle i accelerates at a rate of

$$a_i(t) = \frac{{v^{\max}}^2 - v_i(t)^2}{2Tv^{\max}} \quad m/s^2 \tag{4.6}$$

if its speed is less than the speed limit and the distance to vehicle $i-1$ is greater than $1/k^{\text{jam}} + Tv_i(t)$ metres.

If vehicles accelerate out of a queue and vehicle i is still stationary, it remains stationary until the distance between itself and vehicle $i-1$ at time t is greater than $1/k^{\text{jam}} + Ta_{i-1}(t)$ metres, which corresponds to vehicle i allowing vehicle $i-1$ to accelerate at the rate $a_{i-1}\, m/s^2$ for T seconds before it starts accelerating.

However, in the case where traffic is congested, if the space between the nose of vehicle i and the nose of vehicle $i-1$ is greater than $1/k^{\text{jam}}$ metres and both vehicles are stationary at time t, then vehicle i accelerates at the reduced rate of

$$a_i^1(t) = \frac{\frac{{v^{\max}}^2}{4} - v_i(t)}{2\left(S_{i,i-1} - 1/k^{\text{jam}}\right)} \quad m/s^2. \tag{4.7}$$

The stationary state

Any vehicle along a road section enters the stationary state when its speed decelerates below a predetermined, positive, threshold value (taken here as 0.01 m/s). The reason for this is to avoid rounding errors as the ANYLOGIC simulation environment considers a vehicle to be in motion unless its velocity is absolute zero. Thus, even when a vehicle is travelling extremely slowly and would appear to be stationary, it is still considered to be moving, necessitating the need for the threshold value. A vehicle remains stationary until it is triggered to enter the acceleration state.

4.2 Model implementation

As mentioned in the previous chapter, the simulation model was built adopting an agent-based, discrete-event modelling approach and was implemented in the software simulation suite ANYLOGIC UNIVERSITY 6.5.0 [48], a JAVA based simulation environment.

The first step in the model implementation process was the design of the road network on which vehicles would travel. This included specifying the dimensions of the road sections that would be used, *i.e.* length, width, and the number of lanes. The specifics of intersections also required consideration before implementation in terms of the number of lanes approaching and departing the intersections, as well as the flow properties of these lanes (*i.e.* turning only, turning and straight on, or straight only). Once the road sections and intersections had been implemented it was necessary to overlay them with the paths that vehicles would follow as well as points of importance, such as the entry, stopping, destination, lane change and turning points.

Once the road network design and properties had been successfully implemented, the next step was to introduce the traffic signals that would control traffic flow along the road sections and at the intersections. For this, each individual set of traffic signals was designed as a separate agent in the model. The design of each traffic signal depends on the properties of the intersection which it is required to control. The colour of each light in the traffic signal depends on which state is active in the state chart responsible for the phasing of the traffic signals. Each different phase of the traffic signals had to be incorporated into the state chart that is responsible for the control of the traffic signals. Once the state charts of the traffic signals had been designed, it was necessary to implement the conditions necessary for the different transitions from phase to phase. The different traffic signal control algorithms were each only coded once, allowing for the user of the simulation model to select which control technique would be implemented at the various intersections. More specifically, each control algorithm was coded and made available at each intersection via a "check-box" on the main screen of the simulation thereby allowing the user to interactively select a desired control algorithm. An example of a simple traffic intersection thus implemented in AnyLogic may be seen in Figure 4.2. As mentioned earlier, the details pertaining to the various traffic signal control algorithms will be described later.

After the successful design and implementation of both the road network structure and the traffic signal controls, the model was ready to accommodate vehicles. As mentioned earlier, each road section in the model is equipped with a list of vehicles. An agent was created for each road section, which acts as a template for all the vehicles travelling along that specific road section. This agent is responsible for the colour and orientation of the vehicle as well as its movement characteristics: A typical example of one of these templates may be seen in Figure 4.3. This template is responsible for replicating vehicles as and when is necessary. Generation events are associated with each of the vehicle sets which enter the system from road section entry points. These vehicle generation events may be triggered in two different ways; they may

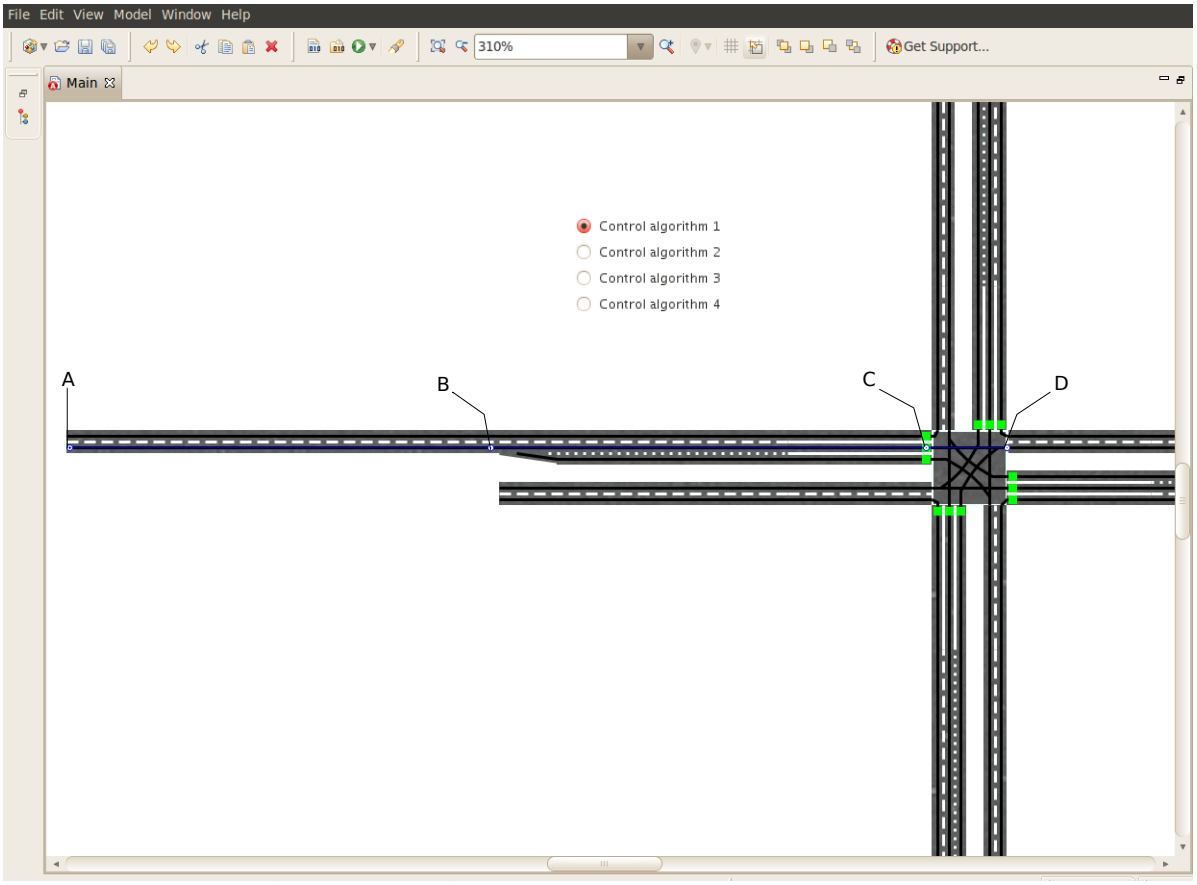


Figure 4.2: Screen shot of a typical intersection implemented in AnyLogic. The black lines indicate the paths that the vehicles follow when travelling along the individual road sections, while the green rectangles represent the traffic signals which are responsible for controlling the flow of traffic along each road section approaching the intersection. A check box may be seen in the upper left-hand quadrant. This check box allows the simulation model user to select a desired traffic signal control algorithm. The road section labels show the points of importance, labelled A, B, C and D. Point A represents the entry point to the road section. This point also serves as an entry point to the system itself. Point B represents a lane change point, where vehicles may leave the labelled road section to enter the road section running parallel, in preparation to turn right at the intersection. Point C represents the stopping point along the road section. Point D represents both the destination point of the highlighted road section as well as the entry point of the adjoining road section.

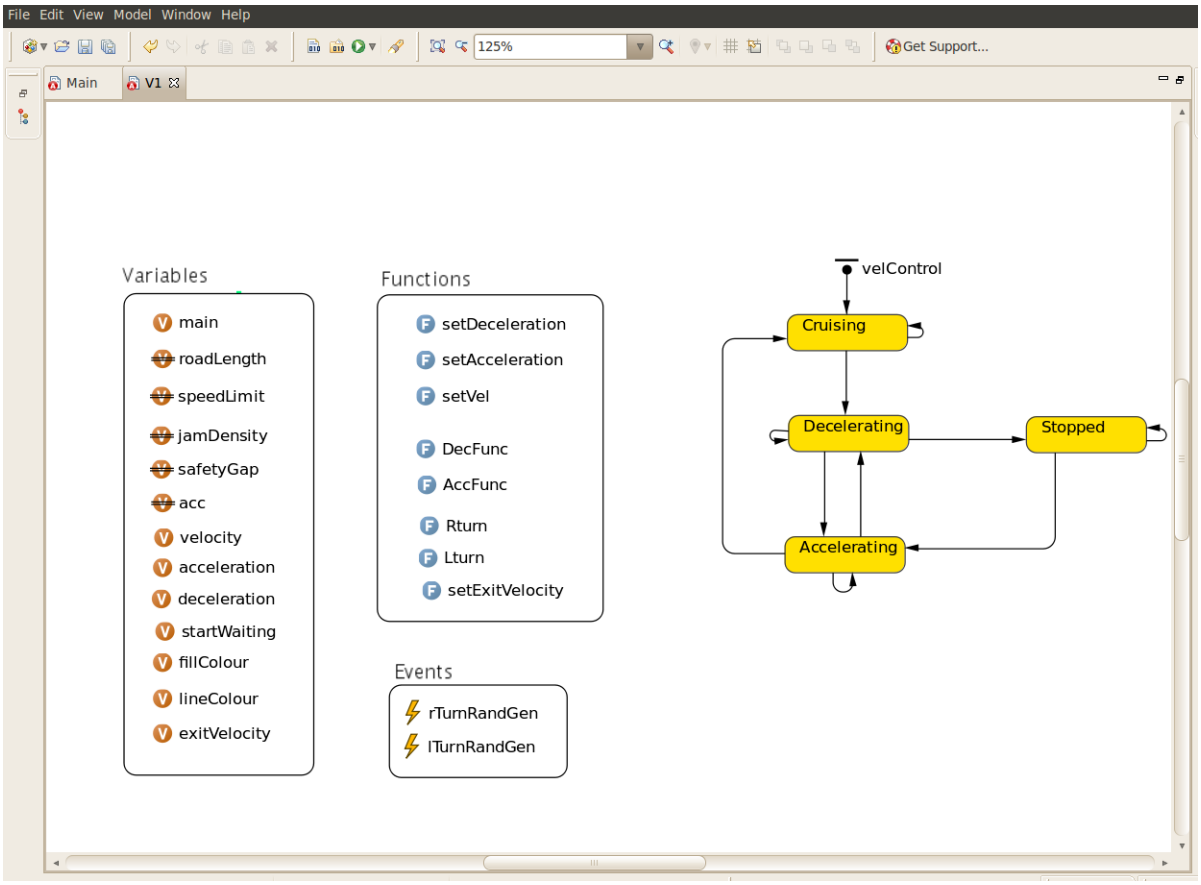


Figure 4.3: Model snap shot showing a general template responsible for the control of each vehicle generated in the system. The state chart responsible for the vehicle acceleration may be seen towards the right of the figure. The values associated with each variable are user-defined.

be generated according to some random distribution, or according to an explicit predetermined arrival profile. A *start-up* code as well as an *upon arrival* code is associated with each vehicle template. The *start up* code is responsible for initialising the newly generated vehicle's position along the road section as well as its speed. It is also responsible for providing the vehicle with its destination point towards which it must travel, as well as the path along which it must travel to reach its destination point. The *upon arrival* code is responsible for the implementation of the events that occur when a vehicle reaches its destination point. If the vehicle's destination point happens to be at an exit point of the system, then the *upon arrival* code is simply responsible for deleting the vehicle from the system. However, if the vehicle's destination point coincides with the entry point of another road section (*i.e.* there is a road section adjoining the section on which the vehicle is currently positioned), then the *upon arrival* code is responsible for generating a new vehicle to be part of the adjoining road section vehicle list, which has the identical properties of the vehicle which has just reached its destination point in terms of colour and speed — such that it appears to be the same vehicle. This is all implemented before the vehicle which has just reached its destination point is almost simultaneously deleted from the list of vehicles travelling along the road section on which it has just reached the destination point.

The process during which information about a vehicle in one list is passed on to a vehicle in an adjoining list makes use of both variables and functions. When a vehicle reaches its destination

point, a function is called as part of the *upon arrival* code. This function sets the value of the variable labelled *exitVelocity* to that of the vehicle's speed at the destination point, and sets the velocity of the newly generated vehicle on the adjoining road section equal to the value of *exitVelocity*. In a similar manner, the colour of the newly generated vehicle is set to match the colour of the vehicle at the destination point.

An integral part of any traffic simulation model is the successful modelling of the turning of the vehicles. To implement turning, as well as lane changes in preparation to turn, the model makes use of events that call functions responsible for deciding whether a vehicle would turn or not, and similarly whether it would change lane in preparation to turn or not. An event is triggered by a vehicle's arrival at a specific turning point along the road section. When the event is triggered, its action is to generate a random number between zero and one according to a uniform distribution. This random number is then used to determine whether the vehicle carries on travelling towards its destination point or "turns," based on a user-specified fraction of vehicles that are expected to turn. The act of turning follows the same procedures as when a vehicle crosses an intersection onto an adjoining road section. The turning vehicle is deleted from its current road section but not before a new vehicle of the same colour and with the same speed has been generated for the road section onto which the vehicle is turning.

Between the entry point of a road section and either a destination, turning, or lane change point, the speed of a vehicle along it is determined by which state it is in, as was described in §4.1.3.

4.2.1 Model performance measures

As well as being responsible for the simulation of traffic in a network, the model is also responsible for gathering relevant information pertaining to the performance of the system being simulated and displaying these data in a clear, readable format. AnyLogic facilitates the linking of the simulation model to an Excel spreadsheet in such a way that data may be read into the model from the spreadsheet as well as allowing the model to write data to the spreadsheet. The writing of data to an Excel spreadsheet is implemented by calling functions which are triggered at specific points in time during the model's execution or by specific events.

The performance measures that are used to gauge the effectiveness of the various traffic control algorithms investigated include the average waiting times of the vehicles in the system for the duration of the simulation study, the maximum waiting time experienced by any vehicle in the system, the total or sum of average queue lengths along all road sections in the system, the average time a vehicle spends in the system as a whole, both travelling and queued, as well as the maximum time any one vehicle spends in the system.

Measuring the time a vehicle spends in the system

In order to measure the time each vehicle spends in the system, the time the vehicle spends along each road section on which it travels is measured and these times are then summed together. This is due to the fact that a vehicle is actually deleted from one list while it is added to another as it appears to transition from one road section to another. To accomplish this, each vehicle has three variables associated with it, one which stores the time at which the vehicle enters onto the road section, another which stores the time it has spent on the current road section upon reaching the destination point, and another that stores the time it has spent on preceding road sections. When a vehicle enters onto a road section, the time at which it does so is assigned to a

variable labelled *enterTime*. When a vehicle reaches the destination point of a road section, the time it spent on the road section is calculated by subtracting the value stored by the *enterTime* variable from the time at which it reaches the destination point. This value is stored in a variable labelled *timeOnSection*. The amount of time that a vehicle has spent on other road sections while in the system (if any) is stored in a variable called *timeSoFar*. One of two events occur when a vehicle reaches the destination point of a road section, depending on whether the vehicle is due to enter onto another road section or whether the vehicle exits the system. If the vehicle is to enter onto another road section, then the value of its *timeSoFar* variable is added to the value of its *timeOnSection* variable and this value is then stored by the newly generated vehicle on the adjoining road section as its *timeSoFar* variable. If the vehicle exits the system upon reaching the destination point of a road section, then the sum of its *timeOnSection* and *timeSoFar* variables is recorded as the vehicle's total time spent in the system.

Measuring vehicle queue dynamics

The ability of a traffic signal control algorithm to reduce vehicle waiting times and alleviate congestion by minimising vehicle queue lengths is central to its performance and effectiveness. In the simulated environment it was necessary to specify when a vehicle is, in fact, queued. For the model implemented, a vehicle was considered to be queued if it was not travelling at its desired speed, *i.e.* the speed limit along the road section, since vehicles always attempt to travel at the speed limit. This choice is motivated by noting that a driver's frustration typically starts to mount when traffic is so congested that it prevents him/her from driving at a speed associated with free flowing traffic. Given this classification of which vehicles are queued, it was necessary for the simulation model to be able to measure the amount of time that a vehicle spends queued while in the system, as well as the queue lengths along all road sections approaching an intersection.

In order to measure the amount of time that vehicles spend in a queued state, a variable called *startWaiting* is associated with each vehicle in such a way that upon a vehicle's transition from the cruising state to the decelerating state, the time at which the transition occurred is stored in the *startWaiting* variable. From that transition onwards the vehicle is considered to remain queued until its speed is once again the speed limit. With the vehicle's transition from the accelerating state back to the cruising state, the time it spent queued is calculated by subtracting the value of the *startWaiting* variable from the time at which the transition occurred. This process is repeated each time a vehicle enters and departs a queue.

Measurement of queue lengths occurs in a similar manner. A variable responsible for storing the length of a queue along a road section is introduced to the model, called *queueSize*. When a vehicle's state changes from one of cruising to one of decelerating, the value of *queueSize* is incremented by one. Analogously, when a vehicle's speed reaches that of the speed limit after accelerating out of a queue, or a vehicle traverses onto another road section, the value of *queueSize* is decremented by one.

4.2.2 Representation of recorded data

Once data have been recorded successfully it is necessary to extract as much information as possible from these data in order to draw accurate conclusions and inferences about the performance of the traffic system being modelled. AnyLogic facilitates this information extraction process by providing a comprehensive array of analysis tools. One such tool which was of utmost

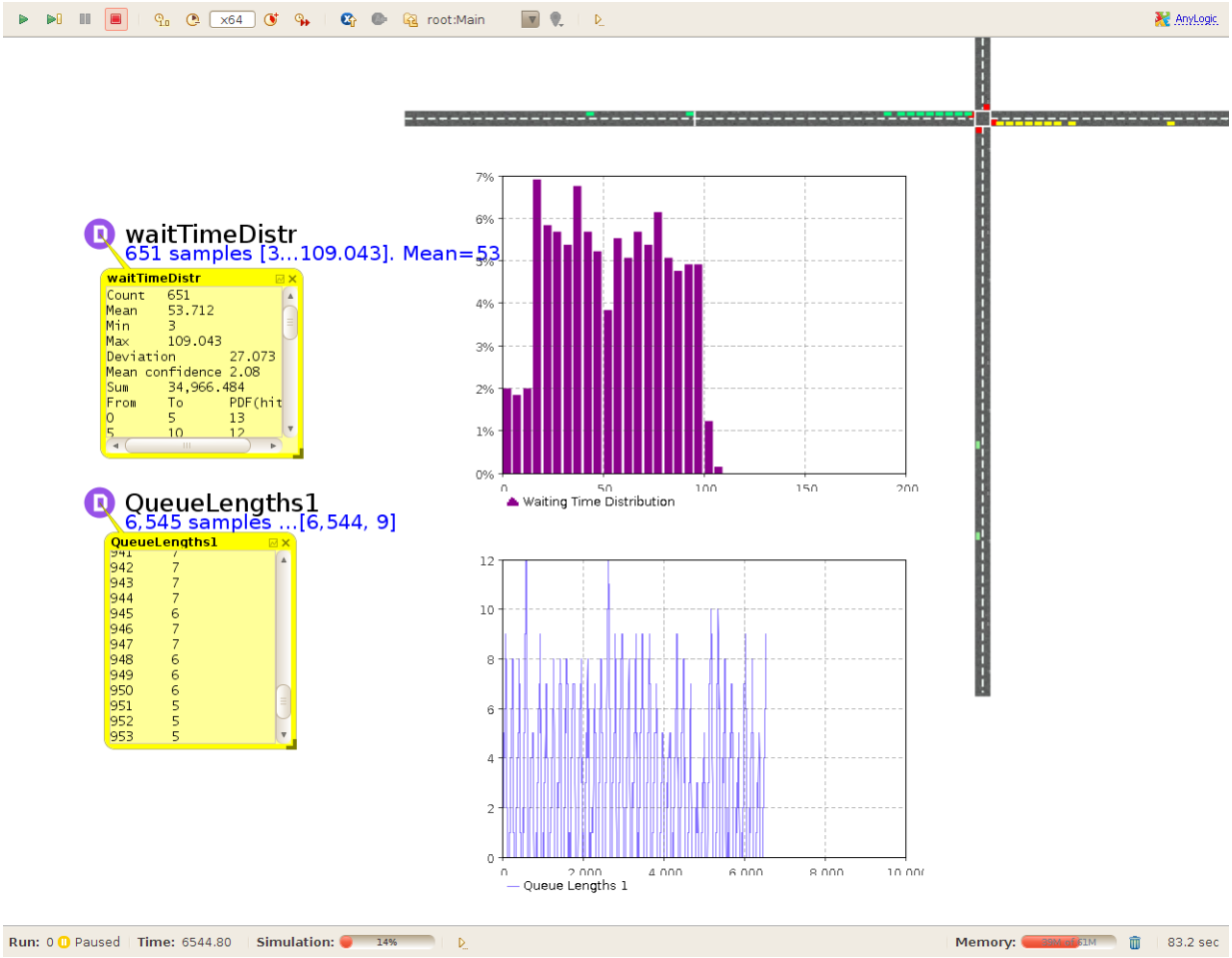


Figure 4.4: Model screen snapshot illustrating the dynamic representation of the system's performance measures. The object entitled *waitTimeDistr* is a *Histogram Data* object which is responsible for collecting the waiting times of all the vehicles in the system and performing statistical analyses on them, as well as providing information to the histogram to its right, labelled *Waiting Time Distribution*. The object labelled *QueueLengths1* is a *Statistics* object which is responsible for collecting the queue length along a road section at each time step of the simulation model and providing the information to the time plot alongside it, labelled *Queue Lengths 1* in such a way that a visual representation of the evolution of the queue may be presented to the simulation model user.

importance to the model data analysis process is the *Histogram Data* object. Collected samples are added to this *Histogram Data* object as they are recorded, and from here the object is responsible for performing standard statistical analysis on the data values being added, including the calculation of the mean, minimum, maximum, deviation, variance and mean confidence interval of the data it is provided with. It is also responsible for constructing an associated *probability distribution function* (PDF) and *cumulative distribution function* (CDF) on the fixed set of intervals provided by the user [48]. The data collected by the *Histogram Data* object are then visualised as a histogram chart which is visible during the model execution and is updated dynamically as the model runs, allowing the user to inspect the evolution of the system as time progresses. This method of collecting data and representing it in the form of a histogram was implemented when recording and visualising the waiting times experienced by vehicles in the system under investigation, as illustrated in Figure 4.4, as well as for the purpose of providing information on the distribution of green times experienced when investigating the various self-organising control algorithms presented in the following chapter.

Another analysis tool that was employed is the *Statistics* object, which calculates statistical information on a series of data samples. These data samples collected by the *Statistics* object may then be visualised using a number of standard charts. For the case of the vehicle queue lengths, the data samples are collected according to the processes described in the previous section. These data are then represented dynamically as a time plot during the running of the model, as illustrated in Figure 4.4.

The statistics provided in terms of the means, minimums and maximums of the data obtained from the system are used in comparing the performances of the various control algorithms. These values are exported to Excel spreadsheets, where they are then used to plot various graphs, allowing the simulation model user to inspect and compare the performances of the various control algorithms in terms of the various performance measures adopted. The results of these investigations are presented and discussed in a later chapter.

4.3 The effects of incorporating acceleration and deceleration

Incorporating vehicle accelerations and decelerations into a simulation model is not a trivial task. As with any simulation model there is a cost associated with improving model accuracy in terms of computational time. Incorporating acceleration of vehicles into a traffic simulation model greatly increases the complexity of the model, and thus results in longer computation times, depending on the size of the model. Nonetheless, vehicle accelerations form an integral part of traffic simulation at a microscopic level as they are associated with vehicle delays, both as vehicles decelerate to join a queue and accelerate out of a queue, as was discussed in §2.1.3.

4.3.1 Alternatives to incorporating vehicle accelerations from the literature

Due to the complex nature of modelling vehicle accelerations and decelerations, certain techniques have been introduced in the literature so as to effectively incorporate the delay times associated with them, without having to explicitly model them. For example, Allsop [2], investigates the expressions derived by Webster [47], Miller [29] and Newell [30, 31] for the average delay per vehicle at a signalised intersection. The work of the latter authors is based on a continuum model in which the individual properties of the vehicles, such as speed and position, are not considered, but rather the average and saturation flow-rates of the individual road sections adjoining an intersection. In [2] expressions are presented for delays for various vehicle arrival

and departure models. The derivation of each of the expressions uses both the aforementioned average and saturation flows of vehicles along a road section as well the timing parameters associated with the traffic light controls at the relevant intersections.

Allsop [2] also documents various findings and assumptions with respect to the arrivals and departures of traffic to and from intersections by other authors. One such observation, made by Greenshields, Schapiro and Erickson [18], was that upon the commencement of a green signal at an intersection, allowing a queue of vehicles to depart from the intersection from rest, it was noted that the intervals between the departures across the stop line of successive vehicles after the sixth vehicle, were on average equal, provided that a queue was present. As was discussed in §2.1.3, the intervals between the first several vehicles were larger, due to the reaction time of the driver of the first vehicle to the commencement of the green signal and because the vehicles are still accelerating when they cross the stop line. A method proposed for incorporating these delays without modelling the actual accelerations of vehicles out of a queue at an intersection, is to assume that *all* queued vehicles depart after equal time intervals of $1/s$ at their desired speeds, where s is the saturation flow rate of the road section, with the front vehicle being delayed for a time period equal to the sum of the times by which the departure intervals of the first few vehicles each exceed $1/s$ after commencement of the green signal.

A similar approach is followed by Lämmer and Helbing [25]. However, instead of delaying the departure of the first vehicle in a queue at an intersection after the commencement of a green signal, they extend the intergreen time, *i.e.* the amber time and all red time associated with a traffic signal cycle, in an attempt to incorporate the delays associated with a queue discharging from rest.

4.3.2 Model comparison with and without vehicle accelerations

During model conception, it was decided that vehicle accelerations and decelerations would be incorporated explicitly, despite the extra programming and logic required to implement them in an effort to improve the realism of the model. In an attempt to validate the decision of incorporating vehicle accelerations and decelerations, a comparison was carried out in which a simulation of a single intersection was implemented, with vehicles approaching and departing from it in all four directions, along single lane road sections. For the model incorporating accelerations, the accelerations were calculated and implemented according to the logic presented in §4.1.

For the case of the model in which vehicle accelerations are not implemented, any vehicle travels at its desired speed, *i.e.* the speed limit $v^{\max}$, until it comes to the stopping point of a road section at an intersection indicating a red signal if it is the front vehicle along a road section, or if it comes within $1/k^{\text{jam}}$ metres of a stationary vehicle in front of it (if it is not the front vehicle), at which point the vehicle comes to an immediate stop. Upon the commencement of a green signal, the front vehicle of the queue (if a queue is present) departs immediately with a speed of $v^{\max}$, with each subsequent vehicle departing when the distance to the vehicle in front of it is at least $\frac{1}{k^{\text{jam}}} + v^{\max}T$ metres.

To ensure a fair comparison of both models, all variables and characteristics of the models were kept the same. The inter-arrival times between all vehicles entering the system at each of the four entry points were modelled according to a displaced exponential distribution, as suggested

by Newell [30]. The probability density function of the inter-arrival times is then given by

$$f(x,\lambda) = \begin{cases} \lambda e^{-\lambda(x-T)}, & \text{if } x \geq T, \\ 0, & \text{if } x < T. \end{cases} \tag{4.8}$$

This displaced exponential distribution ensures a minimum inter-arrival time of T seconds between consecutive vehicles entering the system. It corresponds to a Poisson arrival process with an arrival rate of λ, interrupted immediately after each arrival by a period of T seconds, during which all arrivals that would result from the Poisson process are completely ignored.

The value of T was set to 2 seconds, while the speed limit, $v^{\max}$ on all road sections was imposed at 14 metres per second, (which is approximately 50 kilometres per hour). The jam density k^{jam} along all road sections was taken as 0.2 vehicles per metre, resulting in each vehicle having an *effective* length of $1/k^{\text{jam}} = 5$ metres along the road section. For the comparison, turning of the vehicles was not incorporated into the models, as the objective was purely to investigate the effects of vehicle accelerations, particularly for the case of vehicle queue discharges.

The model incorporating vehicle accelerations was run first in order to investigate the delay due to the finite accelerations of the vehicles as they departed from a queue at the commencement of a green light. The simulation model was run until 100 queues of at least 20 stationary vehicles had formed at a red traffic signal and had departed upon the commencement of a green traffic signal. A fixed-time signal cycle was implemented with a green time of 65 seconds, an amber time of 3 seconds, and an all-red time of 2 seconds for each conflicting traffic flow. The headways between each successive pair of vehicles were recorded as the vehicles crossed the stopping point of a road section, as well as the vehicles' speed as they crossed the stopping point. It was found, on average, that every vehicle after the fourteenth vehicle crossed the stopping point travelling at the speed limit, and thus the headways between the vehicles after the fifteenth vehicle were no longer affected by finite acceleration. To calculate the approximate delay due to accelerations each time a queue is discharged, the average headway between successive vehicles from vehicle fifteen to vehicle twenty was calculated. This average was then subtracted from the average headways between the first fourteen successive vehicles to cross the stopping point, with the differences being summed to produce the total delay. The average headway between vehicles after the fourteenth vehicle was found to be 2.8 seconds (which corresponds to the average headway between all vehicles for the model without vehicle accelerations). The sum by which the first 14 headways exceeds this average headway value is assumed to represent the delay associated with finite acceleration of the queued vehicles and was calculated to be approximately 3 seconds.

A plot of the average headways is shown in Figure 4.5. The first bar represents the time between the crossing of the first vehicle and the second vehicle across the stopping point, and similarly, the second bar represents the time between the crossing of the second vehicle and the third vehicle across the stopping point, and so on. The reason that the headway between the commencement of the green signal and the crossing of the stopping point by the first vehicle was not included is that it was assumed that the first vehicle reacts immediately to the green signal, resulting in a very short headway. The blue bars represent the headways of the vehicles that are still accelerating as they cross the stopping point, while the green bars represent the headways of the vehicles that cross the stopping point travelling at the speed limit. The horizontal red line represents the average headways of those vehicles that cross the stopping point travelling at the speed limit.

With the delay due to acceleration calculated, it was possible to run and compare both simulation models, with and without the explicit incorporation of acceleration and deceleration.

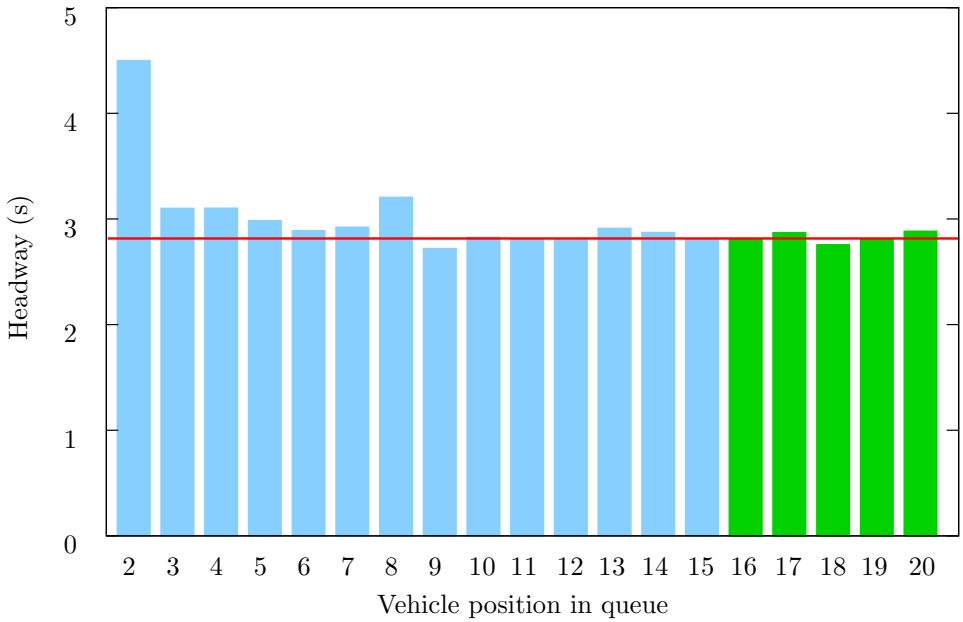


Figure 4.5: Average headways of the first 20 vehicles in a queue (excluding the front vehicle) discharging from rest at the commencement of a green signal.

The models were run for varying values of λ (vehicle arrival rates), with each run lasting the equivalent of 13 hours, including a one-hour warm-up period. The optimal green times for each λ were implemented for both models, with the amber time of each model being set to 3 seconds. The all-red time of the model with acceleration was set to 2 seconds, while that of the model with no acceleration was taken as 5 seconds, with the additional 3 seconds accounting for the delay due to acceleration, calculated earlier. The performance measures investigated included the average and maximum waiting times of the vehicles in the system, and average and maximum time spent by vehicles in the system, as well as the total average queue lengths. The results of both simulation models together with their interpretations are presented in the following section.

4.3.3 Simulation results and interpretations

The results of the simulation runs are presented in Figure 4.6. The results shown are for λ-values of 0.05, 0.1, 0.15, 0.2 and 0.25, used to generate vehicle arrivals according to the distribution in (4.8). The results include comparisons of the average and maximum waiting times experienced by the vehicles, the average and maximum time spent in the system by vehicles, as well as the average total queue lengths along all the road sections for both simulation models.

From Figure 4.6(a) it may be seen that there is an average increase of approximately 20% between the average waiting times experienced by vehicles in the model for which vehicle accelerations have been incorporated compared to the model for which they have not. A possible explanation of this result is that, in spite of the fact that an additional three seconds have been added to the all-red phase of the traffic signal cycle to account for the delay due to finite accelerations for the simulation model incorporating vehicle accelerations, it may be seen that these additional 3 seconds only account for the delays caused by acceleration before a vehicle crosses the stopping point of a road section, and do not consider the time that a vehicle continues to accelerate once it has passed the stopping point. Attempts at accounting for the time delay due

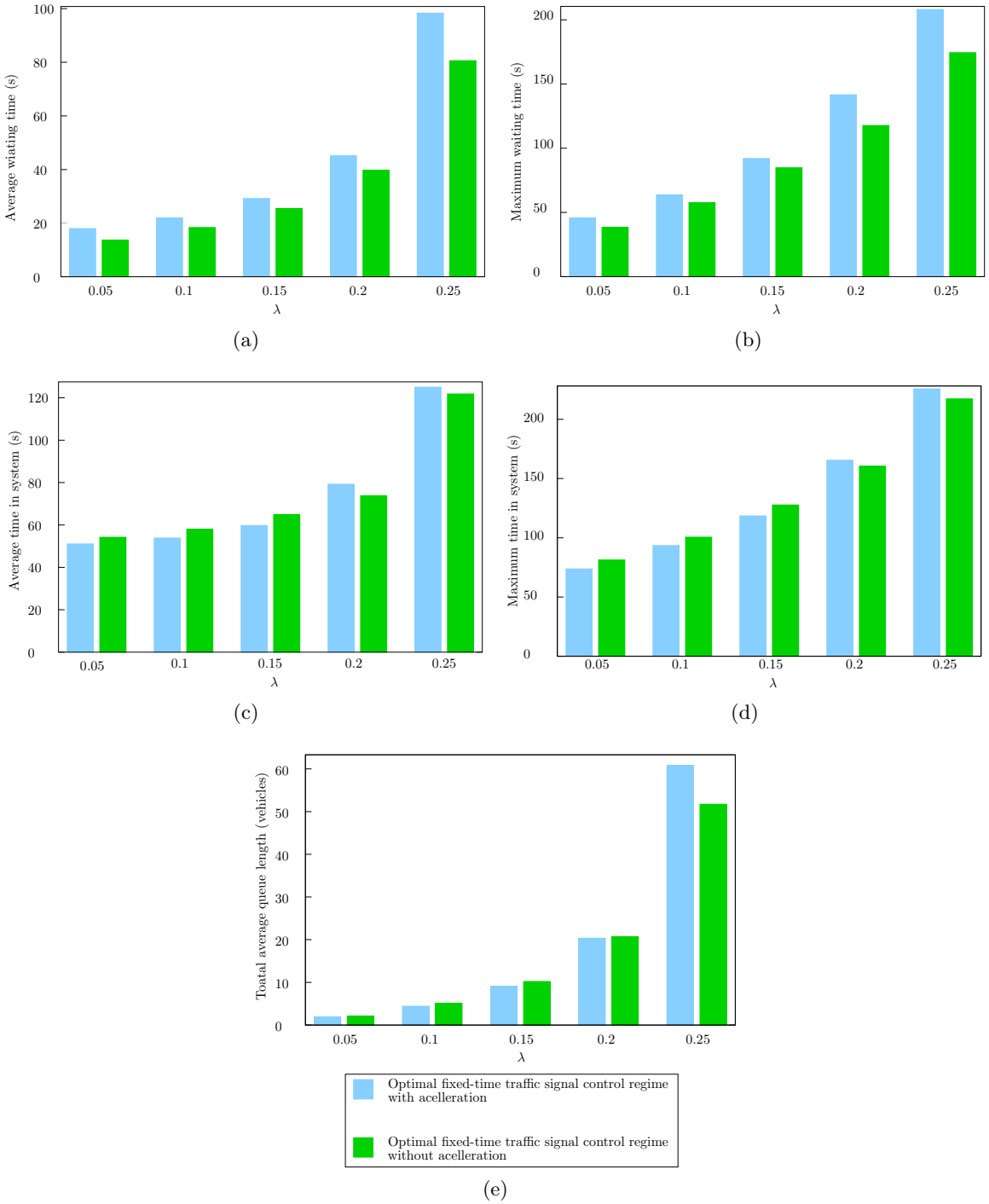


Figure 4.6: (a) The average waiting times of vehicles in the system, (b) the maximum waiting times experienced by vehicles in the system, (c) the average time spent in the system by vehicles, (d) the maximum time spent in the system by vehicles, and (e) the total average queue lengths for the system.

to finite acceleration past the stopping point (or any fixed point for that matter) are expected to be considerably more challenging since vehicles typically reach the speed limit, and thus cease to accelerate. A similar explanation may be offered for the average increase of approximately 16% in the maximum waiting times experienced by vehicles in the simulation model incorporating vehicle accelerations, as is illustrated in Figure 4.6(b).

When considering the average and maximum time spent in the system, as shown in Figures 4.6(c) and 4.6(d) respectively, it may be seen that the vehicles in the simulation model in which acceleration has not been incorporated explicitly spend slightly longer times in the system for λ-values of 0.05, 0.1 and 0.15. For λ-values of 0.2 and 0.25, the vehicles in the simulation model in which acceleration has been incorporated explicitly tend to experience longer times present in the system.

A possible explanation for these results, is that the values of λ for which the average and maximum time spent in the system by the vehicles in the simulation model which does not incorporate accelerations explicitly, exceed those of the vehicles in the simulation model which does incorporate acceleration, correspond to relatively short, total average queue lengths as may be seen in Figure 4.6(e). As was mentioned in §4.3.2, the additional 3 seconds appended to the all-red time of the traffic signal cycle for the model incorporating acceleration, account for the delays due to acceleration of the first 15 vehicles of a queue. Thus, if a queue is shorter than approximately 15 vehicles, as is the case for λ-values of 0.05, 0.1 and 0.15, the additional 3 seconds may be excessive, resulting in unrealistically long times spent in the system. For the λ-values of 0.2, 0.25, however, it may be seen that an underestimation of delay due to finite acceleration results in the vehicles of the simulation model incorporating accelerations experiencing longer times in the system.

Based on the above findings, it was concluded that while it is simpler and computationally more efficient to model a real-world traffic system without explicit vehicle accelerations, the approaches typically adopted in the literature to account for delays due to vehicle accelerations are not sufficient to accurately represent such a real world system. For this reason, an attempt was made to incorporate vehicle accelerations and decelerations explicitly into the traffic simulation model, so as to improve its realism and the validity of the results it produces.

It is expected that the differences between the green and the blue bars in Figure 4.6 would be more pronounced for more complex road topologies due to the fact that the under-estimations in time delays as a result of finite acceleration would be magnified as the number of intersections increased. This is because each vehicle would be experiencing a delay due to finite acceleration each time it was required to queue at an intersection, resulting in an accumulating effect.

4.4 Model Verification

According to Law [26], verification of a simulation model is concerned with determining whether the assumptions regarding the logic in a simulated representation of a real-world system have been correctly translated into a computer program, *i.e.* the successful debugging of a simulation model such that it is free of any errors of logic.

Because errors arise in the writing of simulation programming code, a basic model verification technique is the use of a commercial simulation package as a development platform [26]. The simulation model described in this chapter was implemented in the software suite AnyLogic University 6.5.0 [48] and as a result, the amount of programming was greatly reduced through both the program's user-friendly interface as well as well documented predefined functions and

procedures, thereby reducing the chances of coding errors.

An animated representation of the model described in this chapter forms an integral part of the model's verification. Animation allows for a visual inspection of the implementation of the model's logic [5, 26]. In particular, the animation of the model described in this chapter, allows for a verification that the output of the model sufficiently resembles that of real-world traffic flow, both along road sections and at intersections, in terms of the vehicles accelerating, decelerating and stopping correctly according to model logic, as well as ensuring that unrealistic events do not occur, *e.g.* vehicles passing through one another in an intersection.

Debugging of programming code underlying a simulation model is an essential part of the verification process, and regular debugging is considered to be good programming practice and an important aspect of successful simulation model building [5, 26]. In the building of the model described in this chapter, the built-in interactive run controller, or debugger of ANYLOGIC UNIVERSITY 6.5.0 scans the model code for discrepancies and syntax errors, upon each compilation of the model, before the model is run. If an error is found, attention is focused on the error type and its location in the code, and possible suggestions for rectifying the error are given. If the program code is found to be free of errors upon completion of model compilation, the model may be run. If an error is encountered during the running of the model, the simulation is suspended, and once again, attention is focused on the error type and its location in the code, and possible suggestions for rectifying the error are given. When building the model described in this chapter, an incremental approach was followed where, after each new addition of code to the model, the model was compiled and run, rectifying any coding errors where necessary. The model was first run in real time, *i.e.* 1 simulation second was taken equal to 1 real time second, to allow for a visual inspection of the correctness of the new code addition. Following this visual inspection, the model was run in virtual time *i.e.* the model was run at its maximum speed, for an equivalent of twenty four hours in real time to ensure that no errors arise during the simulation due to the new code addition.

One of the most useful verification tools is the use of a so-called *trace* [5]. It is advisable to have the simulation environment print a wide variety of output statistics as the simulation progresses, and to thoroughly analyse this output for reasonableness. With a trace, the state of the simulated system can be displayed at any time point for any chosen event. For example, a trace can be used to provide per-second updates on the status of each individual vehicle in the system in terms of the identification index of a vehicle, the current speed of the vehicle, the current acceleration or deceleration, as well as the vehicle's distance to the vehicle in front of it or the distance to the next intersection. This trace output, together with the animation output, allows for the logic being implemented to be verified as correct, as it facilitates an ability to test whether a vehicle decelerates or accelerates when it is supposed to do so, as well as an ability to test whether accelerations and decelerations are calculated correctly, by comparing them with hand calculations for the same scenarios.

4.5 Model Validation

Validation, according to Law [26], is the process of determining whether a simulation model is a sufficiently adequate representation of the real world system it is being used to investigate, taking into account the particular objectives of the study.

Like verification, validation of the simulation model described in this chapter was performed throughout the model's inception and building. During this process of validation the simulation

was inspected for continuity by altering the input parameters and testing whether the output produced changed accordingly and as expected. For example, an increase in the average arrival rate of vehicles to the system resulted in a corresponding increase in total average queue lengths as well as increased vehicle waiting times, provided that the traffic signal switching controls remained fixed and constant.

It is also necessary to validate the consistency of a simulation model. For the model described in this chapter, this was achieved by performing multiple simulation runs in which the input parameters remained the same, with the output of each simulation run being compared to ensure that the model produced *similar* results. The results are similar and not identical, for despite the fact that all input parameters remain the same for each simulation run, the inter-arrival times of the vehicles are generated stochastically according to an exponential distribution. It was found that for each set of input parameters, the model produced consistently similar results in terms of various output statistics, such as average waiting times experienced by the vehicles, average queue lengths, and the average amount of time spent in the system by the vehicles.

The next step towards validation of the model described in this chapter was comparing its performance against that of a real world system. The necessary data used in the experimental validation was taken from [44] for one intersection in Stellenbosch, namely the Adam Tas and Bird street intersection. These data were collected in fifteen minute intervals between 06:30 and 18:00 on a typical Tuesday during the university and school term by the Stellenbosch Traffic Department to ensure that the traffic counts are representative of the normal traffic conditions. The data consist of the number of vehicles arriving at each entry point to the intersection, together with the number of vehicles that turn left, right, or carry on through the intersection at each of the intersection entry points. As well as the arrival patterns, the specific traffic signal timing sequences are also available, as well as the total number of vehicles that left the intersection per time interval. In [44], Van der Merwe validated her traffic simulation model by comparing the actual number of vehicles departing the system at different times of the day with the number of vehicles that leave the system in the simulation model. She used the arrival profiles recorded by the traffic department to model the vehicle arrivals to the system and instead of using the timings of each individual traffic signal cycle, she used a fixed representative cycle during which the length of each phase in the cycle is an average of all the recorded observations for that phase throughout the observation window. The same validation approach is followed in this thesis. The recorded data may be found in Appendix A. A total of 100 simulation runs were implemented, and the average number of total vehicles which passed through the intersection after each hour according to these simulation runs are compared to the real world data in Table 4.1. From these results it may be seen that the simulation model built provides a sufficiently adequate representation of the real world system it replicates, allowing for valid and representative results.

4.6 Summary

This chapter aims to provide the reader with an understanding of the simulation model implemented in this thesis and its inner workings. In §4.1 the logic responsible for the various aspects of the simulation model was described, including how the road network, the traffic lights, and the vehicles themselves were modelled. In §4.2 the implementation of this logic as a working computer simulation model was described as well as the performance measures adopted for gauging the effectiveness of traffic control algorithms to be tested in the next chapter. In §4.3 the decision to incorporate vehicle accelerations into the model was motivated by means of a

simple simulation experiment comparing a model in which vehicle accelerations are incorporated to one in which they are not. Finally, in §4.4 and §4.5 various methods adopted for the purposes of verification and validation of the model were described.

Table 4.1: Validation Results for the Adam Tas and Bird Street Intersection in terms of the percentage error between the real world vehicle observations and the throughput of vehicles in the simulation model.

After 1 hour			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	255	256	0.39%
AT Straight	503	502.81	–0.04%
AT Right	7	7.19	2.71%
BS Left	5	7	40.00%
BS Straight	216	216.14	0.06%
BS Right	99	97.86	–1.15%
R44 Left	195	194	–0.51%
R44 Straight	625	631.95	1.11%
R44 Right	128	121.05	–5.43%
N1 Left	31	32	3.23%
N1 Straight	447	451.64	1.04%
N1 Right	347	342.36	–1.34%
Total	**2858**	**2860**	**0.07%**

After 2 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	449	449	0.00%
AT Straight	986	986.36	0.04%
AT Right	20	19.64	–1.80%
BS Left	16	18	12.50%
BS Straight	539	538.56	–0.08%
BS Right	302	301.44	0.19%
R44 Left	555	554	–0.18%
R44 Straight	1286	1292.65	0.52%
R44 Right	279	271.97	–2.52%
N1 Left	77	77	0.00%
N1 Straight	982	984.66	0.27%
N1 Right	667	664.34	–0.40%
Total	**6158**	**6157.62**	**–0.01%**

After 3 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	672	672	0.00%
AT Straight	1363	1362.56	–0.03%
AT Right	37	37.44	1.19%
BS Left	36	38	5.56%
BS Straight	825	826.34	0.16%
BS Right	465	462.66	–0.50%
R44 Left	762	761	–0.13%
R44 Straight	1643	1649.62	0.40%
R44 Right	367	360	–1.91%
N1 Left	133	133	0.00%
N1 Straight	1436	1438.21	0.15%
N1 Right	908	905.79	–0.24%
Total	**8647**	**8646.62**	**0.00%**

After 4 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	872	872	0.00%
AT Straight	1742	1741	–0.06%
AT Right	56	57	1.79%
BS Left	46	48	4.35%
BS Straight	1136	1136.9	0.08%
BS Right	626	624.1	–0.30%
R44 Left	963	962	–0.10%
R44 Straight	1965	1968.16	0.16%
R44 Right	445	440.46	–1.02%
N1 Left	186	187	0.54%
N1 Straight	1829	1831.87	0.16%
N1 Right	1148	1145.13	–0.25%
Total	**11014**	**11013.62**	**0.00%**

After 5 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	1086	1086	0.00%
AT Straight	2152	2151.11	–0.04%
AT Right	68	68.89	1.31%
BS Left	61	63	3.28%
BS Straight	1440	1440.79	0.05%
BS Right	804	802.21	–0.22%
R44 Left	1153	1152	–0.09%
R44 Straight	2285	2290.21	0.23%
R44 Right	532	526.41	–1.05%
N1 Left	249	249	0.00%
N1 Straight	2184	2187.46	0.16%
N1 Right	1366	1362.54	–0.25%
Total	**13380**	**13379.62**	**0.00%**

After 6 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	1298	1298	0.00%
AT Straight	2524	2522.39	–0.06%
AT Right	90	91.61	1.79%
BS Left	75	77	2.67%
BS Straight	1767	1768.65	0.09%
BS Right	985	982.35	–0.27%
R44 Left	1351	1350	–0.07%
R44 Straight	2616	2621.69	0.22%
R44 Right	620	613.93	–0.98%
N1 Left	327	327	0.00%
N1 Straight	2514	2516.07	0.08%
N1 Right	1625	1622.93	–0.13%
Total	**15792**	**15791.62**	**0.00%**

After 7 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	1531	1531	0.00%
AT Straight	2938	2936.64	–0.05%
AT Right	111	112.36	1.23%
BS Left	101	102	0.99%
BS Straight	2140	2141.56	0.073%
BS Right	1199	1196.44	–0.21%
R44 Left	1553	1552	–0.06%
R44 Straight	3008	3014.48	0.22%
R44 Right	703	696.14	–0.98%
N1 Left	387	387	0.00%
N1 Straight	2919	2920.14	0.04%
N1 Right	1893	1891.86	–0.06%
Total	**18483**	**18481.62**	**–0.01%**

After 8 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	1751	1751	0.00%
AT Straight	3341	3339.59	–0.04%
AT Right	133	134.41	1.06%
BS Left	120	121	0.83%
BS Straight	2508	2509.17	0.047%
BS Right	1410	1407.83	–0.15%
R44 Left	1742	1741	–0.06%
R44 Straight	3378	3385.21	0.21%
R44 Right	804	796.41	–0.94%
N1 Left	412	412	0.00%
N1 Straight	3176	3176.69	0.02%
N1 Right	2080	2079.31	–0.03%
Total	**20855**	**20853.62**	**–0.01%**

After 9 hours			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	1995	1995	0.00%
AT Straight	3805	3803.24	–0.05%
AT Right	146	147.76	1.21%
BS Left	139	140	0.72%
BS Straight	2918	2920.06	0.07%
BS Right	1629	1625.94	–0.19%
R44 Left	1940	1939	–0.05%
R44 Straight	3803	3810.38	0.19%
R44 Right	901	893.24	–0.86%
N1 Left	464	464	0.00%
N1 Straight	3542	3541.78	–0.01%
N1 Right	2296	2296.22	0.01%
Total	**23578**	**23576.62**	**–0.01%**

Entire Day (11.5 hours)			
Vehicle Count	Collected Data	Simulation Model	% Error
AT Left	2659	2659	0.00%
AT Straight	5028	5025.97	–0.04%
AT Right	194	196.03	1.05%
BS Left	173	175	1.16%
BS Straight	4202	4316.92	2.73%
BS Right	2415	2299.08	–4.80%
R44 Left	2559	2558	–0.04%
R44 Straight	4815	4820.37	0.11%
R44 Right	1176	1170.25	–0.49%
N1 Left	593	593	0.00%
N1 Straight	4736	4735.73	–0.01%
N1 Right	2820	2820.27	0.01%
Total	**31370**	**31369.62**	**0.00%**

CHAPTER 5

Algorithms Tested and Results Obtained

Contents

This chapter contains a description of the various traffic signal control algorithms that were implemented and investigated. The inspiration and logic behind each algorithm as well as methods of implementation are presented together with the underlying assumptions of the algorithms. The inputs required by each algorithm are discussed as well as their benefits and short-comings.

The algorithms presented include an optimised fixed-time cycle-based control algorithm, while the self-organising algorithms include one such approach which attempts to maximise the number of vehicles through an intersection in the allocated amount of green time, while another attempts to serve a vehicle stream which most urgently requires service, based on the collective distances from the intersection and speeds of the vehicles along the road section.

The various traffic control algorithms are tested for a variety of traffic network topologies and are compared in terms of different performance measures. In addition to the various topologies tested, a real case study is performed on the Adam Tas Road & Bird Street intersection in Stellenbosch, South Africa. The results obtained from each simulation run are presented in §5.3 together with an analysis and interpretation of their meanings.

5.1 The traffic control algorithms implemented

The algorithms that were implemented in the simulation model of Chapter 4 are described in this section. The first algorithm is a fixed-time cycle-based algorithm which acts as a benchmark for the experiment against which the performances of the self-organising traffic control algorithms may be measured. There are two different self-organising traffic control algorithms, each of which combines an optimising strategy with a stabilising strategy. The performance of each of these combined self-organising traffic control strategies is tested, as well as the performances of the individual optimisation and stabilisation strategies.

5.1.1 An optimised fixed-time cycle-based traffic signal control algorithm

An *optimised fixed-time cycle-based traffic signal control algorithm*, abbreviated here as OFTTCA acts as a gauge for which the effectiveness of the self-organising algorithms may be measured, as they are expected to perform at least as well as an optimised fixed time traffic signal control algorithm. The algorithm is free of any vehicle-actuated response, with the same phase timings being implemented, regardless of the current traffic situation. The algorithm can accommodate multiple phases, with the time allocated to each phase being selected before the simulation is run. Each phase is represented by a state in a state chart, responsible for controlling the signal indicated by a set of traffic lights, as explained in §4.1.2. A transition occurs from one state to another when the time elapsed after entering the state equals that of the predetermined time allocated to the specific traffic signal phase.

The pseudo-code listing of this simple fixed-time cycle based traffic signal control algorithm is presented in Algorithm 5.1. The cycle is comprised of four phases, namely (in the sequence in which they occur), a green phase, an amber phase, an all-red phase, and a red phase.

Algorithm 5.1: Four-phase fixed-time cycle-based traffic signal control

```
1  Green signal commences;
2  if Time elapsed since commencement of green signal = greenTime then
3  |  Display amber signal;
4  else
5  |  remain green;
6  if Time elapsed since commencement of amber signal = amberTime then
7  |  Display all-red signal;
8  else
9  |  remain amber;
10 if Time elapsed since commencement of all-red signal = allRedTime then
11 |  Display red signal;
12 else
13 |  remain all-red;
14 if Opposing traffic signal is in red phase then
15 |  Display green signal;
16 else
17 |  remain red;
```

In Algorithm 5.1, *greenTime, amberTime, and allRedTime* are the user-defined times responsible

for the lengths of the green, amber and all-red phases of the traffic signal cycle, respectively.

When referring to an *optimised* fixed-time cycle-based traffic control algorithm, it is meant that the only parameter that requires optimisation in order to select its value is the length for which a green signal would be shown, with amber, red, and all-red phase times remaining constant.

AnyLogic has a built-in parameter variation tool, which affords the opportunity to run a simulation model with different parameters continuously, without having to manually change the model parameters each time, which allows the user to analyse how certain parameters affect the performance of the model [48]. The parameters to be varied are chosen and their ranges and increments specified. The simulation is then run for each combination of the parameters to be varied. For the case of finding the optimal green time for a fixed-time cycle-based traffic control algorithm, the parameters to be varied are the green time of the cycle as well as the parameter λ associated with the exponential distribution of vehicle inter-arrival times to the system. It is assumed that the value of the green time may only take on integer values. After running through each parameter combination, the model is able to output an optimal value of each performance measure considered, for each value of λ together with the associated optimal green time.

5.1.2 Self-organising traffic signal control algorithms

Both self-organising algorithms selected for implementation assume the structure suggested by Lämmer and Helbing [25], as described in §2.4.2. That is, they combine an optimising prioritisation strategy with that of a stabilisation strategy. The fundamental difference between the two algorithms occurs in the optimisation strategy. More specifically, the algorithms differ in the formulation of the dynamic priority indices, $\pi_i(t)$, associated with each traffic flow i approaching the intersection at time t. These differences are elaborated upon in the following descriptions of the two algorithms.

Self-organising traffic control algorithm I

The first self-organising traffic control algorithm modelled and implemented is abbreviated as SOTCA I and incorporates the same prioritisation function as that proposed by Lämmer and Helbing [25], as described in §2.4.2. The priority index π_i of traffic flow i, approaching the intersection is given in (2.25). The value of π_i corresponds to the average service rate experienced by the traffic flow i during its setup period and anticipated green time period. Service is rendered to the traffic flow having the greatest priority index. To anticipate the amount of green time required to serve a traffic flow, the algorithm makes use of the information provided by the detection technology described in §1.2 assumed to be available to all traffic signalling equipment at the intersection. This technology provides information to the algorithm in terms of the number of vehicles within its detection range approaching the intersection, allowing for the estimation of the arrival rate of vehicles. A pseudo-code listing describing the method of the dynamic anticipation of green time is presented as Algorithm 5.2.

Following the notation in [25], Algorithm 5.2 first tests whether if traffic flow i is receiving service, as may be observed in Line 1, where $\sigma(t) = i$ if traffic flow i is receiving service at time t. If traffic flow i is not receiving service, the algorithm calculates the necessary length of green time that will be implemented upon the commencement of service to traffic flow i, given by $\hat{g}_i(t)$. This anticipated green time is calculated such that it allows for the already detected vehicles to be cleared together with the anticipated number of vehicles that are expected to arrive during

Algorithm 5.2: Green time anticipation

```
for each traffic flow i do
    if σ(t) ≠ i then
        ĝ_i(t) ← n_i(t)/Q_i^max + n_i(t)Q_i^arr(t)/(Q_i^max)^2 + τ_i Q_i^arr(t)/Q_i^max;
    else
        ĝ_i(t) ← ĝ_i(t) − 1;
```

this clearing of the detected vehicles and during the setup time associated with the changing of the traffic signals. Assuming that the vehicles depart from the intersection at a rate of $Q_i^{\max}$, the time required to clear the already detected vehicles is $n_i(t)/Q_i^{\max}$, where $n_i(t)$ represents the number of vehicles already detected, while the time required to clear the vehicles which are expected to require service during the setup and green time is $n_i(t)Q_i^{\text{arr}}/(Q_i^{\max})^2 + \tau_i Q_i^{\text{arr}}/Q_i^{\max}$, where τ_i represents the setup time associated with traffic flow i and $Q_i^{\text{arr}}(t)$ is the arrival rate of vehicles at the intersection along traffic flow i at time t.

While traffic flow i receives service, the amount of anticipated green time is decremented by one every second. This occurs because, even though vehicles still arrive along traffic flow i during service, they have already been accounted for in the calculation of the green time required while traffic flow i was not receiving service. It should be noted that Algorithm 5.2 is implemented on a per second basis.

The next step of the self-organising traffic control algorithm is to calculate the dynamic priority index of traffic flow i, which makes use of the value of the anticipated green time $\hat{g}_i(t)$. A pseudo-code listing associated with the determination of the dynamic priority indices is presented as Algorithm 5.3.

Algorithm 5.3: Calculating dynamic priority indices

```
for ∀ i do
    if σ(t) = i then
        π_i(t) ← n̂_i(t) / (τ_i(t) + ĝ_i(t));
    else
        π_i(t) ← n̂_i(t) / (τ_σ^pen + τ_i(t) + ĝ_i(t));
```

Again using the notation of Lämmer and Helbing [25], it may be seen in Algorithm 5.3 that if traffic flow i is receiving service at time t, *i.e.* if $\sigma(t) = i$, then the value of $\pi_i(t)$ corresponds to the anticipated number of vehicles to be served during the setup and green time of traffic flow i, that is, the number of vehicles which are expected to depart from the intersection at the maximum flow rate $Q_i^{\max}$ during the service period, given by $\hat{n}_i(t) = \hat{g}_i(t)Q_i^{\max}$. In the case where traffic flow i is not receiving service, *i.e.* when $\sigma(t) \neq i$, the expression for $\pi_i(t)$ is similar to the case where $\sigma(t) = i$, the only difference being that τ_σ^{pen} is added to the denominator. Here τ_σ^{pen} is considered a penalty term for terminating the current service process and represents the time losses associated with this termination. It can take on any value from 0 up to τ_σ^0.

As was mentioned in §2.4.2, service is provided to the traffic flow with the highest priority index value, provided that the prioritisation control strategy of the algorithm is responsible the service selection process and not the stabilisation strategy. The pseudo-code listing of the optimising prioritisation strategy of SOTCA I, abbrviated as OPS I, which makes use of the priority indices

calculated according to Algorithm 5.3 is presented as Algorithm 5.4.

Algorithm 5.4: Optimising prioritisation strategy I (OPS I)

```
1  for each traffic flow i do
2      if σ(t) ≠ i then
3          if π_i(t) > π_σ(t) then
4              σ(t) ← i;
5          else
6              σ(t) ← σ(t − 1);
```

The stabilisation strategy, abbreviated here as the SS, makes use of two critical threshold values, namely the critical queue length along traffic flow i, n_i^{crit}, and the maximum allowable green time which can be afforded to traffic flow i, namely $g_i^{\max}$. These values are determined such that they satisfy two important service interval time requirements, as was mentioned in §2.4.2. The first requirement is that each traffic flow shall be served once, on average, within a desired service interval of length $U > 0$. The second requirement is that each traffic flow shall be served at least once within a maximum service interval of length $U^{\max} \geq U$. These two parameters, U and $U^{\max}$, are the only two adjustable parameters of the control algorithm. To derive the values of n_i^{crit} and $g_i^{\max}$ the processes proposed in [25] are followed, as discussed in §2.4.2.

A pseudo-code listing of the stabilisation strategy is presented as Algorithm 5.5.

Algorithm 5.5: Stabilisation strategy (SS)

```
1  Ω ← ∅;
2  for each traffic flow i do
3      if n_i(t) ≥ n_i^crit(t) then
4          Ω ← Ω ∪ {i};
5      else
6          Ω ← Ω;
7      if i is the first element of Ω then
8          if n_i(t) = 0 or g_i(t) ≥ g_i^max then
9              Ω ← Ω\ {i};
10         else
11             Ω ← Ω;
```

In Line 3 of Algorithm 5.5, for each traffic flow i, the algorithm tests whether the detected number of vehicles requiring service along traffic flow i at time t, $n_i(t)$, exceeds $n_i^{\text{crit}}(t)$. If so, the argument of traffic flow i is added to the set labelled as Ω. A traffic flow receives service as soon as its argument is added to the set Ω, provided there are no other traffic flow arguments currently stored in that set, otherwise it is queued behind the previously stored traffic flow arguments in Ω. Only the traffic flow associated with the first element or the "head" of Ω receives service, so that all elements in the set Ω are served in a first-come-first-served manner. The argument of traffic flow i is removed from the set Ω after the queue of vehicles requiring service along it has been cleared, *i.e.* when $n_i(t) = 0$, or it has received a green signal for a time interval of length $g_i^{\max}$, as shown in Line 8 of Algorithm 5.5.

A pseudo-code listing of the combined control strategy may be seen in Algorithm 5.6. As may

Algorithm 5.6: Self-organising traffic control algorithm I (SOTCA I)

```
1 if Ω ≠ ∅ then
2 |   σ(t) ← head of Ω;
3 else
4 |   σ(t) ← arg max_i π_i(t);
```

be seen in Line 1 of Algorithm 5.6, the algorithm first testes whether the set Ω is empty. If this is the case, then service selection is determined by the prioritisation strategy of the algorithm, with the traffic flow having the highest priority index value receiving service. Otherwise the traffic flow associated with the first argument of the set Ω receives service. The prioritisation regime of the algorithm therefore aims to service all incoming traffic as quickly and efficiently as possible, with the stabilisation strategy only being involved when the prioritisation strategy fails to keep vehicle queue lengths below some critical value.

Because the two service interval parameters mentioned earlier, U and $U^{\max}$, are adjustable, the same optimisation process is employed with respect to the values of these parameters as was mentioned in §5.1.1. Again AnyLogic's parameter variation tool is used, with the values of U and $U^{\max}$ being altered with each simulation run in which self-organising traffic control algorithm I is implemented. The combination which results in the most favourable results for each performance measure and arrival rate is used for experimentation and a comparison of results.

Self-organising traffic control algorithm II

Self-organising traffic control algorithm II, abbreviated here as SOTCA II follows the same structure and ideology as the first control algorithm adapted from the work of Lämmer and Helbing [25] in that it combines a prioritising optimisation strategy with that of a stabilisation strategy. The only difference is in the definition of the priority indices of the individual traffic flows which are used by the optimising prioritisation strategy of the algorithm to determine which traffic flow receives service.

In the case of the second traffic control algorithm, the priority index of traffic flow i is defined as

$$\pi_i^{\mathrm{II}} = \sum_{j=1}^{n} \frac{1}{\mu + S_{j,\beta}^i}, \tag{5.1}$$

where μ is a positive constant and $S_{j,\beta}^i$ represents the distance of vehicle j ($j = 1, \ldots, n$) along traffic flow i to the stopping point, β of the road section. Here π_i^{II} may be interpreted as approximately the sum of the inverse of the distance of each detected vehicle in traffic flow i to the stopping point of the road section they are travelling on. The constant μ is added to the denominator to prevent the occurrence of very large numbers as the distance between a vehicle and the stopping point approaches zero. A pseudo-code listing of the optimising prioritisation strategy, which makes use of (5.1), is presented as Algorithm 5.7.

In Algorithm 5.7 the optimising prioritisation strategy II, abbreviated here as OPS II is presented. It may be seen that if traffic flow i is not currently selected for service, if *i.e.* $\sigma \neq i$, then if the value of π_i^{II} exceeds the value of the priority index of the traffic flow which is currently receiving service, π_σ^{II}, by a value of at least ϵ (a predefined constant), then service is provided to traffic flow i. If not, then service continues with respect to the currently selected traffic

Algorithm 5.7: Optimising prioritisation strategy II (OPS II)

```
for each traffic flow i do
    if σ(t) ≠ i then
        if π_i^II(t) > π_σ^II(t) + ε then
            σ(t) ← i;
        else
            σ(t) ← σ(t − 1);
```

flow. Algorithm 5.7 therefore places a priority on vehicles which are closer to the intersection, and therefore require service more urgently. The constant ϵ is included to prevent too frequent switching of service between traffic flows which would result in increased waiting time due to an increased number of setup phases that would be induced.

The stabilisation strategy of self-organising traffic control algorithm II is the same as that of self-organising traffic control algorithm I, and the reader is referred to Algorithm 5.5 for an illustration of a pseudo-code listing describing the stabilisation strategy.

Combining the optimising prioritisation regime proposed in Algorithm 5.7 with that of the stabilisation strategy proposed in Algorithm 5.5 results in the completed self-organising traffic control algorithm II, which is presented as Algorithm 5.8.

Algorithm 5.8: Self-organising traffic control algorithm II (SOTCA II)

```
if Ω ≠ ∅ then
    σ(t) = head of Ω;
else
    for for each traffic flow i do
        if σ(t) ≠ i then
            if π_i^II(t) > π_σ^II(t) + ε then
                σ(t) = i;
            else
                σ(t) = σ(t − 1);
```

Comparing self-organising traffic control algorithms I and II, it may be seen that algorithm I uses the detection technology to anticipate the amount of green time required to clear the vehicles detected and those expected to arrive during the service time. It places a priority on the number of vehicles that will be served during this service time, with the traffic flow maximising this value receiving service while accounting for time losses due to the setup phase in between each change in service between traffic flows. Algorithm II, on the other hand, uses the detection technology to track a vehicle's position along a road length, giving priority to vehicles which are closer to the intersection. The performances of the algorithms discussed in this section in terms of their ability to ease congestion and minimise waiting and commute times are compared later in this chapter for various road network topologies.

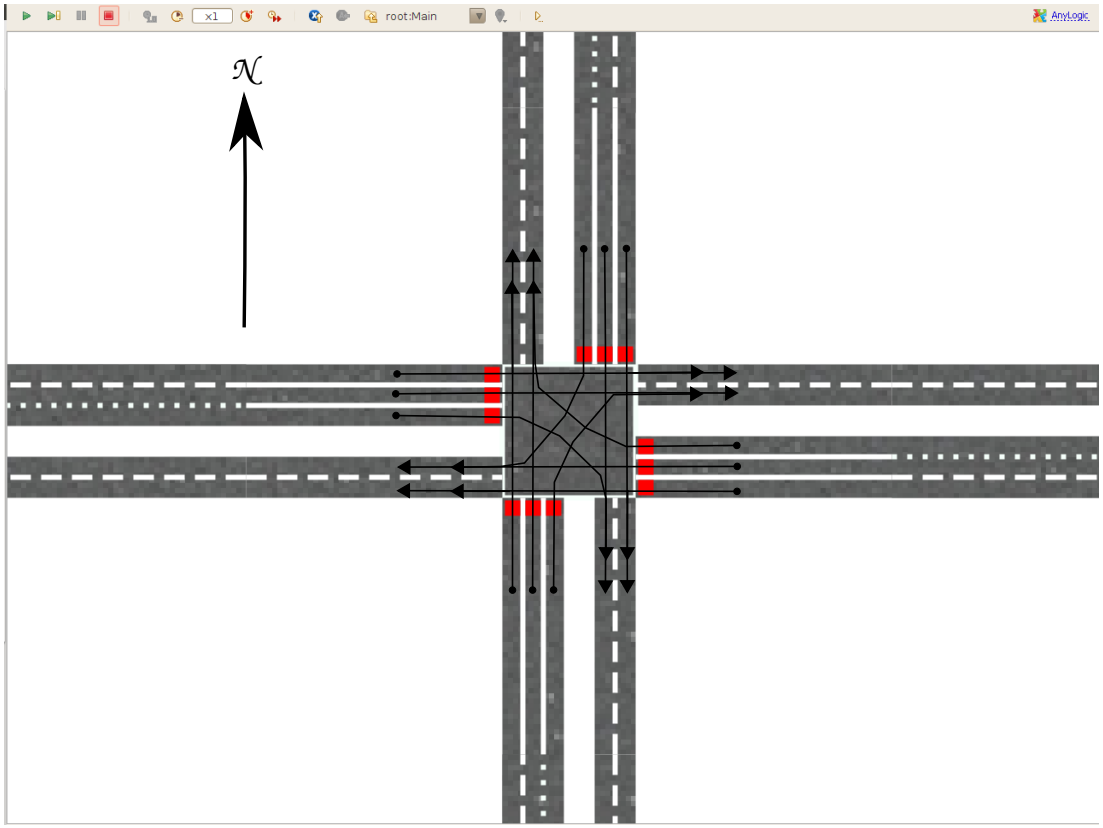


Figure 5.1: Example of a homogeneous intersection.

5.2 Simulation experimental design

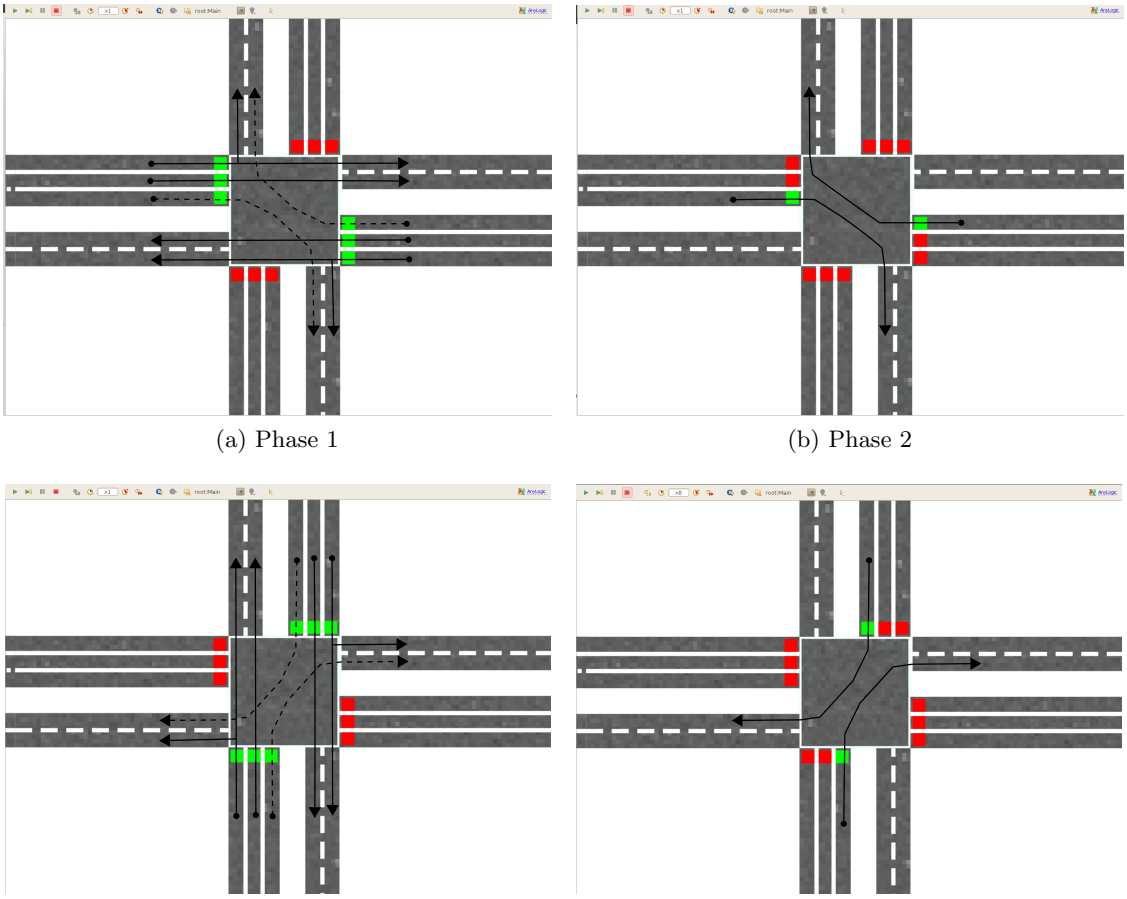
This section contains a description of the characteristics of the intersections and the traffic signals which control flow through them, as well as the parameter values imposed for each agent (*i.e.* each vehicle and each set of traffic lights) in the simulation model experiments. The methods adopted to determine adequate warm-up periods are also described.

5.2.1 Intersection design

Each of the topologies investigated in the simulation model comprises a number of homogeneous intersections. An example of one of these homogeneous intersections and its associated vehicle movements is illustrated in Figure 5.1. Each approach to the intersection comprises three lanes. Vehicles in the left-most lane may turn left or they may travel straight through the intersection, depending on the lane's position in the network. Along the centre lane, vehicles may only travel straight through the intersection. Vehicles along the right-most lane have all turned off the centre lane in preparation to turn right. Vehicles along the right-most lane may therefore only turn right.

5.2.2 Traffic signal phasing

The number of phases comprising a traffic control signal cycle depended on the control algorithm implemented at the traffic control signal (*i.e.* fixed or self-organising), but resembles the four-



(a) Phase 1 (b) Phase 2

(c) Phase 3 (d) Phase 4

Figure 5.2: The four different phases of one complete cycle and their associated vehicle movements.

phase plan with two exclusive lagging right-turn phases, depicted in Figure 2.2. In the case of fixed time control, the number of phases comprising one complete cycle is 4. These phases are illustrated sequentially in Figure 5.2.

The first phase of the traffic signal control cycle may be seen in Figure 5.2(a). During this phase all vehicles travelling from West to East and from East to West as well as all vehicles turning left, may travel freely through the intersection, while all vehicles intending to turn right are required to wait until there is a sufficient gap in oncoming traffic which permits the safe execution of a right turn. During Phase 2, illustrated in Figure 5.2(b), all vehicles intending to turn right receive an exclusive green signal, which permits them to travel freely through the intersection without having to consider oncoming traffic. In between Phase 1 and Phase 2, the traffic lights responsible for controlling traffic flow along the centre and left most lanes in both directions, which displayed a green signal during Phase 1, display an amber signal for a certain time period, before indicating red during Phase 2.

The vehicle movements associated with Phase 3, shown in Figure 5.2(c), are similar to those shown in Phase 1; however, they are concerned with all vehicles travelling from South to North and from North to South. In between Phase 2 and Phase 3, there is a period during which all the signals of the intersection indicate red, known as an all-red period which allows the

intersection to be cleared of any vehicles before the commencement of Phase 3. Phase 4, shown in Figure 5.2(d) allows for the protected movement of all vehicles wishing to turn right, which were travelling from South to North or from North to South. Like, Phase 2, Phase 4 is preceded by an amber period and followed by an all-red period.

In the case of self-organising traffic light control, the number of phases comprising one complete cycle, may be two, three or four phases. The reason for this is that with the radar technology associated with the self-organising control comes the ability to detect any vehicles which are waiting to turn right. If there are no vehicles waiting to turn right during Phase 1, Phase 2 to may skipped altogether, with the state of the lights proceeding to Phase 3, following an amber and all-red period. A similar situation arises if there are no vehicles waiting to turn right during Phase 3, in which case Phase 4 is skipped. The self-organising control regime is also able to ensure that vehicles do not receive a red signal when it is not absolutely necessary. An example of this may be seen in Figure 5.3, which shows an alternative version of Phase 2 than in Figure 5.2(b). At the time of the change from Phase 1 to Phase 2, there is a vehicle in the right-most lane of Figure 5.3, labelled *V1*, along the approach carrying vehicles travelling from West to East through the intersection. This vehicle intends to turn right and is thus afforded a green signal, while the oncoming traffic in the left-most and centre lanes of the approach carrying vehicles moving from East to West, are shown a red signal. However, there are no vehicles intending to turn right along the approach carrying vehicles from East to West, and therefore the oncoming traffic is still shown a green signal, which will change to amber and then red if a vehicle does, at some later stage, intend to turn right, travelling from East to West, provided enough time remains in the phase for an amber and red period, and for the vehicle to clear the intersection.

5.2.3 Simulation model initialisation and parameter values

There are three parameters associated with each road section in the simulation model. These are: the speed limit imposed upon all vehicles travelling along the road section, the jam density of the road section and the safe time gap maintained between all consecutive vehicles along the road section (see §2.3.1). Also associated with certain road sections are turning probabilities, which are used to determine if vehicles turn left or right off the road section on which they are currently travelling, as described in §4.2.

The speed limit imposed on each road section is set to 14 metres per second, which equates to approximately 50 kilometres per hour. The jam density of each road section is set to 0.2 vehicles per metre of road, which translates to each vehicle having an effective length of 5 metres (see §2.3.1). The safe time gap imposed upon following vehicles was set at 2 seconds for every 1 metre per second of speed at which the vehicle travels. The saturation flow rate along each road section was set at 0.3 vehicles per second. For the road sections from which vehicles may turn, whether it be left or right, the turning probabilities were set at 0.2, *i.e.* if a vehicle can turn left or right off from its current road section, there is a 20% chance that it will do so.

The parameter settings described above remain fixed for each simulation run regardless of the control strategy being implemented, and their effects on the system, were therefore not investigated. There are, however, parameters which were altered from from one simulation run to another in order to investigate their effects on the system as a whole. The parameters which were varied depend on the traffic control regime implemented.

In the case of fixed traffic signal control, the green times associated with Phase 1 and Phase 3, (as shown in Figures 5.2(a) and 5.2(c), respectively) were set equal to each other, but were varied

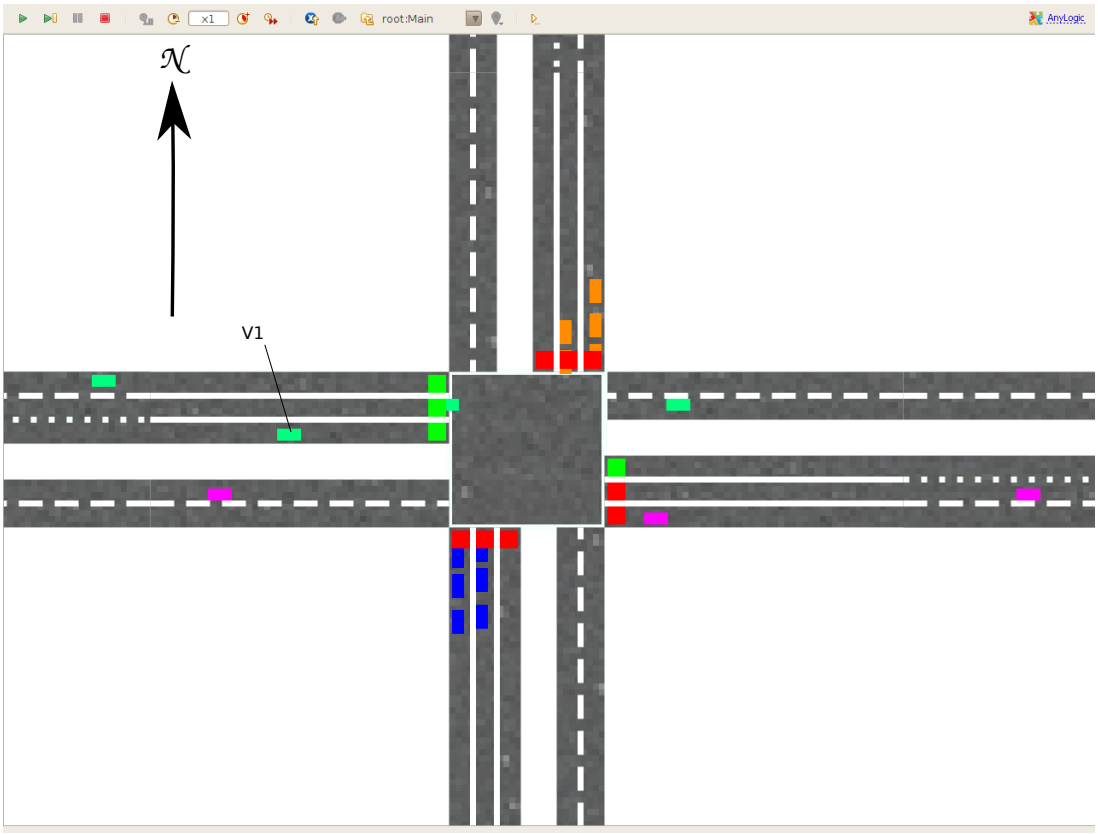


Figure 5.3: Example of a dynamic exclusive right-turn phase.

between a range of values for each of the different arrival rates, in order to find the most effective green time for each each performance measure, for each arrival rate. The inter-arrival times of vehicles at all the entry points of the simulation model were once again modelled according to a displaced exponential distribution, (see (4.8)), employing the arrival rate parameter λ. The effect of this value of λ on the system for the various topologies was investigated for the values of 0.05, 0.1, 0.15, 0.2, and 0.25. For each of the above values of λ, the effects of altering the green times associated with Phase 1 and Phase 3 of the fixed-time cycle-based control regime were investigated for the values of 5, 10, 20, 30, 40, 50 , 60, 70, 80, 90, and 100 seconds. The green times associated with Phase 2 and Phase 4 of the fixed control regime were fixed at 10 seconds for each simulation run.

In the case of the self-organising traffic control regimes OPS I, of Algorithm 5.4, OPS II, of Algorithm 5.7, as well as the SS, of Algorithm 5.5 were tested for each value of λ. In the case of SOTCA I, of Algorithm 5.6, which combines the optimising prioritisation strategy of Algorithm 5.3 with the stabilisation strategy of Algorithm 5.5, SOTCA II, of Algorithm 5.8, which combines the optimising prioritisation strategy of Algorithm 5.7 with the stabilisation strategy of Algorithm 5.5, the various combinations of U and $U^{\max}$ (see §2.4.2) were considered for each value of λ. The values of U considered were 40, 60, 80, 100, 120, 140 and 160 seconds, and the values of $U^{\max}$ considered were 60, 80, 100, 120, 140, 160 and 180 seconds. The values of U and $U^{\max}$ were combined such that $U^{\max} > U$. If an exclusive right-turn phase was required, then the length of the right-turn phase was determined by the right turning lane with the greatest number of vehicles requiring service upon it. That is, if the larger number of vehicles intending to turn right were found along road section i, then the green time allocated to the

exclusive right-turn phase would be set to the minimum of the number of vehicles along road section i divided by the saturation flow rate of the road section, $Q_i^{\max}$, and some maximum allowable green time, which was implemented at 20 seconds for each simulation run.

Depending on the arrival rate of vehicles into the system for each simulation run, there was a specific warm-up period implemented before any observations with respect to the model performance measures were made. The determination of these warm-up periods is described below.

5.2.4 Simulation model warm-up times

When investigating certain simulation model performance measures over an extended period of time, it may often be seen that the initial few observations do not provide a true representation of the steady-state behaviour of the model. To account for this lack of representation, a warm-up period is usually introduced to the simulation model during which all observations made are ignored or discarded. However, a natural question arising from the introduction of a warm-up period, is how long the warm-up period should be.

When attempting to estimate the steady-state mean $v = E(Y)$, of a system, which is also generally defined by

$$v = \lim_{i \to \infty} E(Y_i)$$

associated with the observations $Y_1, \ldots, Y_m$ of a variable Y in a simulation model, it may be seen that the transient means converge to the steady state mean. However, Law [27] describes the *problem of the initial transient* where $E[\bar{Y}(m)] \neq v$ for any m. To overcome this problem, he suggests to introduce a warm-up period of length l should be introduced during which all observations are ignored, and that

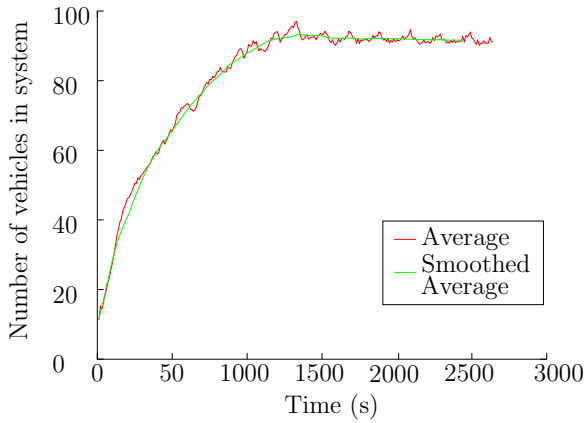
$$\bar{Y}(m, l) = \frac{\sum_{i=l+1}^{m} Y_i}{m - l}$$

should be used as an estimator of v rather than $\bar{Y}(m)$, for all $1 \leq l \leq m - 1$. To determine the size of l, Law suggests employing the following four steps:

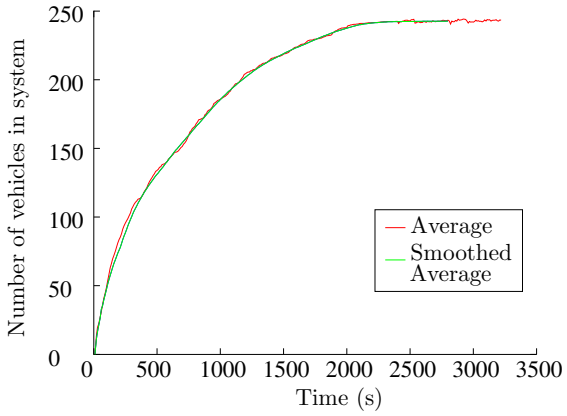
1. Produce n replications of the simulation, each of length m, letting Y_{ji} be the ith observation of the variable Y from the jth replication, for all $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. In the case of the simulation model implemented in this thesis, Y_{ji} was taken as the total number of vehicles present in the system at observation point i during simulation run j. Each simulation ran for the equivalent of 2400 seconds (or 40 minutes), unless no steady state was apparent after this time, in which case this time was extended to allow for the emergence of a steady state. Observations were made every 10 seconds, (*i.e.* $m = 240$). The number n of replications produced for each arrival rate was initially set to 10. However, this value could be increased if it did not yield a satisfactory indication of a steady state, as prescribed by Law [27].

2. Let $\bar{Y}_i = \sum_{j=1}^{n} Y_{ji}/n$ for $i = 1, 2, \ldots, m$. Thus the averaged processes $\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_m$ have means $E(\bar{Y}_i) = E(Y_i)$ and variances $\mathrm{Var}(\bar{Y}_i) = \mathrm{Var}(Y_i)/n$, *i.e.* the averaged process has the same transient mean curve as the original process, but its plot has only $(1/n)$th of the variance.

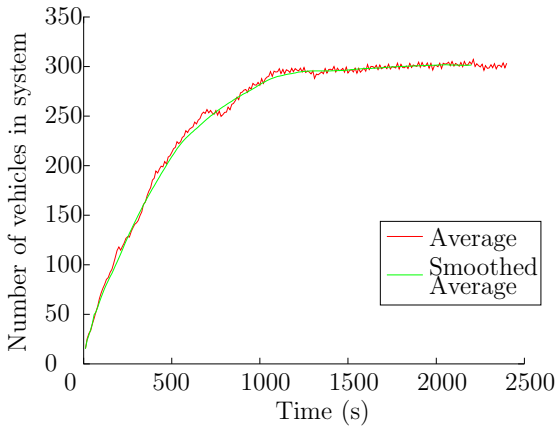
3. To smooth out the high frequency oscillations in $\bar{Y}_1, \bar{Y}_2, \ldots, \bar{Y}_m$ while at the same time leaving the low-frequency oscillations or trends which are of interest to the investigation,



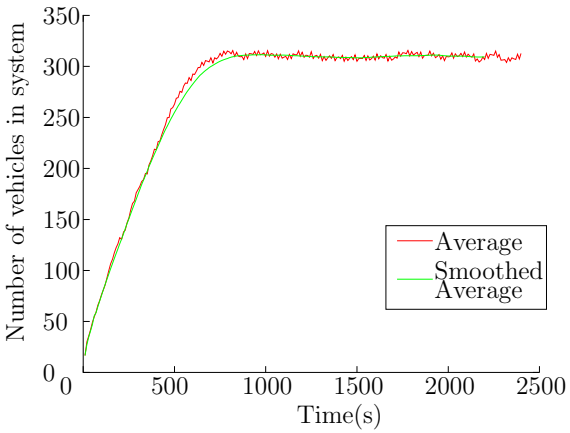
(a) $\lambda = 0.05$



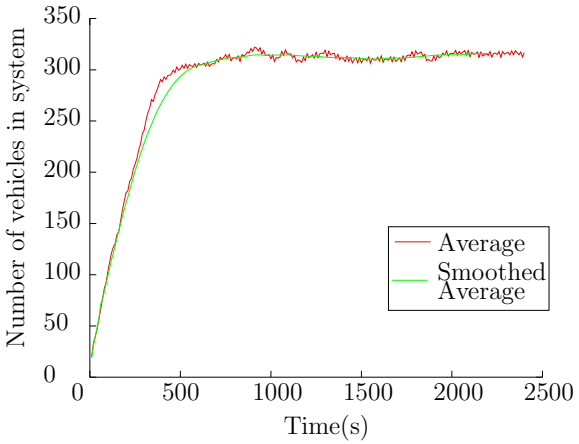
(b) $\lambda = 0.1$



(c) $\lambda = 0.15$



(d) $\lambda = 0.2$



(e) $\lambda = 0.25$

Figure 5.4: Determination of warm-up periods for vehicle arrival rates of (a) $\lambda = 0.05$, (b) $\lambda = 0.1$, (c) $\lambda = 0.15$, (d) $\lambda = 0.2$ and (e) $\lambda = 0.25$.

a moving average

$$\bar{Y}_i(w) = \begin{cases} \frac{\sum_{s=-w}^{w} \bar{Y}_{i+s}}{2w+1} & \text{if } i = w+1, \ldots, m-w \\ \frac{\sum_{s=-(i-1)}^{i-1} \bar{Y}_{i+s}}{2i-1} & \text{if } i = 1, \ldots, w, \end{cases}$$

where w is the window of the moving average and is a positive integer such that $w \leq \lfloor m/4 \rfloor$.

4. Plot $\bar{Y}_i(w)$ for $1 = 1, 2, \ldots, m-w$ and choose l to be that value of i beyond which $\bar{Y}_1(w), \bar{Y}_2(w), \ldots$ appears to have converged.

The plots achieved for each arrival rate parameter λ, following the steps outlined above, may be seen in Figure 5.4. The investigation with respect to the determination of appropriate warm-up periods was performed upon a single intersection with a topology and allowable vehicle movements resembling that of Figure 5.1 and a signal phasing cycle depicted in Figure 5.2, with a green time of 20 seconds associated with Phases 1 and 3 and an explicit right turn green time of 10 seconds associated with Phases 2 and 4.

For each value of λ, the system was assumed to have reached its steady-state when the number of vehicles in the system remained relatively constant. This phenomenon may be interpreted from the plots in Figure 5.4 as the point in time when the curve representing the number of vehicles present in the system plateaus. For systems having arrival rate parameters of $\lambda = 0.05, \lambda = 0.1, \lambda = 0.15, \lambda = 0.2$ and $\lambda = 0.25$, the associated warm-up periods were estimated to be 1800 seconds, 1800 seconds, 1500 seconds, 1000 seconds, and 800 seconds, respectively. It may be noted that the warm-up time required decreases as the frequency of vehicles arriving into the system increases. The simulation model was run for the equivalent of an additional ten and half hours once the warm-up period had elapsed in order to obtain a true reflection of the performance of the various traffic control algorithms for the different arrival patterns of vehicles into the system for an extended period of time.

5.3 Results

The results presented in this section were obtained for four different traffic network topologies. The first topology is a single, isolated intersection, the second is a two-by-two grid of intersections, the third is a three-by-three grid of intersections, and the fourth is a real case study investigating how the aforementioned self-organising control algorithms compare to currently implemented control techniques at the Adam Tas Road and Bird Street intersection in Stellenbosch, South Africa.

The performance measures considered in each simulation include the mean waiting time of vehicles in the system, the maximum waiting time of vehicles in the system, the mean time spent in the system by vehicles, the maximum time spent by vehicles in the system, and the sum of the mean queue lengths along all the road sections approaching intersections in the system.

Two sets of results are presented for each topology. In the first set of results, the combination of parameters associated with each traffic control algorithm described in §5.1 are varied such that an optimal value for each performance measure may be found for every value of the arrival rate parameter λ. These optimal values are then plotted against one another in order to provide a visual comparison of the performances of each of the algorithms tested.

OFTTCA					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	26.75	34.51	45.07	69.08	148.76
Green time (s)	20.00	40.00	60.00	90.00	90.00
Max. waiting time (s)	94.00	174.00	354.00	541.00	1577.00
Green time (s)	30.00	50.00	50.00	90.00	90.00
Total mean queue length	9.19	22.95	43.11	87.14	206.24
Green time (s)	20.00	40.00	60.00	90.00	90.00
Mean time in system (s)	59.69	67.96	79.60	106.73	188.79
Green time (s)	20.00	40.00	60.00	90.00	90.00
Max. time in system (s)	119.23	196.19	374.07	557.20	1591.16
Green time (s)	30.00	50.00	50.00	90.00	90.00

Table 5.1: Simulation results obtained by OFTTCA for a single, isolated intersection with corresponding green time values.

In the second set of results, the various parameter combinations are fixed for each algorithm, while the value of λ increases over time. More specifically, the simulation is initiated with $\lambda = 0.05$. This value of λ is assumed throughout the simulation's warm-up period of 1 800 seconds (the determination of which was motivated in §5.2.4) together with an additional 7 200 seconds, or two hours. Once this time has elapsed, the value of λ increases by 0.05 to 0.1, and remains fixed for the following two hours. This process continues until the simulation has run for a total of ten and a half hours. Thus, the simulation runs for a total of ten and a half hours, which includes a 30 minute warm-up period, after which the value of λ increases by 0.05 every two hours, while, the combination of algorithm parameters remains fixed throughout. The values of λ considered consecutively are therefore $\lambda = 0.05$, 0.1, 0,15, 0.2 and 0.25.

5.3.1 A single intersection

The first set of results obtained for a single intersection, as depicted in Figure 5.5, are presented in Figure 5.6. For each algorithm tested, the results obtained for the mean and maximum waiting times of vehicles in the system are shown in Figures 5.6(a) and 5.6(b), respectively. The mean and maximum times spent by vehicles in the system are shown in Figures 5.6(c) and 5.6(d), respectively, while the sum of the average queue lengths along all the lanes approaching the intersection is shown for each algorithm in Figure 5.6(e). The results used to plot the graphs in Figure 5.6 are also presented in Tables 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6. Similar data tables associated with the rest of the graphs in this chapter may be found in Appendix B. The results obtained via the implementation of OFTTCA are presented in Table 5.1 where the value of each performance measure is given along with the associated green time implemented to achieve it. The results obtained via the implementation of SOTCA I, SOTCA II and the SS are presented in Tables 5.2, 5.3 and 5.6, respectively. In these tables performance measure values are presented together with the corresponding values of U and $U^{\max}$ as well as the resulting average green times. The values of the performance measures obtained by OPS I and OPS II are shown in Tables 5.4 and 5.5, respectively together with the resulting average green times.

In each of the graphs in Figure 5.6, it may be seen that for $\lambda = 0.05$ and $\lambda = 0.1$, the values of the various performance measures obtained by SOTCA I and SOTCA II are very similar to

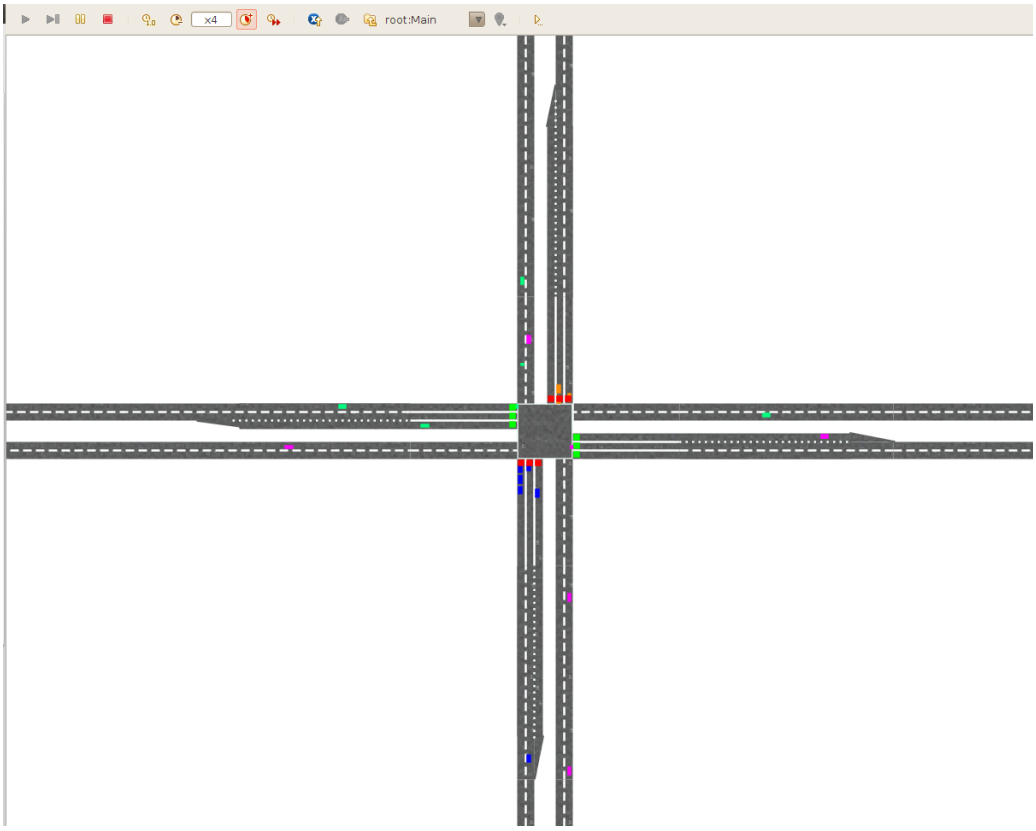


Figure 5.5: Screen shot of the simulation model in the case of a single, isolated intersection.

those achieved by OPS I and OPS II. It may also be seen that for $\lambda = 0.15$ to $\lambda = 0.25$, the performance measure values obtained by SOTCA I and SOTCA II are very similar to those of the SS. An explanation for these observations is that at the lower values of λ, which correspond to lighter traffic flows, the optimisation or prioritisation strategy of both SOTCA I and SOTCA II is predominantly responsible for the control of the switching of the traffic signals. As the value of λ increases, the number of vehicles arriving in the system per time unit increases, and so the stabilisation strategy of SOTCA I and SOTCA II begins to intervene more frequently in the control of the switching of the traffic signals, as the vehicle queues along the various approaches to the intersection tend to exceed their maximum allowable lengths, $\hat{n}^{\text{crit}}$, as determined by the values U and $U^{\max}$ in (2.28).

The performance measures associated with OPS I become considerably large for values of λ greater than or equal to 0.15. A reason for this observation is that allocating enough green time to clear the currently queued vehicles along an approach, as well as enough green time to serve the number of vehicles expected to arrive while the queue is being discharged, results in excessively long waiting times for the vehicles which are waiting to receive service. In the case of OPS II, the various performance measures may also be seen to grow excessively large for values of λ greater than 0.1. A reason for this is that as the vehicle queues grow in length, even when an approach is selected for service, a relatively large number of vehicles remain within close proximity to the intersection as they discharge from rest. This results in the value of the priority index of the approach receiving service, π_{σ}^{II}, to remain greater than that of competing approaches by a value exceeding ϵ for an extended period of time. This too results in longer waiting times for vehicles not currently receiving service.

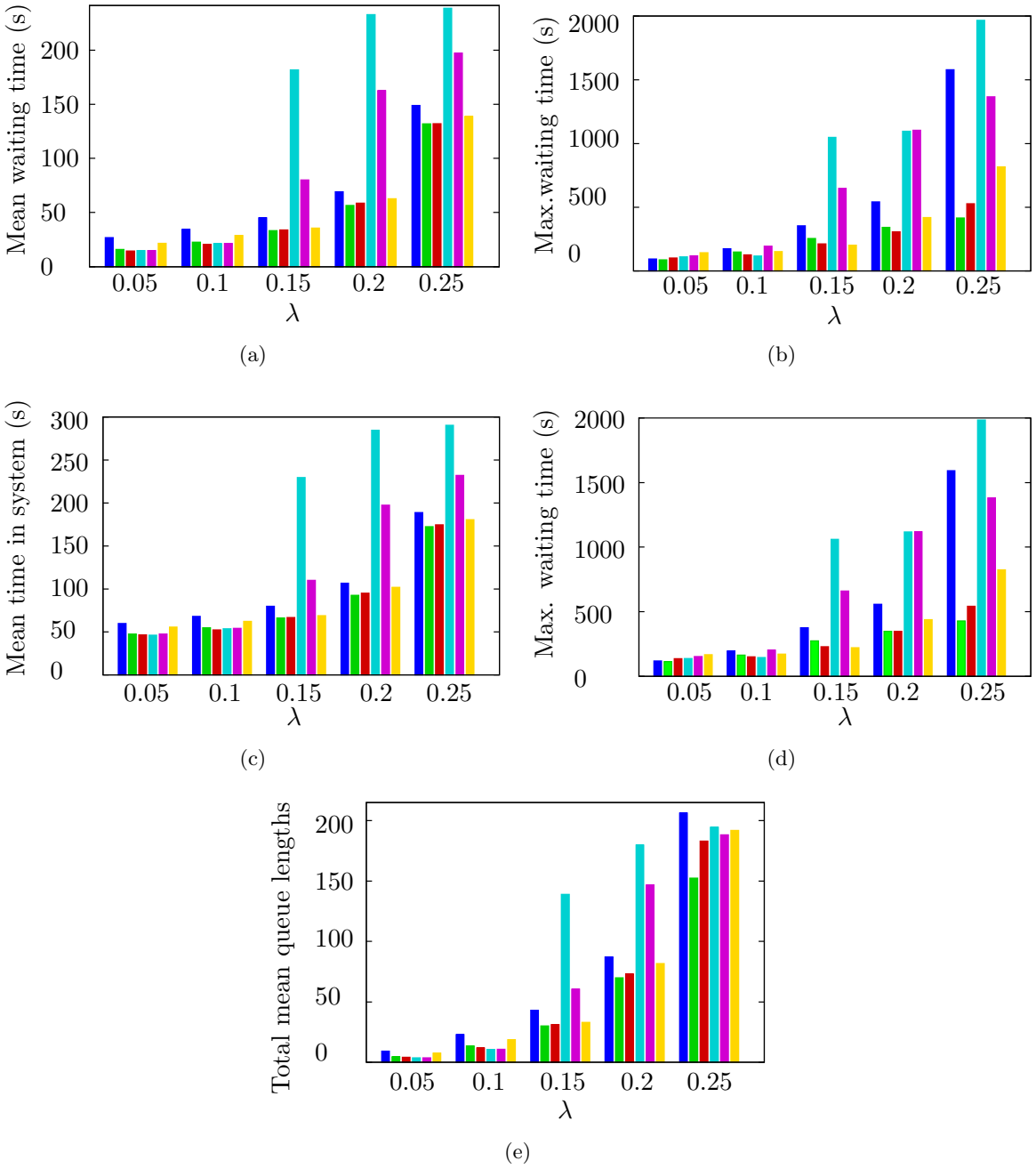


Figure 5.6: Simulation results obtained for a single, isolated intersection topology in which the algorithm parameters are adjusted for each value of λ separately such that an optimal performance measure value may be found.

SOTCA I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	15.77	22.41	33.25	56.54	131.91
U	40.00	60.00	60.00	100.00	140.00
$U^{\max}$	100.00	80.00	80.00	140.00	180.00
Mean green time (s)	7.67	21.33	43.04	77.95	126.12
Max. waiting time (s)	86.00	149.00	254.00	341.00	415.00
U	40.00	60.00	60.00	80.00	60.00
$U^{\max}$	100.00	80.00	80.00	120.00	160.00
Mean green time (s)	7.67	21.33	43.04	64.09	52.43
Total mean queue lengths	4.62	13.47	30.08	69.92	152.36
U	40.00	60.00	60.00	100.00	140.00
$U^{\max}$	100.00	100.00	80.00	140.00	180.00
Mean green time (s)	7.67	20.92	43.04	77.95	126.12
Mean time in system (s)	47.32	54.52	66.12	92.38	172.40
U	40.00	60.00	60.00	100.00	140.00
$U^{\max}$	100.00	80.00	80.00	140.00	180.00
Mean green time (s)	7.67	21.33	43.04	77.95	126.12
Max. time in system (s)	113.22	163.03	273.19	347.03	427.22
U	40.00	60.00	60.00	80.00	60.00
$U^{\max}$	100.00	80.00	80.00	120.00	160.00
Mean green time (s)	7.67	21.33	43.04	64.09	52.43

Table 5.2: Simulation results obtained by SOTCA I for a single, isolated intersection with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

Through close inspection of all the graphs in Figure 5.6, it may be observed that for the lower values of λ, (*i.e.* 0.05, 0.1 and 0.15) SOTCA II slightly outperforms SOTCA I, while the converse is true for λ-values of 0.2 and 0.25. A reason for this is that SOTCA I may tend to slightly over-estimate the anticipated number of vehicles expected to arrive for lower values of λ due to the fact that the arrivals of vehicles into the system are generated according to a random process, while for larger values of λ, SOTCA II may take slightly too long to switch service from one approach to another due to the size of ϵ.

OFTTCA is consistently outperformed by SOTCA I and SOTCA II due to the fact that it is simply not as flexible with respect to accommodating slight fluctuations in vehicle arrivals to the intersection as are SOTCA I and SOTCA II. For example, OFTTCA may at times provide service to an approach which currently contains no vehicles requiring service, while vehicles along other approaches are made to wait needlessly. The operation of the SS is similar to that of OFTTCA; however, it is slightly more flexible in that if an approach is selected for service, it will receive service for a predetermined length of time, *i.e.* $g^{\max}$ (see §2.4.2) or until the queue along the approach receiving service is fully discharged. This added flexibility of the SS over OFTTCA is the reason why it outperforms OFTCA with respect to the majority of the performance measures.

There is a clear correspondence between the observations in Figure 5.6(a) and those in Figure 5.6(c), as well as between those in Figure 5.6(b) and those in Figure 5.6(d). A simple explanation for this correspondence is that the amount of time a vehicle spends in the system correlates with

SOTCA II					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	14.47	20.53	33.82	58.49	132.01
U	40.00	40.00	60.00	100.00	100.00
$U^{\max}$	60.00	80.00	80.00	120.00	120.00
Mean green time (s)	14.64	16.79	28.33	77.94	89.74
Max. waiting time (s)	101.00	126.00	212.00	307.00	527.00
U	40.00	40.00	60.00	80.00	80.00
$U^{\max}$	180.00	100.00	100.00	160.00	140.00
Mean green time (s)	36.02	30.39	29.01	62.38	71.70
Total mean queue lengths	4.15	11.78	31.16	73.05	182.95
U	40.00	40.00	60.00	100.00	100.00
$U^{\max}$	60.00	80.00	80.00	160.00	120.00
Mean green time (s)	14.64	16.79	28.33	73.98	89.74
Mean time in system (s)	46.38	52.19	66.53	95.19	174.63
U	40.00	40.00	60.00	100.00	100.00
$U^{\max}$	60.00	80.00	80.00	160.00	120.00
Mean green time (s)	14.64	16.79	28.33	73.98	89.74
Max. time in system (s)	135.83	150.14	229.11	347.46	542.07
U	40.00	40.00	60.00	80.00	80.00
$U^{\max}$	180.00	100.00	100.00	160.00	140.00
Mean green time (s)	36.02	30.39	29.01	62.38	71.70

Table 5.3: Simulation results obtained by SOTCA II for a single, isolated intersection with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OPS I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	14.88	21.22	181.71	232.70	238.80
Mean green time (s)	7.89	26.19	471.13	588.92	650.11
Max. waiting time (s)	111.00	117.00	1047.00	1965.00	1096.00
Mean green time (s)	7.89	26.19	471.13	588.92	650.11
Total mean queue lengths	3.62	10.58	138.87	179.74	194.44
Mean green time (s)	7.89	26.19	471.13	588.92	650.11
Mean time in system (s)	46.30	53.42	229.64	284.52	290.56
Mean green time (s)	7.89	26.19	471.13	588.92	650.11
Max. time in system (s)	137.23	145.01	1060.12	1984.17	1117.16
Mean green time (s)	7.89	26.19	471.13	588.92	650.11

Table 5.4: Simulation results obtained by OPS I for a single, isolated intersection with the resulting mean green time values.

OPS II					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	14.76	21.27	79.90	162.48	197.40
Mean green time (s)	22.23	29.46	22.16	23.71	29.51
Max. waiting time (s)	118.00	194.00	647.00	1366.00	1103.00
Mean green time (s)	22.23	29.46	22.16	23.71	29.51
Total mean queue lengths	3.69	10.70	60.37	146.74	188.20
Mean green time (s)	22.23	29.46	22.16	23.71	29.51
Mean time in system (s)	47.43	53.93	110.08	197.54	231.98
Mean green time (s)	22.23	29.46	22.16	23.71	29.51
Max. time in system (s)	152.22	202.25	660.22	1380.86	1117.72
Mean green time (s)	22.23	29.46	22.16	23.71	29.51

Table 5.5: Simulation results obtained by OPS II for a single, isolated intersection with the resulting mean green time values.

SS					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	21.54	28.74	35.61	62.69	138.69
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.84	60.86	60.09	79.72	107.84
Max. waiting time (s)	143.00	152.00	201.00	419.00	816.00
U	40.00	60.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	100.00	120.00	160.00
Mean green time (s)	52.84	60.86	60.09	79.72	89.82
Total mean queue lengths	7.55	18.66	32.98	81.69	191.90
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.84	60.86	60.09	79.72	107.84
Mean time in system (s)	55.80	62.36	68.89	101.74	180.77
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.84	60.86	60.09	79.72	107.84
Max. time in system (s)	167.08	171.71	222.07	438.07	825.07
U	40.00	60.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	100.00	120.00	160.00
Mean green time (s)	52.84	60.86	60.09	79.72	89.82

Table 5.6: Simulation results obtained by the SS for a single, isolated intersection with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

the amount of time it spends waiting in a queue.

The results obtained with respect to the total mean queue lengths in Figure 5.6(e) follow the same general distribution as in the other graphs in Figure 5.6 for all values of λ, except for $\lambda = 0.25$. For the case of $\lambda = 0.25$, however, OPS I outperforms OFTTCA and OPS II outperforms OFTTCA as well as the SS. A reason for this may be that due to the excessively long periods of service afforded by OPS I and OPS II, there are typically only queues, albeit very long queues, along approaches not receiving service, while there are seldomly queues, or short queues, on the approaches receiving service. This stands in contrast to the other control algorithms which generally tend to result in the system exhibiting queues of fluctuating size on all approaches at all times for the case where $\lambda = 0.25$.

The second set of results obtained for the case in which the parameter combinations remain fixed while the value of λ increases over time are shown in Figure 5.7. The results shown in Figures 5.7(a), 5.7(b), 5.7(c), 5.7(d) and 5.7(e) were all obtained by the parameter combinations of each algorithm which minimised the respective performance measures over the entire ten and a half hour simulation run. The general pattern of the results in each of the graphs in Figure 5.7 may be seen to resemble those of their counterparts in Figure 5.6.

There is, however, one noticeable difference between Figures 5.7(a), 5.7(c) and 5.7(e) and their counterparts in Figure 5.6. This difference is that for the first six-and-a-half hours of the simulation, during which the value of λ increases from 0.05 to 0.15, SOTCA I, SOTCA II, OPS I, and OPS II outperform OFTTCA and the SS in terms of the mean waiting times of vehicles in the system and the mean time spent by vehicles in the system, as well the total mean queue lengths of the system by a greater margin than in Figure 5.6. An explanation for this observation is that in order to minimise the overall respective performance measures for the entire simulation run, both OFTTCA and the SS implement parameter combinations which are best suited to minimise each performance measure for higher traffic flows. A consequence of this is that at lower traffic volumes, vehicles in the system experience unnecessarily long waiting times due to green times which are longer than they ought to be. Because OPS I and OPS II (and also SOTCA I and SOTCA II for the case where traffic signal switching is controlled by the optimisation or prioritisation strategy of the algorithm) are free of any predetermined parameter values, they are able to effectively adjust their signal timing settings so as to minimise vehicle queue lengths at the intersection, thus minimising vehicle waiting times and the total time spent by vehicles in the system.

An inspection of each of the graphs in Figure 5.7 shows a clear dominance in performance by both SOTCA I and SOTCA II. This may be attributed to the ability of each of these algorithms to optimise traffic flow at lower traffic volumes while being able to stabilise traffic flow at higher traffic volumes.

These observations bring to the fore the advantages offered by the flexibility of self-organising traffic signal control over fixed signal control when seeking to minimise vehicle waiting times and easing traffic congestion in a very simple traffic network topology.

5.3.2 A two-by-two grid of intersections

The results obtained for a two-by-two grid of intersections, as depicted in Figure 5.8, for the case in which the parameters of each algorithm are adjusted for each value of λ separately such that an optimal value of each performance measure may be found are shown in Figure 5.9. The same performance measures are considered as in the case of the single intersection in §5.3.1.

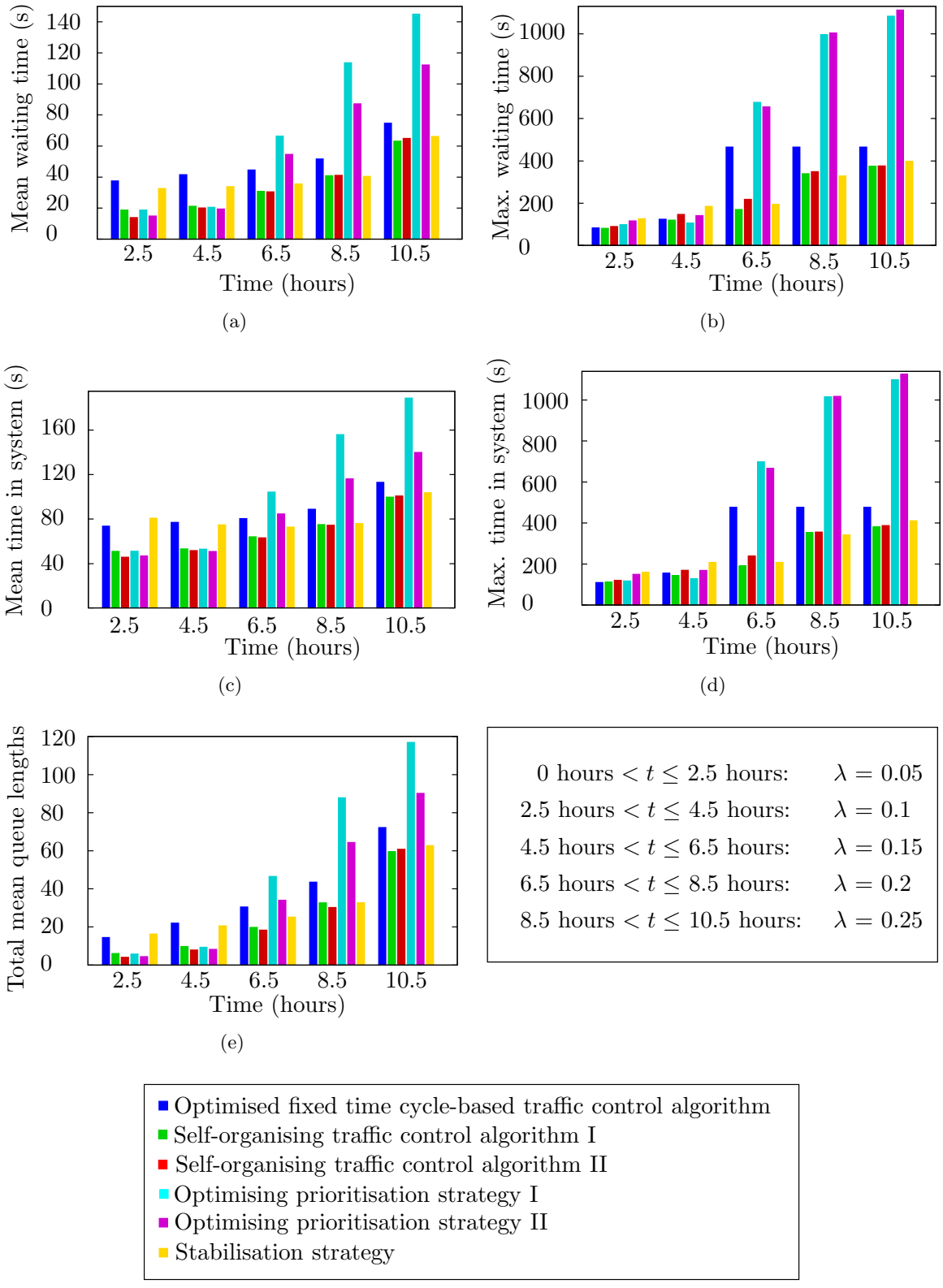


Figure 5.7: Simulation results obtained for fixed parameter combinations and varying values of λ for a single, isolated intersection.

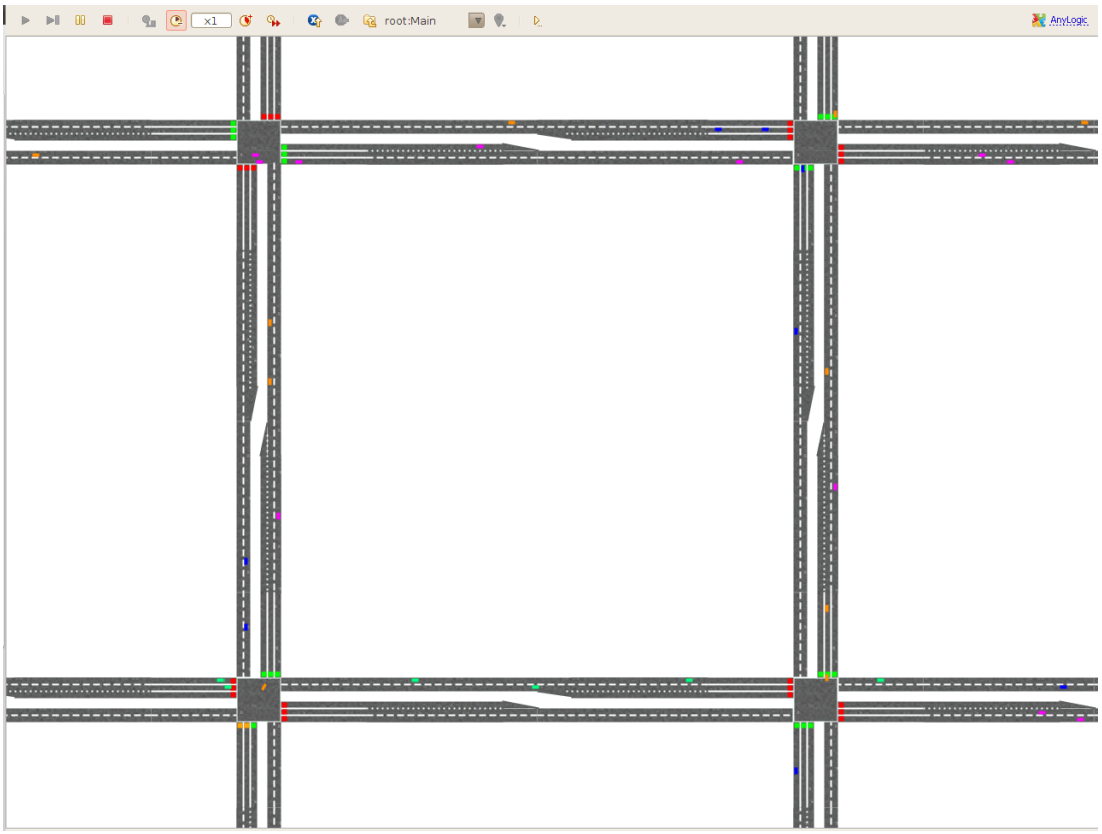


Figure 5.8: Screen shot of the simulation model in the case of a two-by-two grid of intersections.

The distribution of the results shown in Figure 5.9 may be seen to resemble those of Figure 5.6 closely save for the fact that they obtain relatively higher values. From Figures 5.9(a) and 5.9(c) it may be seen that the mean waiting times of the vehicles present in the system and the mean time spent by the vehicles in the system, respectively, are approximately twice that shown by their counterparts in Figure 5.6. This is to be expected, as the majority of vehicles passing through the system will encounter only two intersections as a results of the probabilities imposed with respect to vehicles turning left, or right at an intersection (see §5.2.3). The sum of the mean queue lengths along each intersection approach in the system, shown in Figure 5.9(e) may be seen to be approximately four times that of those in Figure 5.6(e). This corresponds with the fact that there are four times as many intersections in the two-by-two grid network as in the single intersection case, and therefore, four times as many intersection approaches.

The superior performance of SOTCA II over SOTCA I at lower values of λ is accentuated in the two-by-two grid of intersections, with even OPS II showing a superior performance over SOTCA I across all performance measures. This may again be attributed to the fact that for lower traffic flow volumes, the anticipation technique of SOTCA I tends to over-estimate the actual number of vehicles arriving, and as a result, reserves excess green time for the service of each intersection approach. Another reason is that SOTCA II and OPS II are able to achieve greater coordination among the traffic signals of the adjacent intersections. This is because, irrespective of the number of vehicles departing from one intersection, the same number of vehicles are received and detected by another, without any reliance on arrival estimates.

This idea of a greater sense of coordination is further affirmed by the performance of OPS II

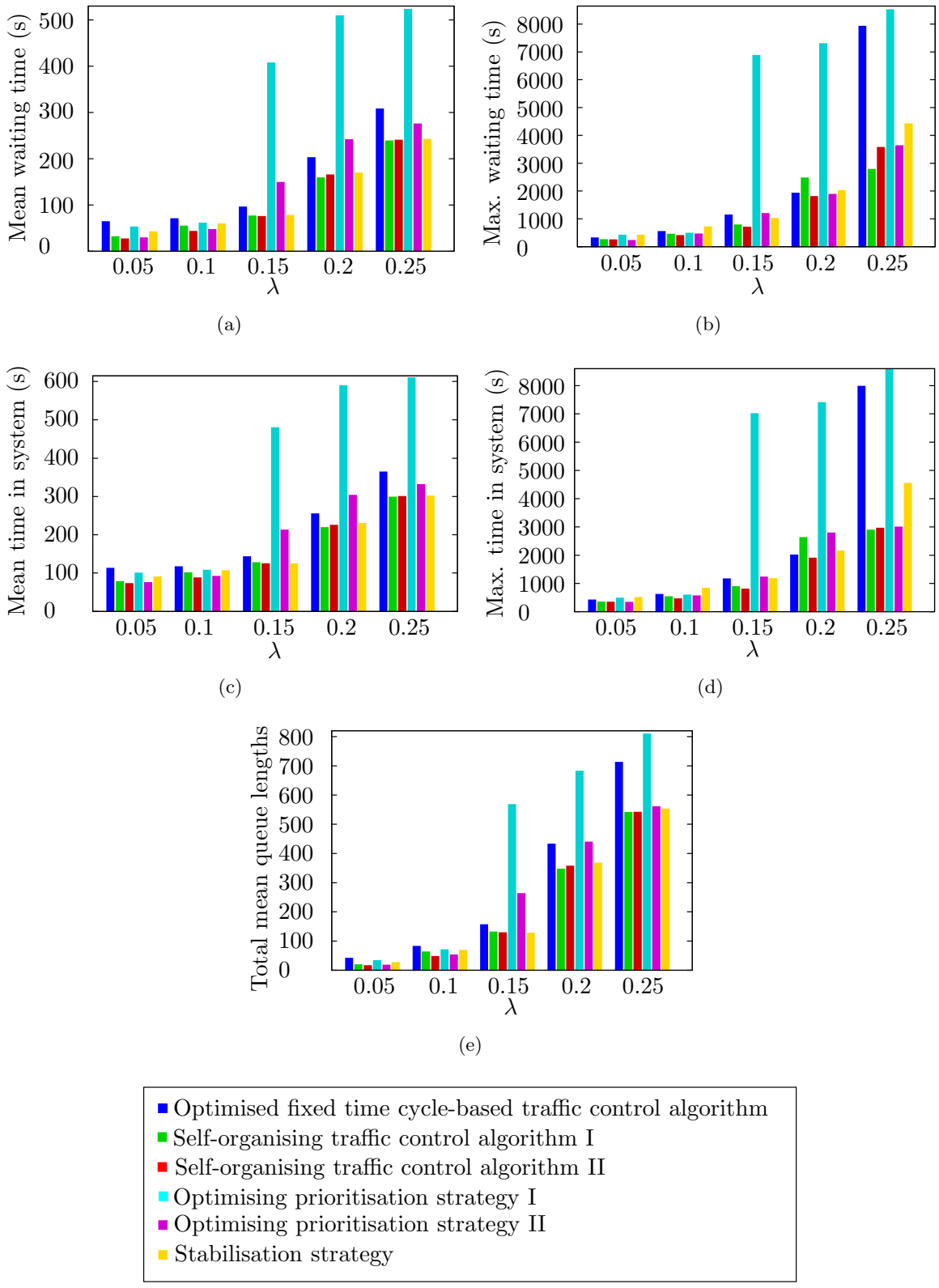


Figure 5.9: Simulation results obtained for a two-by-two grid of intersections in which the algorithm parameters are adjusted for each value of λ separately such that an optimal performance measure value may be found.

in terms of the mean and maximum waiting times of the vehicles in the system, as well the mean and maximum times they spend in the system for values of λ above 0.15. It may be seen that, although OPS II is outperformed by SOTCA I, SOTCA II and the SS, the performance measure values it achieves remain relatively constant for $\lambda = 0.2$ and $\lambda = 0.25$. This shows that through attempts to optimise traffic flow locally at each intersection, the system has reached a stable state in which each traffic flow is served in a timely and constant manner, which results in the absence of excessively long waiting times.

It may appear counter-intuitive that SOTCA I outperforms SOTCA II for the larger values of λ in spite of the fact that OPS II outperforms OPS I. However, this may be attributed to the fact that the stabilisation strategy of both SOTCA I and SOTCA II makes use of the green time anticipated by the prioritisation strategy of SOTCA I. The result of this anticipation is that even when SOTCA II is implemented, the stabilisation strategy intervenes under the premise that the optimisation strategy of SOTCA I is being implemented. This is because the green time anticipated by the prioritisation strategy associated with SOTCA I may not equal the green time implemented by the prioritisation strategy of SOTCA II, which may result in the situation where the stabilisation strategy will be implemented by SOTCA II when it is not entirely necessary. In the same sense, the stabilisation strategy may not intervene early enough to take over control from the optimisation strategy of SOTCA II in some cases where vehicle queues grow excessively large.

The results obtained for the two-by-two grid of intersections for the case in which the algorithm parameters remain fixed while the value of λ varies over time are shown in Figure 5.10. The most noticeable difference between the graphs in Figure 5.10 and those in Figure 5.7 is the increase in the improvement in the performance of OPS II when compared to that of OPS I. As was mentioned earlier, this may attributed to the greater level of coordination among intersections OPS II is able to achieve through the implementation of a prioritisation strategy which is not reliant on the anticipation of vehicle arrivals, but rather on the distance between the intersection and the vehicles approaching it. To quantify these results, it was found that the percentage improvement of OPS II over OPS I in terms of minimising the mean waiting times of vehicles over the entire ten and a half hours increased from 22.58% for the single intersection considered in §5.3.1, to 47.20% for the two-by-two grid of intersections. Similarly, in terms of the mean time spent by vehicles in the system the difference in performance superiority increased from 25.94% to 42.10%, while for the total mean queue sizes, it increased from 22.93% to 37.58%.

A similar analysis may be performed when comparing the performances of SOTCA I and SOTCA II to that of OFTTCA in an attempt to gauge the increase in performance superiority (if any) between self-organising and fixed-time traffic control paradigms when implemented in a larger traffic network with more signalised intersections. In terms of minimising the mean waiting times experienced by vehicles in the system, the margin of improvement exhibited by SOTCA I over OFTTCA increased from 15.4% to 23.96%, while for SOTCA II over OFTTCA, the same value increased from 13.11% to 26.13%. In terms of minimising the mean time spent by vehicles in the system, the differences in performance between SOTCA I and OFTTCA, and between SOTCA II and OFTTCA increased from 11.88% to 17.36% and from 10.94% to 19.53%, respectively. Finally, when considering the total mean queue lengths which were achieved as a result of the implementation of each of the algorithms OFTTCA, SOTCA I and SOTCA II, it was found that the value by which SOTCA I outperformed OFTTCA increased from 17.25% when implemented for a single intersection to 22.58% when implemented for a two-by-two grid of intersections, while for SOTCA II over OFTTCA, this value increased from 15.82% to 25.35%.

In addition to indicating the advantages offered by the flexibility of self-organising traffic signal

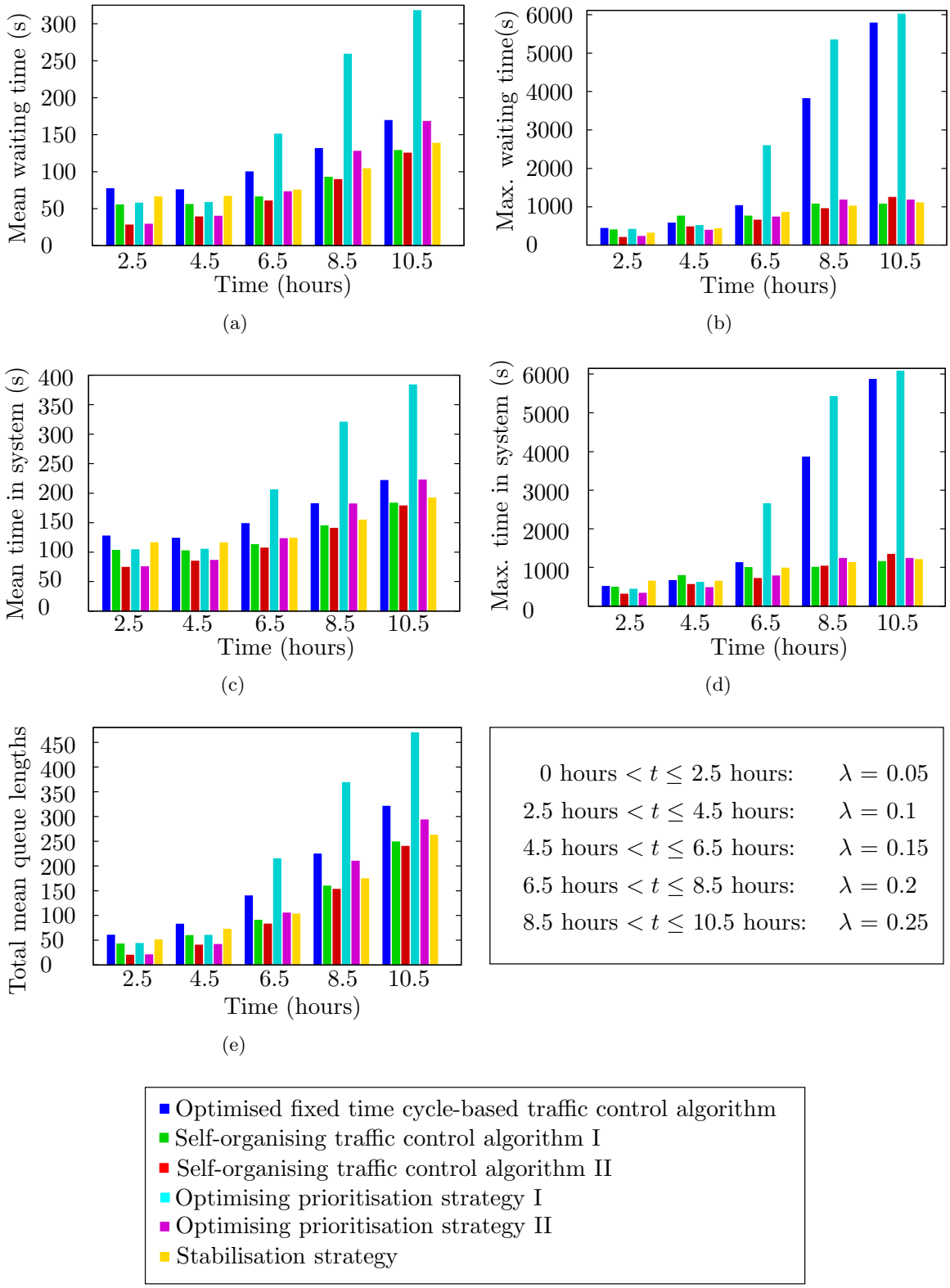


Figure 5.10: Simulation results obtained for fixed parameter combinations and varying values of λ for a two-by-two grid of intersections.

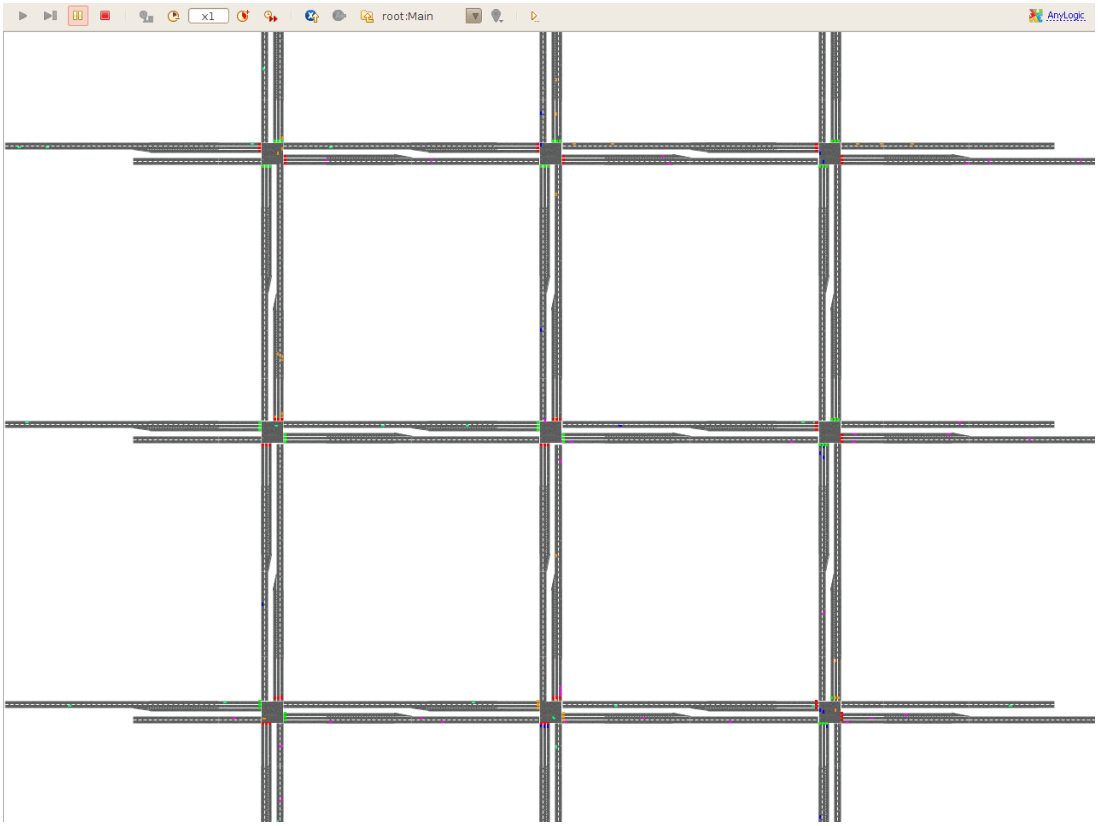


Figure 5.11: Screen shot of the simulation model in the case of a three-by-three grid of intersections.

control in its ability to adapt to fluctuating traffic flows, the results presented in this section serve to highlight the further advantages offered through the natural coordination which occurs among traffic signals at adjacent intersections.

5.3.3 A three-by-three grid of intersections

In the case of a three-by-three grid of intersections, as depicted in Figure 5.11, the results are once again presented in the same format as in the case of the single intersection and the two-by-two grid of intersections in §5.3.1 and §5.3.2, respectively. Figure 5.12 shows the results obtained when the parameter combinations of each traffic control algorithm are adjusted for each value of λ separately such that a corresponding optimal performance measure may be found.

As is expected when comparing the results in the graphs in Figure 5.12 with their counterparts in Figure 5.9, it may be seen that the magnitude of the increase in the performance measure observations has increased proportionally with the increase in the number of intersections. One observation worth noting in Figure 5.12 is the improvement in the performance of OPS II when compared to SOTCA I, SOTCA II and the SS for all values of λ. The reason for this improvement was touched upon in §5.3.2. It was attributed to the fact that through attempts to optimise traffic flow locally at each intersection, a global pattern of coordination emerged among the traffic signal controllers. The benefits of this system coordination are amplified by the introduction of additional intersections to the system, resulting in an approach towards a stable state as is indicated by the small increases in terms of mean waiting times of vehicles and

the mean time spent in the system by vehicles between λ values of 0.2 and 0.25 in Figures 5.12(a) and 5.12(c), respectively. The manifestation of the benefits of this greater sense of coordination is most clearly seen in Figure 5.12(e) in the case where $\lambda = 0.25$. Here, OPS II is able to achieve a 7.81% improvement over the SS in terms of minimising the total mean queue lengths along all intersection approaches in the system. A consequence of this improvement is that for $\lambda = 0.25$, it may be assumed that the optimising or prioritisation strategy of SOTCA II is largely responsible for traffic flow control, which stands in contrast to SOTCA I, which depends almost solely on its stabilisation strategy to control traffic for the same value of λ.

In the case where OPS I is implemented for λ values above 0.15, it was observed that saturation occurred along many of the intersection approaches. This resulted in a waste of intersection capacity due to the fact that in some instances a queue of vehicles would be shown a green signal while the vehicles would not be able to cross the intersection because there was no room to accommodate them on the adjoining road section. This waste of intersection capacity which sees no vehicles being served due to saturation is thought to contribute towards the considerably large performance measure values associated with the implementation of OPS I in the case where λ is larger than 0.15.

In the case where the algorithm parameters remained fixed while the value of λ increased for the three-by-three grid of intersections, the results obtained are shown in Figure 5.13.

When comparing the graphs of Figure 5.13 with their counterparts in Figure 5.10, the most noticeable difference is a further deterioration in the performance of OPS I for larger values of λ. As was mentioned earlier in this section, this may be attributed to the fact that saturation occurs along the intersection approaches resulting in a gridlock, in which traffic flow comes to a stand still as there is no space available for vehicles to cross intersections onto adjoining road sections.

It may be seen that SOTCA I and SOTCA II still clearly outperform OFTTCA for a larger traffic network. However, the differences in these performances are more or less equal to what they were in the comparison of the single intersection with those of the two-by-two grid of intersections. A possible reason for this improvement dissipation is that in the case of the implementation of SOTCA I or SOTCA II, vehicles were seldom observed to pass through three consecutive intersections in a green-wave manner. This is clearly seen from the reduced queue sizes present along the approaches of the intersection at the centre of the three-by-three grid. An explanation for this phenomenon is that the inter-arrival times of vehicles at the the centre-most intersection are relatively more uniform along all of the approaches of the intersection as opposed to the inter-arrival times of vehicles at the intersections on the boundaries of the grid, as they are released in platoons from the surrounding adjacent intersections as opposed to being generated by a random process. Nonetheless, this aspect of failing to achieve extended green waves is seen as a pitfall of the self-organising traffic control algorithms and is identified as an area for future studies.

5.3.4 A case study of the Adam Tas & Bird Street intersection

A real case study was performed for the same intersection which was used to validate the simulation model in §4.5, that is the Adam Tas Road & Bird Street intersection in Stellenbosch, South Africa. An aerial image of the intersection (taken from Google Maps [17]) may be seen in Figure 5.14.

The intersection is known to be one of the busier intersections in Stellenbosch. The majority

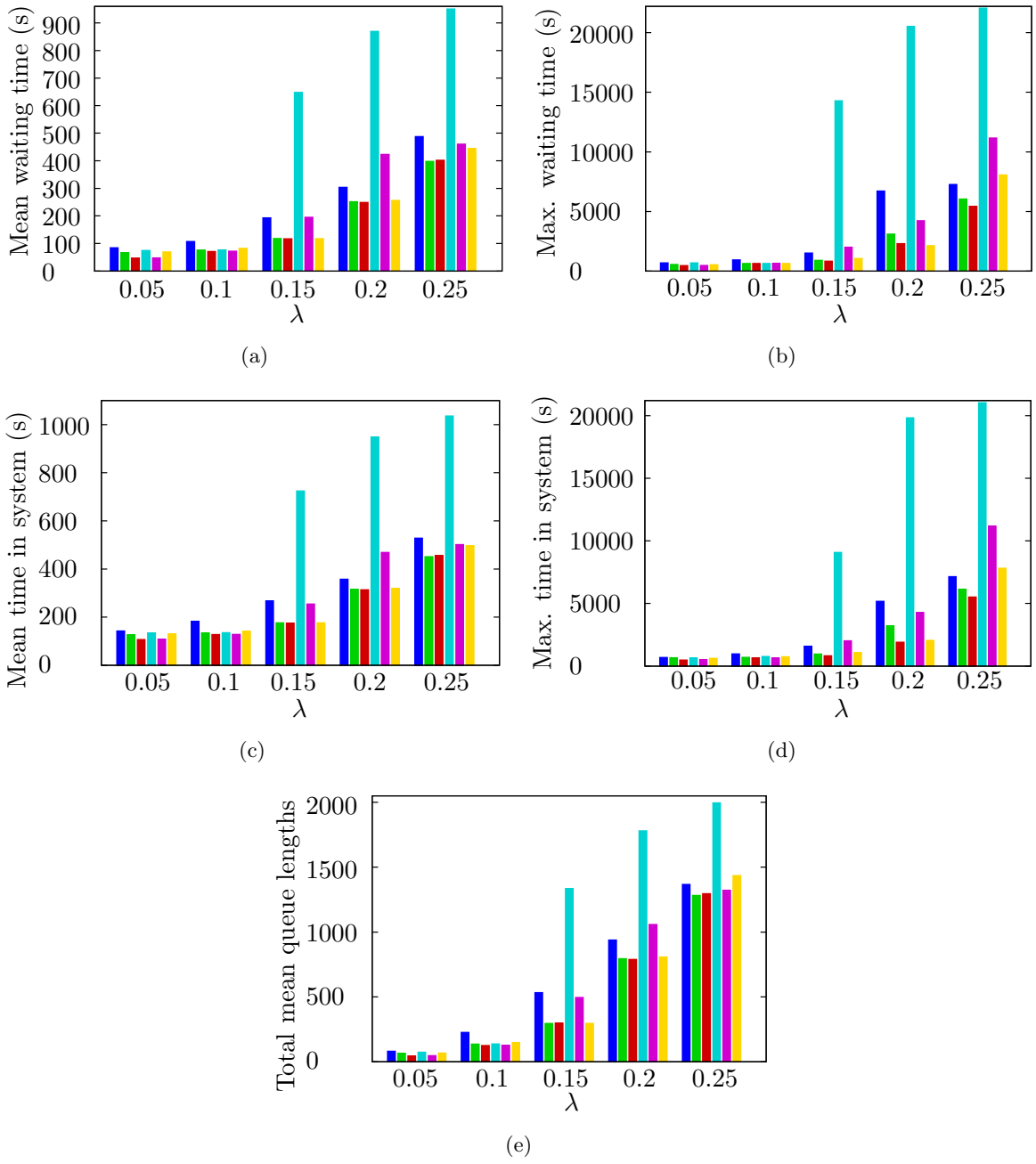


Figure 5.12: Simulation results obtained for a three-by-three grid of intersections in which the algorithm parameters are adjusted for each value of λ separately such that an optimal performance measure value may be found.

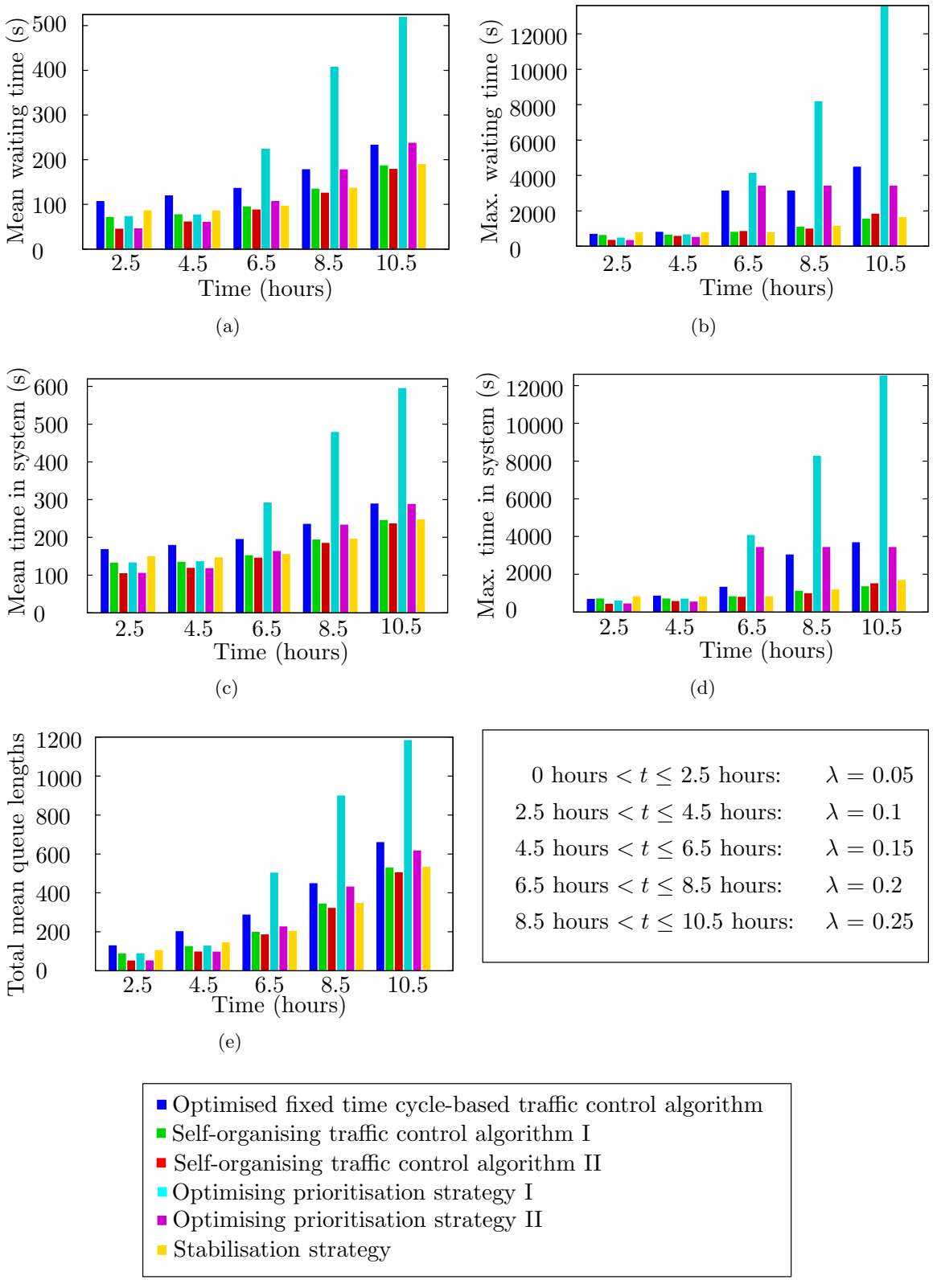


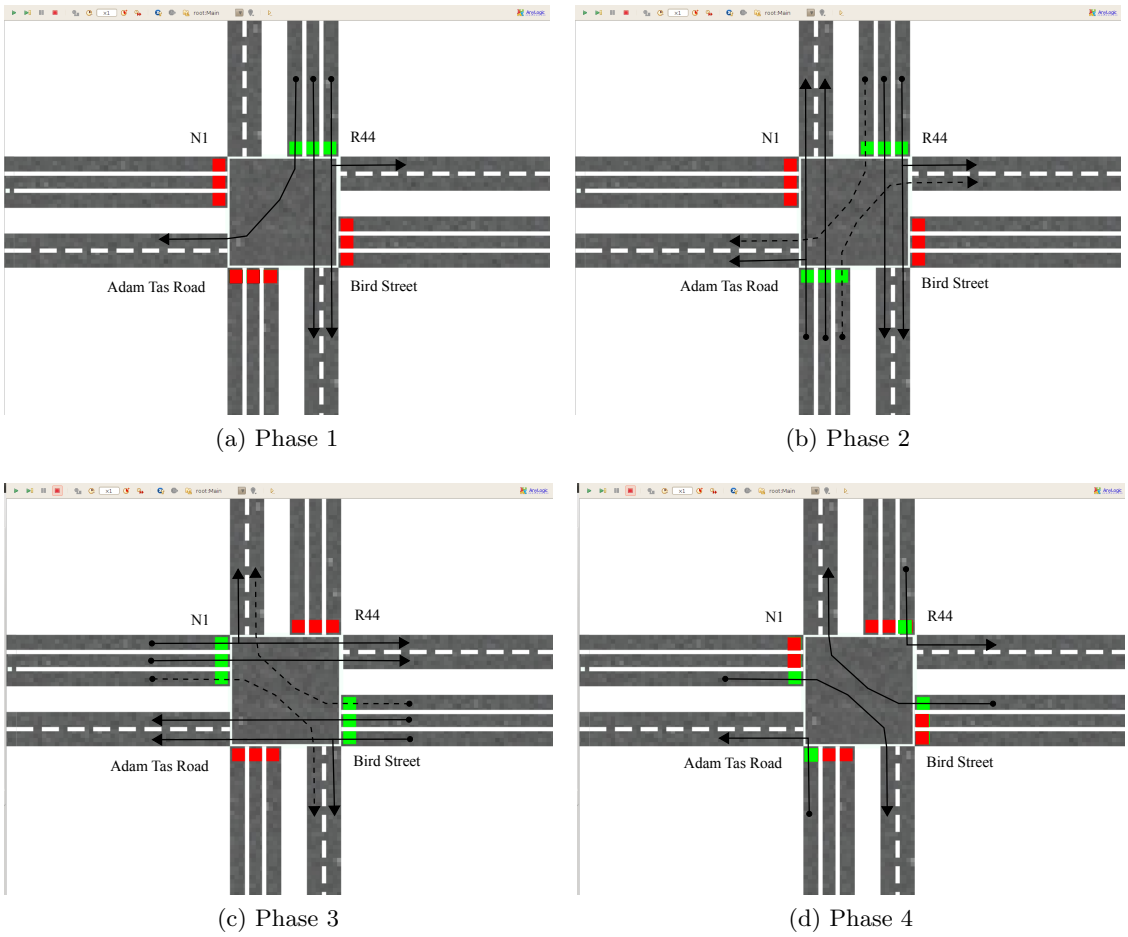
Figure 5.13: Results obtained for fixed parameter combinations and varying values of λ for a three-by-three grid of intersections.

Figure 5.14: Aerial view of the Adam Tas Road & Bird Street Intersection [17].

of the traffic through the intersection arrive from the N1 or leave Stellenbosch towards the N1. The N1 is South Africa's largest national highway and is most commonly used by vehicles when travelling between Stellenbosch and surrounding areas, such as Cape Town and Paarl. The majority of the traffic entering into Stellenbosch from the N1 turn off from the N1 onto the R304 and proceed through the intersection. The majority of the traffic flow along the R44 approach to the intersection is from a large residential complex on the outskirts of Stellenbosch, while the Bird Street and Adam Tas Road approaches typically carry vehicles travelling out of Stellenbosch.

For the simulation experiment involving the Adam Tas Road & Bird Street intersection, the same input data were considered as were used in the validation of the model in §4.5, and may be seen in Appendix B. The data were recorded by the Stellenbosch Municipal Traffic Department [44] over an eleven-and-a-half hour interval between 06:30 and 18:00 in fifteen minute intervals. From these data average arrival profiles as well as turning probabilities were constructed for each of the intersection approaches for consecutive fifteen minute intervals. The purpose of the case study was to investigate the performances of the self-organising traffic control algorithms presented in this thesis and compare them to those of the currently implemented vehicle actuated control regimes of the Adam Tas Road & Bird Street intersection. To accomplish this it was necessary to adopt the self-organising traffic control algorithms for the phase cycle of the traffic signals currently implemented at the intersection. These phases are shown in Figure 5.15.

Figure 5.15(a) shows that during the first phase of the traffic control cycle all vehicles approaching the intersection along the R44 may travel freely through the intersection whether they are proceeding straight through onto Bird Street or turning left or right onto Adam Tas road or towards the N1, respectively. During the second phase, vehicles arriving along the Adam Tas



(a) Phase 1 (b) Phase 2

(c) Phase 3 (d) Phase 4

Figure 5.15: The four different phases of one complete cycle and their associated vehicle movements at the Adam Tas Road & Bird Street intersection in Stellenbosch, South Africa.

Road approach receive a green signal and all vehicles intending to turn right from the R44 or Adam Tas road may only do so when large enough gaps in on coming traffic permit them to do so, as is shown in Figure 5.15(b). During Phase 3, shown in Figure 5.15(c), all vehicles arriving from the N1 as well as along Bird Street receive a green signal with right turns again being permissive. In the final phase of the traffic cycle, Phase 4, all vehicles arriving from the N1 and turning right onto the R44 as well as those arriving along Bird Street and turning right onto Adam Tas road are provided with a protected right turn, and all vehicles turning left from the R44 or Adam Tas Road are permitted to do so, since they do not interfere with the aforementioned protected right-turns, as is shown in Figure 5.15(d).

The arrival patterns of traffic to the intersection were divided into three distinguishable categories. These were a morning peak traffic flow period which contained traffic data collected between 06:30 and 09:00, a midday traffic flow period which contained data collected between 09:00 and 16:30 and an afternoon peak traffic flow period which contained data collected between 16:30 and 18:00. The current traffic signal control regime of the intersection is vehicle actuated and as a result the timings associated with each signal phase are not constant. Samples of the individual phase timings for each period of the day were available, and from these timings the average time of each phase was calculated and implemented as an approximation to the actual phase timings for each distinct traffic flow period throughout the day. These time

samples may be found in Appendix A.

When testing the various self-organising traffic control algorithms, their parameter combinations remained fixed throughout the duration of the simulation run. The results obtained in terms of the same performance measures considered in the previous sections of this chapter are presented in Figure 5.16.

Observations of the various performance measures were recorded every two and a half hours in terms of the cumulative values up to that time of day. From the graphs in Figure 5.16 it may clearly be seen that maximum values attained for each of the performance measures occur during the morning-peak traffic flow conditions. These relatively large traffic flow volumes are associated with the commutes of motorists towards their places of work. A large majority of these commuters work outside Stellenbosch in surrounding areas, such as Cape Town and Durbanville, which are accessible via the N1. This increases traffic through the intersection, since the road leading from the intersection to the N1 is the most commonly used road when travelling between Stellenbosch and the N1. The traffic flow volumes may be seen to subside somewhat during the midday traffic flow period only to increase once again during the afternoon peak-traffic flow period. The increase in traffic flow during the afternoon peak-traffic flow period corresponds with commuters returning home from their places of work. These volumes are not as large as those experienced during the morning-peak traffic flow period, possibly because the times at which motorists leave work to return home are spaced over a larger time horizon.

In Figures 5.16(b) and 5.16(d) it may be seen that the maximum waiting time of vehicles in the system, as well as the maximum time spent by vehicles in the system are experienced during the morning-peak traffic flow period. Figures 5.16(a), 5.16(c) and 5.16(e) all indicate that SOTCA I and SOTCA II both outperform the currently implemented vehicle actuated regime of the intersection, abbreviated here as CIVACR, for the morning-peak traffic flow period. This may be attributed to the superior queue stabilisation techniques employed by SOTCA I and SOTCA II. In the same graphs, it may be seen that during the midday traffic flow period SOTCA II outperforms both SOTCA I and CIVACR. This may be due to the superior self-organising logic of SOTCA II, which is made evident during periods of lighter traffic flow. For the same period of the day SOTCA I performs on par with CIVACR. In the case of the afternoon-peak traffic flow period, however, CIVACR may be seen to outperform SOTCA I in terms of reducing queue lengths and thereby vehicle waiting times. A reason for this is that the optimising or prioritisation strategy of SOTCA I fails to maintain queue length stability as vehicle arrivals to the intersection increase and, as a result, switching of the traffic signals is predominantly controlled by the stabilisation strategy of SOTCA I, as may be seen by the similarities in performance of SOTCA I and the SS. The optimising or prioritisation strategy of SOTCA II, on the other hand, is able to prevent congestion up along the intersection approaches in spite of an increase in traffic flow volumes and therefore the intervention of the stabilisation strategy is less frequent. This is indicated by the similarities in performance of SOTCA II and OPS II. The improvements in the performances of SOTCA II, and in some instances SOTCA I, over CIVACR are expected to become more pronounced if they are to be implemented at the intersections adjacent to the Adam Tas Road & Bird Street intersection, based on the findings of their performances in the cases of the larger signalised traffic networks in §5.3.2 and §5.3.3.

5.3.5 Concluding remarks

Based on the findings of this simulation study, there seems to be a clear advantage in implementing self-organising traffic control regimes instead of fixed-time cycle-based regimes. Of the

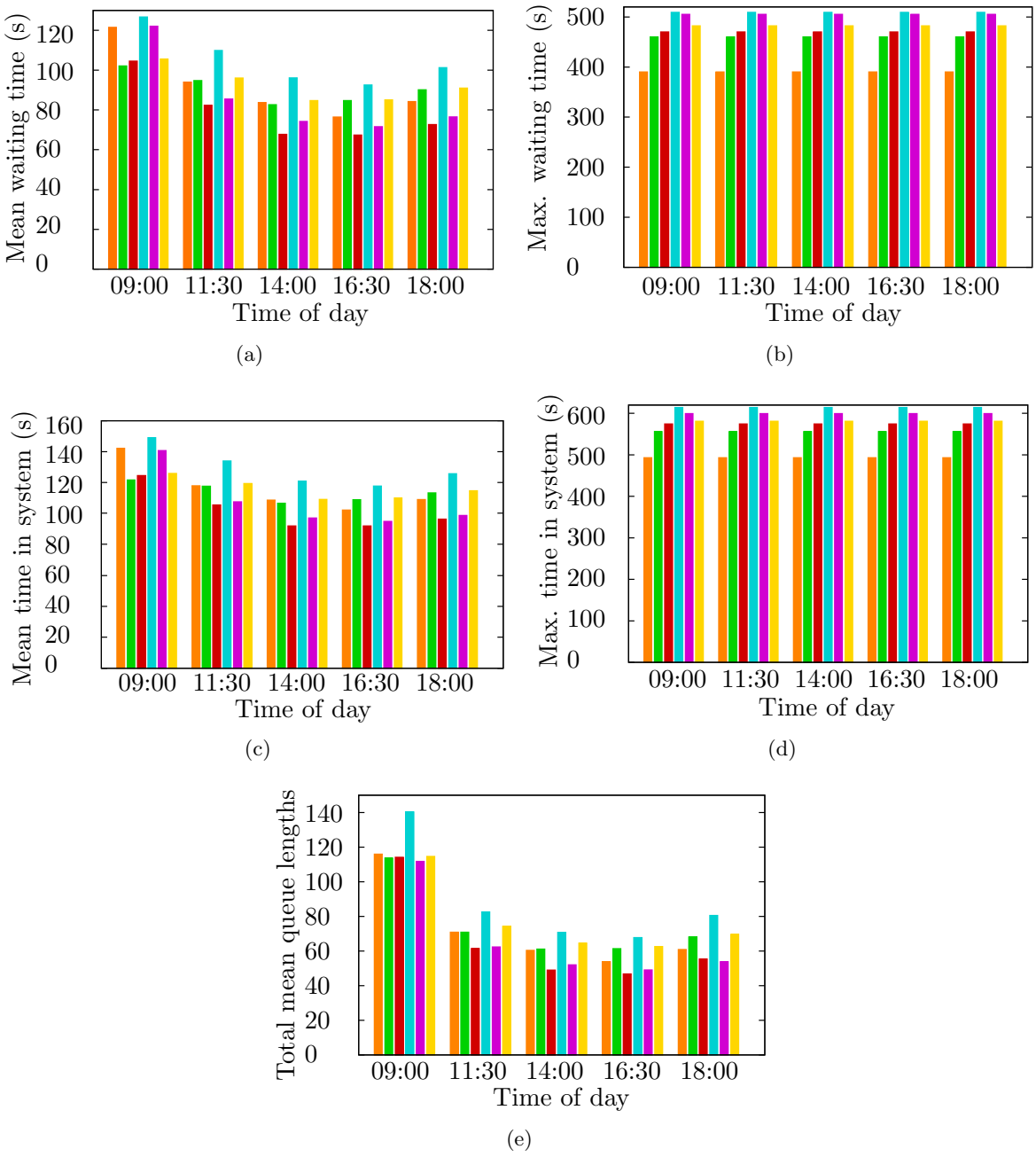


Figure 5.16: Simulation results obtained for fixed parameter combinations and for real arrival rates at the Adam Tas Road & Bird Street intersection.

algorithms tested it was found that SOTCA II was superior for lower traffic volumes in terms of minimising vehicle waiting times in the system, the time a vehicle spends in the system, and the queue lengths along intersection approaches in the system. This may be attributed to the superior performance of the optimising or prioritisation strategy of SOTCA II. It is believed that with the introduction of a more appropriate stabilisation strategy, SOTCA II may well outperform SOTCA I for larger traffic flow volumes. The values of the maximum waiting times of the vehicles in the system as well as the maximum time they spend in the system were shown so as to indicate possible worst-case scenarios, but due to the large variability associated with these observations, they may not be considered to be accurate performance indicators of the traffic control algorithms implemented.

5.4 Summary

In this chapter the algorithms selected for implementation and testing were described in terms of their logic and inner-workings. The experimental design processes followed for each simulation run were explained in the case of each traffic control algorithm implemented. This was followed by the presentation and interpretation of the results obtained by each of the control algorithms tested for a variety of traffic network topologies, as well as for a real case study performed on the Adam Tas Road & Bird Street intersection in Stellenbosch, South Africa.

CHAPTER 6

Conclusion

Contents

This chapter contains three sections. In §6.1 the contents of the thesis are summarised, in §6.2 an appraisal of the contributions of the thesis is presented, and in §6.3 suggestions are made with respect to possible future work.

6.1 Thesis Summary

In the introductory chapter to this thesis, Chapter 1, some of the detrimental effects that traffic congestion has on the environment, economies and societies were presented according to the findings of two reports published by IBM. Possible solutions that have been proposed to ease traffic congestion were discussed with a focus on a self-organising approach towards traffic control which utilises radar detection equipment. The scope and objectives of the study were presented, as was a brief outline of the thesis organisation.

Chapter 2 contains a literature review with respect to the basic theory associated with the dynamics of traffic flow in road traffic networks and at signalised intersections in accordance with Thesis Objective 1(a), as outlined in §1.3. Various mathematical models proposed in the literature to describe the flow of traffic along a road section were also researched and were presented in fulfilment of Thesis Objective 1(b). The chapter closed with a discussion on various self-organising traffic control techniques proposed in the literature, as required by Thesis Objective 1(c).

The third chapter was devoted to a discussion on various computer simulation modelling techniques and approaches in the literature, in fulfilment of Objective 1(d). The advantages and disadvantages of using simulation as a research tool were considered. The chapter ended with a description of various traffic simulation packages commercially available.

Chapter 4 contains a thorough description of the traffic simulation model built for the purpose of this study, as required by Thesis Objective 2(a). The procedures followed to implement each component of the simulation model were described. This was followed by a description of the performance measures considered when evaluating the efficiency of the various traffic control algorithms implemented. An investigation into the effects of incorporating vehicle accelerations and decelerations into the model was performed in an attempt to ascertain whether they were necessary in the simulation study. The chapter closed with a description of the verification and validation processes followed in order to show that model had been built correctly and is capable of producing accurate and reliable output, in accordance with Objectives 2(b) and 2(c), respectively.

The various algorithms that were selected for implementation were described in Chapter 5. The experimental design and initialisation procedures associated with each simulation run were explained. The results obtained from the implementation and testing of the various traffic control algorithms were presented for a variety of traffic network topologies, as required by Thesis Objective 2(d), and analysed in fulfilment of Thesis Objective 4. The chapter closed with a real case study of the Adam Tas Road and Bird Street intersection in Stellenbosch, South Africa, in which the performances of the selected traffic control algorithms are compared against those of the currently implemented control regimes in fulfilment of Thesis Objective 3.

6.2 An appraisal of the contributions of this thesis

The four main contributions of this thesis towards the fields of traffic simulation modelling and self-organised traffic control are presented in this section.

Contribution 1 *The development of a user-friendly traffic simulation tool.*

A microscopic traffic simulation tool was built in the simulation software suite AnyLogic 6.5 [48]. The vehicle motions were modelled according to the vehicle following concept presented in §2.3.1 and incorporate vehicle accelerations and decelerations. The simulation tool allows a user to investigate the effects of altering model parameters on the performance of the system dynamically and provides a comprehensive statistical analysis of key performance measures, the results of which may be exported seamlessly as text files or Microsoft Excel workbooks.

Contribution 2 *The validation of incorporating vehicle accelerations and decelerations into a traffic simulation model.*

The effects of individual vehicle accelerations and decelerations on observations made via a traffic simulation model in terms of vehicle waiting and delay times are considerable. However, due to the complex nature of accurately modelling these accelerations, it is often suggested in the literature that they be omitted, instead opting to account for the time losses due to finite acceleration and driver reactions through the introduction of an analytically determined constant. In §4.3 a simple experiment was conducted in which both approaches were implemented for a single, isolated intersection. It was found that there were noticeable discrepancies between the results obtained via the simulation which included vehicle accelerations and the simulation which did not. More specifically, it was shown that the method in which the time losses due to acceleration were accounted for by the introduction of an analytically determined delay constant

tended to underestimate the waiting time of vehicles in the system and the total time spent by vehicles in the system as well as the actual queue lengths. These discrepancies are expected to grow in magnitude as the traffic network topology increases in complexity (*i.e.* with an increase in the number of intersections in the simulation model). These observations led the author to conclude that, depending on the level of detail and accuracy required of a traffic simulation model, it may not be adequate to omit vehicle accelerations and decelerations in an attempt to incorporate them analytically.

Contribution 3 *The introduction of an alternative optimising prioritisation rule.*

In §5.1.2, an alternative to SOTCA I, the self-organising traffic control algorithm proposed by Lämmer and Helbing [25] (see Algorithm 5.6), was introduced. The algorithms both combine an optimising or prioritisation strategy with a stabilisation strategy. The difference between them lies in the prioritisation strategy. The prioritisation strategy of SOTCA I, called OPS I (see Algorithm 5.4), gives priority to the traffic flow which maximises the expected number of vehicles to be served for a period during which a green signal is anticipated to be shown while accounting for the time losses associated with switching service between traffic flows. The proposed prioritisation strategy, called OPS II (see Algorithm 5.7), places priority on the traffic flow in which the larger number of vehicles is closer to the intersection which, therefore, require service more urgently. From the simulation experiments it follows that the differences in performance between OPS I and OPS II were considerable, and it may be seen from each of the graphs in §5.3 that OPS II outperformed OPS I, particularly in the case of higher traffic flow volumes for larger traffic networks. This is attributed to the improved aspect of global coordination achieved through the local implementation of OPS II at each intersection in the traffic network as well as the poor performance of the anticipatory mechanism of OPS I for higher traffic flow volumes. In summary, SOTCA II was shown to outperform SOTCA I in cases of lower traffic flow volumes, before the stabilisation strategy of each algorithm was required to intervene for larger traffic flow volumes.

Contribution 4 *Experimental evidence highlighting the advantages of self-organising traffic control.*

It is conceded that the results of this study were obtained in a simulated environment which is abstracted from reality through the imposition of certain simplifying assumptions. These assumptions, however, are not considered significant enough that through their inclusion in the simulation model they would greatly compromise or alter the results of the study. It is concluded, therefore, that while it is recommended that the accuracy of the model be improved upon, there seems to be compelling circumstantial evidence to suggest that the question posed in §1.2 may be answered in the affirmative, based on the results obtained by the simulation model of Chapter 4, particularly for the real case study performed on the Adam Tas Road and Bird Street intersection.

6.3 Possible future work

As with most research projects, there is often not enough time available to investigate every aspect of the specific problem at hand. This section is dedicated to highlighting ideas for further work related to improving traffic flow control through implementation of self-organising traffic

control algorithms at signalised intersections in a traffic network.

6.3.1 Improving traffic simulation model accuracy

In order to improve upon the accuracy of the simulation model it is suggested that the scope of the model be widened in an attempt to incorporate a greater number of real-world characteristics associated with traffic flow and traffic flow control. More specifically, these characteristics may include vehicles of varying size, which have varying maximum speeds. The rates of acceleration and deceleration may also vary for each vehicle, as is the case along real-world road sections. It is also suggested that further research take place into more detailed mathematical models describing the motion of individual vehicles along a road section. One such example is that of a vehicle following model which accounts for driver reaction times associated with braking and accelerating.

6.3.2 Investigating alternative self-organising rules

Before any investigation can take place in terms of alternative rules for self-organising traffic control algorithms, it is suggested that a more thorough investigation is performed in the form of a sensitivity analysis with respect to the parameters of the algorithms presented in §5.1. In particular, it would be beneficial if a relationship could be determined between the arrival rate of vehicles into the system and the optimal parameter values and combinations in the self-organising traffic control algorithms (U and $U^{\max}$ in the case of SOTCA I, and U, $U^{\max}$ and ϵ in the case of SOTCA II).

In terms of investigating alternative self-organising rules it is suggested that efforts be made with respect to the research and development of alternative optimising or prioritisation strategies and stabilisation strategies, as well as their combinations. An ideal self-organising traffic control algorithm would be free of any predetermined parameters, *i.e.* the optimal operation of the traffic control algorithm would depend solely on the input received from the immediate local traffic conditions.

It is suggested that clustering techniques be considered in determining alternative optimising prioritisation strategies in which the size of a platoon or cluster of vehicles is considered rather than individual vehicles themselves. Also, the critical values which are used to determine when a group of vehicles may be considered a cluster require further investigation and motivation. It is also suggested that the speed of vehicles feature more prominently such that priority is placed on vehicles which will occupy the intersection for a shorter period of time depending on their distance from the intersection and the speed at which they are travelling.

6.3.3 Improved real-world case study

In an attempt to better gauge the performance of any self-organising traffic control algorithm it is suggested that that simulation runs be implemented for actual road network topologies, using actual data relevant to the road network which is being modelled. The corridor along the Adam Tas Road in Stellenbosch, South Africa would, for example, make a good case study. Between the Adam Tas Road and Dorp Street intersection, and the Adam Tas Road and Bird Street intersection, there are six signalised intersections (inclusive). Traffic along the corridor is known to become heavily congested during certain hours of the day, making it an appropriate site for investigating the potential effectiveness of self-organising traffic control algorithms.

The quality of the output of a simulation model, however, depends on the quality of the data which are used as input to the simulation model. For this reason, accurate data which are relevant to the road network being investigated is crucial in order to validate the results of the simulation model. Due to the fact that the data necessary for the execution of a simulation study on a microscopic level are not readily available, or easily obtainable, it is suggested that various traffic data collection methods and techniques are researched and implemented so as to provide a solid starting point for a proper case study.

Bibliography

[1] Agent KR & Deen RC, 1979, *Warrants for left-turn signal phasing*, (Unpublished) Technical Report 737, Transportation Research Board, Washington (DC).

[2] Allsop RE, 1972, *Delay at a fixed time traffic signal: I Theoretical analysis*, Transportation Science, **6(3)**, pp. 260–285.

[3] Banks J, 1998, *Principles of simulation*, pp. 3–30 in Banks J (Ed), *Handbook of simulation: Principles, methodology, advances, applications and practice*, Wiley-Interscience, New York (NY).

[4] Banks J, Carson JS, Nelson BL & Nicol DM, 2001, *Introduction to simulation*, pp. 3–22 in Fabrycky WJ & Mize JH (Eds), *Discrete-event system simulation*, 3rd Edition, Prentice-Hall, Upper Saddle River (NJ).

[5] Banks J, Carson JS, Nelson BL & Nicol DM, 2001, *Verification and validation of simulation models*, pp. 367–397 in Fabrycky WJ & Mize JH (Eds), *Discrete-event system simulation*, 3rd Edition, Prentice-Hall, Upper Saddle River (NJ).

[6] Barceló J, 2010, *Models, traffic models, simulation and traffic simulation*, pp. 1–62 in Barceló J (Ed), *Fundamentals of traffic simulation*, Springer, New York (NY).

[7] Borschev A & Filippov A, 2004, *Anylogic — Multi-paradigm simulation for business, engineering and research*, Proceedings of the 6th IIE Annual Simulation Solutions Conference.

[8] Borshchev A & Filippov A, 2004, *From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools*, Proceedings of the 22nd International Conference of the System Dynamics Society, pp. 25–29.

[9] Brockfeld E, Barlovic R, Schadschneider A & Schreckenberg M, 2001, *Optimizing traffic lights in a cellular automaton model for city traffic*, Physical Review E, **64(5)**, p. 056132.

[10] Camazine S, 2003, *Self-organization in biological systems*, Princeton University Press, Princeton (NJ).

[11] Department of transport chief directorate: National roads, 1988, *Guidelines for the application of traffic signal phasing and control equipment*, (Unpublished) Technical Report 87, National Transport Commission, Pretoria, South Africa.

[12] Fellendorf M & Vortisch P, 2010, *Microscopic traffic flow simulator VISSIM*, pp. 63–94 in Barceló J (Ed), *Fundamentals of traffic simulation*, Springer, New York (NY).

[13] Fouladvand ME, Shaebani MR & Sadjadi Z, 2004, *Intelligent controlling simulation of traffic flow in a small city network*, Journal of the Physical Society of Japan, **73(11)**, pp. 3209–3214.

[14] Gerlough DL & Huber MJ, 1975, *Traffic flow theory*, (Unpublished) Technical Report 165, Transportation Research Board, Washington (DC).

[15] Gershenson C, 2005, *Self-organizing traffic lights*, Complex Systems, **16(1)**, pp. 29–53.

[16] Gibson D, 1981, *Available computer models for traffic operations analysis*, (Unpublished) Technical Report 194, Transportation Research Board, Washington (DC).

[17] Google, 2011, *Google maps*, [Online], [Cited October 1st, 2011], Available from `http://maps.google.co.za/maps`.

[18] Greenshields BD, Schapiro D & Ericksen EL, 1946, *Traffic performance at urban street intersections*, (Unpublished) Technical Report 1, Bureau of Highway Traffic, Yale University, New Haven (CT).

[19] Helbing D, 2003, *A section-based queueing-theoretical traffic model for congestion and travel time analysis in networks*, Journal of Physics A: Mathematical and General, **36(46)**, pp. L593–L598.

[20] Helbing D, Lämmer S & Lebacque JP, 2005, *Self-organized control of irregular or perturbed network traffic*, Advances in Computational Management Science, **7(4)**, pp. 239–274.

[21] International Business Machines, 2007, *The roads to a smarter planet*, [Online], [Cited October 31st, 2011], Available from `http://www.ibm.com/smarterplanet/global/files/us__en_us__traffic__smarterplanet_traffic.pdf`.

[22] International Business Machines, 2011, *Frustration rising: IBM 2011 commuter pain survey*, [Online], [Cited October 31st, 2011], Available from `http://www-03.ibm.com/press/us/en/presskit/35314.wss`.

[23] Kell JH & Fullerton IJ, 1982, *Manual of traffic signal design*, Prentice-Hall, Eaglewood Cliffs (NJ).

[24] Lämmer S, Donner R & Helbing D, 2008, *Anticipative control of switched queueing systems*, The European Physical Journal B — Condensed Matter and Complex Systems, **63(3)**, pp. 341–347.

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

[26] Law AM, 2000, *Building valid, credible, and appropriately detailed simulation models*, pp. 243–276 in *Simulation modelling and analysis*, 4th Edition, McGraw-Hill, Boston (MA).

[27] Law AM, 2007, *Output data analysis for a single system*, pp. 485–547 in *Simulation modelling and analysis*, 4th Edition, McGraw-Hill, Boston (MA).

[28] Lighthill MJ & Whitham G, 1955, *On kinematic waves: II A theory of traffic flow on long crowded roads*, Proceedings of the Royal Society A, **229(1178)**, pp. 317–345.

[29] Miller AJ, 1963, *Settings for fixed-cycle traffic signals*, Operations Research Quarterly, **14(4)**, pp. 373–386.

[30] Newell GF, 1956, *Statistical analysis of the flow of highway traffic through a signalized intersection*, Quarterly of Applied Mathematics, **13(4)**, pp. 353–369.

[31] Newell GF, 1965, *Statistical analysis of the flow of highway traffic through a signalized intersection*, SIAM Review, **7(2)**, pp. 223–239.

[32] Papacostas CS & Prevedouros PD, 2001, *Capacity and level of service analysis*, pp. 133–230 in Curless L (Ed), *Transportation engineering and planning*, Prentice-Hall, Upper Saddle River (NJ).

[33] Papacostas CS & Prevedouros PD, 2001, *Queueing and simulation*, pp. 611–625 in Curless L (Ed), *Transportation engineering and planning*, Prentice-Hall, Upper Saddle River (NJ).

[34] Papacostas CS & Prevedouros PD, 2001, *Traffic stream flow models*, pp. 100–132 in Curless L (Ed), *Transportation engineering and planning*, Prentice-Hall, Upper Saddle River (NJ).

[35] Papacostas CS & Prevedouros PD, 2001, *Transportation software*, pp. 626–652 in Curless L (Ed), *Transportation engineering and planning*, Prentice-Hall, Upper Saddle River (NJ).

[36] Papacostas CS & Prevedouros PD, 2001, *Urban and intelligent transportation systems*, pp. 263–317 in Curless L (Ed), *Transportation engineering and planning*, Prentice-Hall, Upper Saddle River (NJ).

[37] Pignataro LJ, Cantilli EJ & Falcocchio JC, 1973, *Traffic engineering: Theory and practice*, Prentice-Hall, Eaglewood Cliffs (NJ).

[38] Rouphail NM, 1986, *Analytical warrant for separate left turn phasing*, (Unpublished) Technical Report 1069, Transportation Research Board, Washington (DC).

[39] SCOOT, 2009, *How SCOOT works*, [Online], [Cited October 19th, 2011], Available from `http://www.scoot-utc.com/documents/1_SCOOT-UTC.pdf`.

[40] Serugendo GDM, Foukia N, Hassas S, Karageorgos A, Mostéfaoui SK, Rana OF, Ulieru M, Valckenaers P & Aart CV, 2004, *Self-organisation: Paradigms and applications*, Lecture Notes in Computer Science, **2997**, pp. 1–19.

[41] Traffic Management Technologies, 2006, *Wavetronix SmartSensor Advance Model 200*, [Online], [Cited October 31st, 2011], Available from `http://www.tmtservices.co.za/changes2/images/products/TMT_WT_SSADV.pdf`.

[42] Transportation Research Board, 1994, *Highway capacity manual*, (Unpublished) Technical Report 209, National Research Council, Washington (DC).

[43] Upchurch JE, 1986, *Guidelines for selecting type of left-turn phasing*, (Unpublished) Technical Report 1069, Transportation Research Board, Washington (DC).

[44] Van der Merwe E, 2009, *Improving the flow of an isolated traffic system*, Fourth Year Engineering Thesis, Stellenbosch University, Stellenbosch.

[45] Vermeulen MJ, 1984, *Pretimed control of arterial traffic signal systems*, PhD Dissertation, University of California, Berkley (CA).

[46] Waterhouse N, 2010, *What can we learn from bees about how to work smarter?*, [Online], [Cited November 4th, 2011], Available from `http://firmfollowsform.com/?p=98`.

[47] Webster FV, 1958, *Traffic signal settings*, (Unpublished) Technical Report 39, Her Majesty's Stationery Office, London.

[48] XJ Technologies, 2010, *Anylogic help*, [Online], [Cited August 4th, 2011], Available from `http://www.xjtek.com/`.

[49] Young HD & Freedman RA, 2000, *University physics*, 10th Edition, Addison Wesley Longman Inc., San Francisco (CA).

APPENDIX A

Input Data for The Adam Tas and Bird Street Intersection

Contents

This appendix contains the observed vehicle proportions for each of the approaches to the Adam Tas Road & Bird Street intersection, as well as samples of the observed green times from the morning-peak, midday, and afternoon-peak traffic flow periods which were used as input to the traffic simulation model in §5.3.4.

A.1 Vehicle movement proportions

This section contains the observed vehicle proportions for each of the approaches to the Adam Tas Road & Bird Street intersection in Stellenbosch, South Africa.

Table A.1: Turning proportions of vehicles approaching along Adam Tas Road.

Vehicles approaching along Adam Tas Road							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
06:30–06:45	155	48	105	2	31	68	1
06:45–07:00	161	60	101	0	37	63	0
07:00–07:15	256	90	165	1	35	64	0
07:15–07:30	193	57	132	4	30	68	2
07:30–07:45	155	30	122	3	19	79	2
07:45–08:00	150	36	112	2	24	75	1
08:00–08:15	226	65	155	6	29	69	3
08:15–08:30	159	63	94	2	40	59	1
08:30–08:45	137	41	91	5	30	66	4

Continued on next page

Table A.1 – continued from previous page

Vehicles approaching along Adam Tas Road							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
08:45–09:00	166	63	97	6	38	58	4
09:00–09:15	160	62	95	3	39	59	2
09:15–09:30	154	57	94	3	37	61	2
09:30–09:45	164	52	103	9	32	63	5
09:45–10:00	151	52	98	1	34	65	1
10:00–10:15	136	49	85	2	36	63	1
10:15–10:30	147	47	93	7	32	63	5
10:30–10:45	172	47	122	3	27	71	2
10:45–11:00	120	44	74	2	37	62	2
11:00–11:15	171	55	111	5	32	65	3
11:15–11:30	173	68	103	2	39	60	1
11:30–11:45	122	38	81	3	31	66	2
11:45–12:00	158	53	102	3	34	65	2
12:00–12:15	205	73	123	9	36	60	4
12:15–12:30	121	48	66	7	40	55	6
12:30–12:45	152	52	96	4	34	63	3
12:45–13:00	164	52	107	5	32	65	3
13:00–13:15	198	76	114	8	38	58	4
13:15–13:30	154	53	97	4	34	63	3
13:30–13:45	156	38	108	10	24	69	6
13:45–14:00	173	72	98	3	42	57	2
14:00–14:15	132	52	77	3	39	58	2
14:15–14:30	184	58	120	6	32	65	3
14:30–14:45	211	65	144	2	31	68	1
14:45–15:00	166	52	109	5	31	66	3
15:00–15:15	144	53	90	1	37	63	1
15:15–15:30	200	74	121	5	37	61	3
15:30–15:45	152	49	99	4	32	65	3
15:45–16:00	202	80	107	15	40	53	7
16:00–16:15	100	32	67	1	32	67	1
16:15–16:30	185	76	105	4	41	57	2
16:30–16:45	194	64	125	5	33	64	3
16:45–17:00	248	80	157	11	32	63	4
17:00–17:15	175	55	120	0	31	69	0
17:15–17:30	194	66	124	4	34	64	2
17:30–17:45	279	90	187	2	32	67	1
17:45–18:00	206	72	132	2	35	64	1

Table A.2: Turning proportions of vehicles approaching along Bird Street.

Vehicles approaching along Bird Street							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
06:30–06:45	44	1	31	12	2	70	27
06:45–07:00	77	2	57	18	3	74	23
07:00–07:15	101	0	62	39	0	61	39
07:15–07:30	98	2	66	30	2	67	31
07:30–07:45	133	3	77	53	2	58	40
07:45–08:00	156	3	87	66	2	56	42
08:00–08:15	111	4	63	44	4	57	40
08:15–08:30	137	1	96	40	1	70	29
08:30–08:45	121	4	74	43	3	61	36
08:45–09:00	117	8	70	39	7	60	33
09:00–09:15	112	2	70	40	2	63	36
09:15–09:30	119	6	72	41	5	61	34
09:30–09:45	88	3	51	34	3	58	39
09:45–10:00	120	3	79	38	3	66	32
10:00–10:15	112	3	75	34	3	67	30
10:15–10:30	162	1	106	55	1	65	34
10:30–10:45	106	2	68	36	2	64	34
10:45–11:00	116	6	71	39	5	61	34
11:00–11:15	128	4	78	46	3	61	36
11:15–11:30	147	3	87	57	2	59	39
11:30–11:45	85	3	60	22	4	71	26
11:45–12:00	136	4	81	51	3	60	38
12:00–12:15	171	6	114	51	4	67	30
12:15–12:30	130	1	72	57	1	55	44
12:30–12:45	116	2	75	39	2	65	34
12:45–13:00	159	7	100	52	4	63	33
13:00–13:15	185	9	110	66	5	59	36
13:15–13:30	153	8	88	57	5	58	37
13:30–13:45	149	5	95	49	3	64	33
13:45–14:00	170	6	98	66	4	58	39
14:00–14:15	111	5	79	27	5	71	24
14:15–14:30	168	3	96	69	2	57	41
14:30–14:45	178	2	111	65	1	62	37
14:45–15:00	136	2	86	48	1	63	35
15:00–15:15	156	6	105	45	4	67	29
15:15–15:30	178	9	108	61	5	61	34
15:30–15:45	178	8	114	56	4	64	31
15:45–16:00	168	6	121	41	4	72	24
16:00–16:15	151	4	99	48	3	66	32
16:15–16:30	195	2	131	62	1	67	32
16:30–16:45	227	0	135	92	0	59	41
16:45–17:00	269	7	155	107	3	58	40

Continued on next page

Table A.2 – continued from previous page

Vehicles approaching along Bird Street							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
17:00–17:15	243	2	145	96	1	60	40
17:15–17:30	246	1	148	97	0	60	39
17:30–17:45	212	2	116	94	1	55	44
17:45–18:00	215	2	120	93	1	56	43

Table A.3: Turning proportions of vehicles approaching along the R44.

Vehicles approaching along the R44							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
06:30–06:45	121	18	88	15	15	73	12
06:45–07:00	201	37	126	38	18	63	19
07:00–07:15	364	88	233	43	24	64	12
07:15–07:30	262	52	178	32	20	68	12
07:30–07:45	459	151	254	54	33	55	12
07:45–08:00	243	78	127	38	32	52	16
08:00–08:15	260	77	150	33	30	58	13
08:15–08:30	210	54	130	26	26	62	12
08:30–08:45	262	76	153	33	29	58	13
08:45–09:00	90	35	44	11	39	49	12
09:00–09:15	180	56	97	27	31	54	15
09:15–09:30	120	40	63	17	33	53	14
09:30–09:45	179	56	108	15	31	60	8
09:45–10:00	100	36	53	11	36	53	11
10:00–10:15	161	45	83	33	28	52	20
10:15–10:30	160	64	76	20	40	48	13
10:30–10:45	164	57	87	20	35	53	12
10:45–11:00	102	33	54	15	32	53	15
11:00–11:15	144	41	77	26	28	53	18
11:15–11:30	188	59	104	25	31	55	13
11:30–11:45	154	57	77	20	37	50	13
11:45–12:00	169	45	98	26	27	58	15
12:00–12:15	171	60	85	26	35	50	15
12:15–12:30	123	36	71	16	29	58	13
12:30–12:45	177	61	94	22	34	53	12
12:45–13:00	136	39	79	18	29	58	13
13:00–13:15	189	48	117	24	25	62	13
13:15–13:30	175	54	102	19	31	58	11
13:30–13:45	206	61	125	20	30	61	10

Continued on next page

Table A.3 – continued from previous page

Vehicles approaching along the R44							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
13:45–14:00	106	34	61	11	32	58	10
14:00–14:15	185	47	99	39	25	54	21
14:15–14:30	163	47	85	31	29	52	19
14:30–14:45	190	50	120	20	26	63	11
14:45–15:00	167	55	91	21	33	54	13
15:00–15:15	164	52	95	17	32	58	10
15:15–15:30	199	41	119	39	21	60	20
15:30–15:45	128	31	78	19	24	61	15
15:45–16:00	211	59	125	27	28	59	13
16:00–16:15	159	51	86	22	32	54	14
16:15–16:30	239	55	132	52	23	55	22
16:30–16:45	211	60	119	32	28	56	15
16:45–17:00	176	64	89	23	36	51	13
17:00–17:15	202	84	96	22	42	48	11
17:15–17:30	186	71	83	32	38	45	17
17:30–17:45	206	77	109	20	37	53	10
17:45–18:00	188	67	95	26	36	51	14

Table A.4: Turning proportions of vehicles approaching along the N1.

Vehicles approaching along the N1							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
06:30–06:45	157	8	74	75	5	47	48
06:45–07:00	207	8	105	94	4	51	45
07:00–07:15	241	9	141	91	4	59	38
07:15–07:30	220	6	127	87	3	58	40
07:30–07:45	208	12	126	70	6	61	34
07:45–08:00	211	11	135	65	5	64	31
08:00–08:15	252	10	148	94	4	59	37
08:15–08:30	230	13	126	91	6	55	40
08:30–08:45	168	16	104	48	10	62	29
08:45–09:00	207	10	135	62	5	65	30
09:00–09:15	233	15	106	112	6	45	48
09:15–09:30	143	15	109	19	10	76	13
09:30–09:45	175	11	96	68	6	55	39
09:45–10:00	160	8	103	49	5	64	31
10:00–10:15	183	17	103	63	9	56	34
10:15–10:30	169	18	91	60	11	54	36

Continued on next page

Table A.4 – continued from previous page

Vehicles approaching along the N1							
		Total number of vehicles that:			% of vehicles that:		
Time	Total number of vehicles	turn left	travel straight	turn right	turn left	travel straight	turn right
10:30–10:45	177	16	99	62	9	56	35
10:45–11:00	156	16	85	55	10	54	35
11:00–11:15	134	17	76	41	13	57	31
11:15–11:30	168	13	95	60	8	57	36
11:30–11:45	179	19	84	76	11	47	42
11:45–12:00	191	27	90	74	14	47	39
12:00–12:15	142	22	69	51	15	49	36
12:15–12:30	155	10	87	58	6	56	37
12:30–12:45	169	14	86	69	8	51	41
12:45–13:00	139	15	68	56	11	49	40
13:00–13:15	203	14	121	68	7	60	33
13:15–13:30	222	17	130	75	8	59	34
13:30–13:45	107	9	67	31	8	63	29
13:45–14:00	100	4	58	38	4	58	38
14:00–14:15	127	8	63	56	6	50	44
14:15–14:30	135	4	69	62	3	51	46
14:30–14:45	164	3	107	54	2	65	33
14:45–15:00	171	19	86	66	11	50	39
15:00–15:15	159	16	87	56	10	55	35
15:15–15:30	140	14	86	40	10	61	29
15:30–15:45	152	7	86	59	5	57	39
15:45–16:00	159	12	96	51	8	60	32
16:00–16:15	155	13	82	60	8	53	39
16:15–16:30	159	10	95	54	6	60	34
16:30–16:45	170	5	119	46	3	70	27
16:45–17:00	233	13	166	54	6	71	23
17:00–17:15	188	15	117	56	8	62	30
17:15–17:30	208	21	139	48	10	67	23
17:30–17:45	196	16	128	52	8	65	27
17:45–18:00	227	17	166	44	7	73	19

A.2 Individual phase green times

Samples of the green times observed during the morning-peak, midday and and afternoon-peak traffic flow periods for each of the four phases of the traffic signal cycle at the Adam Tas Road & Bird Street intersection are presented in this section. The means of these observed green times were used as representative values for the green times implemented by CIVACR in §5.3.4.

Time of day	Phase 1	Phase 2	Phase 3	Phase 4
07:58	13	50	51	17
08:02	10	48	65	18
08:04	10	53	70	11
08:04	11	60	34	19
08:06	8	55	60	17
08:07	10	45	60	17
08:09	10	56	40	16
08:10	8	64	50	16
08:12	10	70	53	13
08:14	7	70	40	15
08:16	10	58	58	4
08:18	11	48	76	14
08:19	11	50	54	15
08:21	11	45	70	14
Mean	10.00	55.14	55.79	14.71

Table A.5: Observed morning-peak green times (in seconds) at the Adam Tas Road & Bird Street intersection.

Time of day	Phase 1	Phase 2	Phase 3	Phase 4
10:43	12	43	65	15
10:44	15	57	41	15
10:45	14	62	50	15
10:47	12	50	75	16
10:48	10	45	60	17
10:50	13	47	84	0
10:52	13	38	90	0
10:53	10	32	86	0
Mean	12.38	46.75	68.88	9.75

Table A.6: Observed midday green times (in seconds) at the Adam Tas Road & Bird Street intersection.

Time of day	Phase 1	Phase 2	Phase 3	Phase 4
17:18	7	48	80	0
17:21	9	60	48	14
17:23	0	65	56	14
17:26	12	51	65	14
17:28	12	40	70	14
17:31	12	60	80	0
17:33	10	45	77	0
17:35	9	46	75	0
17:38	0	70	70	0
17:40	9	42	75	16
17:43	10	60	38	0
17:45	0	70	80	0
17:48	0	52	80	0
17:50	12	56	65	0
17:52	12	65	60	0
17:55	10	57	80	0
17:57	9	45	90	0
18:00	11	42	75	0
18:02	10	30	80	0
Mean	8.11	52.84	70.74	3.79

Table A.7: Observed afternoon-peak green times (in seconds) at the Adam Tas Road & Bird Street intersection.

APPENDIX B

Experimentation results

Contents

This appendix contains the results that were obtained by each traffic control algorithm described in §5.1 and for each road network topology investigated in §5.3, as well as the results for the case study described in §5.3.4.

B.1 Results obtained for a single, isolated section

The results presented in this section were obtained for a single, isolated intersection. The results for the case in which the algorithm parameters are adjusted for each value of λ separately so that an optimal performance measure value may be found are presented in Tables 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6 in §5.3.1. The results shown in Tables B.1, B.2, B.3, B.4, B.5 and B.6 were obtained for a single, isolated intersection for the case in which the traffic control algorithm parameters remained fixed while the value of λ varied.

OFTTCA					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	37.46	41.44	44.55	51.64	74.67
Green time (s)	80.00	80.00	80.00	80.00	80.00
Max. waiting time (s)	81.00	124.00	464.00	464.00	464.00
Green time (s)	30.00	30.00	30.00	30.00	30.00
Total mean queue length	14.36	21.96	30.42	43.47	72.13
Green time (s)	80.00	80.00	80.00	80.00	80.00
Mean time in system (s)	73.56	77.01	80.28	88.85	112.97
Green time (s)	80.00	80.00	80.00	80.00	80.00
Max. time in system (s)	109.87	154.22	477.22	477.22	477.22
Green time (s)	30.00	30.00	30.00	30.00	30.00

Table B.1: Second set of simulation results obtained by OFTTCA for a single, isolated intersection with corresponding green time values.

SOTCA I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	18.68	21.11	30.77	40.85	63.07
U	120.00	120.00	120.00	120.00	120.00
$U^{\max}$	140.00	140.00	140.00	140.00	140.00
Mean green time (s)	7.70	11.78	17.69	22.75	27.13
Max. waiting time (s)	80.00	120.00	170.00	339.00	375.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	8.16	11.80	16.14	19.65	22.34
Total mean queue lengths	5.96	9.71	19.67	32.60	59.49
U	120.00	120.00	120.00	120.00	120.00
$U^{\max}$	140.00	140.00	140.00	140.00	140.00
Mean green time (s)	7.70	11.78	17.69	22.75	27.13
Mean time in system (s)	50.89	53.20	64.04	75.03	99.55
U	120.00	120.00	120.00	120.00	120.00
$U^{\max}$	140.00	140.00	140.00	140.00	140.00
Mean green time (s)	7.70	11.78	17.69	22.75	27.13
Max. time in system (s)	111.46	143.38	190.07	353.22	380.38
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	8.16	11.80	16.14	19.65	22.34

Table B.2: Second set of simulation results obtained by SOTCA I for a single, isolated intersection with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

SOTCA II					
	λ				
	0.05	0.10	0.15	0.20	0.25
Mean waiting time (s)	13.82	20.04	30.41	41.08	64.88
U	100.00	100.00	100.00	100.00	100.00
$U^{\max}$	160.00	160.00	160.00	160.00	160.00
Mean green time (s)	15.62	15.55	18.00	22.91	27.04
Max. waiting time (s)	89.00	147.00	218.00	349.00	376.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	15.91	16.70	20.17	23.87	26.69
Total mean queue lengths	4.05	7.82	18.31	30.19	60.72
U	100.00	100.00	100.00	100.00	100.00
$U^{\max}$	180.00	180.00	180.00	180.00	180.00
Mean green time (s)	14.43	14.39	16.77	21.20	25.13
Mean time in system (s)	45.68	51.46	63.02	74.30	100.61
U	100.00	100.00	100.00	100.00	100.00
$U^{\max}$	160.00	160.00	160.00	160.00	160.00
Mean green time (s)	15.62	15.55	18.00	22.91	27.04
Max. time in system (s)	120.22	169.19	239.07	355.38	387.27
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	15.91	16.70	20.17	23.87	26.69

Table B.3: Second set of simulation results obtained by SOTCA II for a single, isolated intersection with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OPS I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	18.65	20.46	66.33	113.55	144.96
Mean green time (s)	8.13	12.17	19.22	26.66	34.02
Max. waiting time (s)	98.00	106.00	676.00	995.00	1083.00
Mean green time (s)	8.13	12.17	19.22	26.66	34.02
Total mean queue lengths	5.74	9.28	46.35	87.76	116.90
Mean green time (s)	8.13	12.17	19.22	26.66	34.02
Mean time in system (s)	50.92	52.73	104.09	155.88	188.92
Mean green time (s)	8.13	12.17	19.22	26.66	34.02
Max. time in system (s)	116.85	128.19	698.19	1016.16	1098.38
Mean green time (s)	8.13	12.17	19.22	26.66	34.02

Table B.4: Second set of simulation results obtained by OPS I for a single, isolated intersection with the resulting mean green time values.

OPS II					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	14.84	19.40	54.60	87.13	112.23
Mean green time (s)	14.51	14.27	14.45	14.80	15.96
Max. waiting time (s)	116.00	141.00	654.00	1004.00	1112.00
Mean green time (s)	14.51	14.27	14.45	14.80	15.96
Total mean queue lengths	4.35	8.23	33.99	64.21	90.09
Mean green time (s)	14.51	14.27	14.45	14.80	15.96
Mean time in system (s)	46.69	50.71	84.72	116.04	139.92
Mean green time (s)	14.51	14.27	14.45	14.80	15.96
Max. time in system (s)	149.45	168.22	667.22	1017.77	1125.22
Mean green time (s)	14.51	14.27	14.45	14.80	15.96

Table B.5: Second set of simulation results obtained by OPS II for a single, isolated intersection with the resulting mean green time values.

SS					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	32.56	33.78	35.57	40.41	66.03
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	160.00	160.00	160.00	160.00	160.00
Mean green time (s)	101.26	90.92	83.52	79.68	78.14
Max. waiting time (s)	126.00	185.00	195.00	329.00	398.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	160.00	160.00	160.00	160.00	160.00
Mean green time (s)	60.73	58.85	56.86	55.81	55.17
Total mean queue lengths	16.18	20.46	25.08	32.68	62.62
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	150.26	113.16	91.12	84.25	81.70
Mean time in system (s)	80.68	74.76	72.83	76.05	103.58
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	150.26	113.16	91.12	84.25	81.70
Max. time in system (s)	159.08	208.07	208.07	341.96	411.22
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	160.00	160.00	160.00	160.00	160.00
Mean green time (s)	60.73	58.85	56.86	55.81	55.17

Table B.6: Second set of simulation results obtained by the SS for a single, isolated intersection with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

B.2 Results obtained for a two-by-two grid of intersections

The results presented in this section were obtained for a two-by-two grid of intersections. The results shown in Tables B.7, B.8, B.9, B.10, B.11 and B.12 were obtained for a two-by-two grid of intersections for the case in which the algorithm parameters were adjusted separately for each value of λ so that an optimal performance measure value could be attained. The results shown in Tables B.13, B.14, B.15, B.16, B.17 and B.18 were obtained for a two-by-two grid of intersections for the case in which the traffic control algorithm parameters remained fixed while the value of λ varied.

OFTTCA					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	63.82	69.99	95.46	201.95	307.58
Green time (s)	20.00	40.00	60.00	90.00	90.00
Max. waiting time (s)	249.00	443.00	776.00	2464.00	2777.00
Green time (s)	20.00	40.00	60.00	90.00	90.00
Total mean queue length	41.09	81.43	155.92	431.38	711.77
Green time (s)	20.00	40.00	60.00	90.00	90.00
Mean time in system (s)	112.40	116.57	142.75	254.64	363.87
Green time (s)	20.00	40.00	60.00	90.00	90.00
Max. time in system (s)	412.16	604.21	1159.49	2003.91	7971.14
Green time (s)	20.00	40.00	60.00	90.00	90.00

Table B.7: First set of simulation results obtained by OFTTCA for a two-by-two grid of intersection with corresponding green time values.

SOTCA I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	30.95	54.06	76.00	158.59	238.19
U	40.00	40.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	120.00	120.00	120.00
Mean green time (s)	19.67	21.32	55.74	81.30	89.96
Max. waiting time (s)	249.00	443.00	776.00	2464.00	2777.00
U	40.00	40.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	120.00	120.00	120.00
Mean green time (s)	19.67	21.32	55.74	81.30	89.96
Total mean queue lengths	18.40	62.07	130.75	345.97	539.97
U	40.00	40.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	120.00	120.00	120.00
Mean green time (s)	19.67	21.32	55.74	81.30	89.96
Mean time in system (s)	77.58	100.45	126.95	218.58	297.98
U	40.00	40.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	120.00	120.00	120.00
Mean green time (s)	19.67	21.32	55.74	81.30	89.96
Max. time in system (s)	337.73	525.71	882.15	2621.95	2883.54
U	40.00	40.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	120.00	120.00	120.00
Mean green time (s)	19.67	21.32	55.74	81.30	89.96

Table B.8: First set of simulation results obtained by SOTCA I for a two-by-two grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

SOTCA II					
	λ				
	0.05	0.10	0.15	0.20	0.25
Mean waiting time (s)	26.29	42.54	75.96	164.63	239.78
U	40.00	60.00	60.00	100.00	120.00
$U^{\max}$	60.00	120.00	80.00	140.00	140.00
Mean green time (s)	17.19	20.53	35.34	81.47	107.98
Max. waiting time (s)	244.00	396.00	695.00	1802.00	3561.00
U	40.00	60.00	60.00	100.00	120.00
$U^{\max}$	60.00	140.00	80.00	120.00	140.00
Mean green time (s)	17.19	21.03	35.34	81.47	107.98
Total mean queue lengths	15.34	47.05	127.93	356.29	540.72
U	40.00	60.00	60.00	100.00	120.00
$U^{\max}$	60.00	120.00	80.00	140.00	140.00
Mean green time (s)	17.19	20.53	35.34	81.47	107.98
Mean time in system (s)	72.47	87.45	124.21	224.69	299.95
U	40.00	60.00	60.00	100.00	120.00
$U^{\max}$	60.00	120.00	80.00	140.00	140.00
Mean green time (s)	17.19	20.53	35.34	81.47	107.98
Max. time in system (s)	334.29	453.41	797.09	1889.91	2950.91
U	40.00	60.00	60.00	100.00	120.00
$U^{\max}$	60.00	140.00	80.00	120.00	140.00
Mean green time (s)	17.19	21.03	35.34	81.47	107.98

Table B.9: SFirst set of simulation results obtained by SOTCA II for a two-by-two grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OPS I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	52.04	60.61	406.71	508.62	522.97
Mean green time (s)	11.53	31.76	444.12	553.07	615.61
Max. waiting time (s)	409.00	478.00	6873.00	7298.00	8515.00
Mean green time (s)	12.58	31.76	444.12	553.07	615.61
Total mean queue lengths	32.79	69.97	567.12	681.07	808.68
Mean green time (s)	11.53	31.76	444.12	553.07	615.61
Mean time in system (s)	99.76	107.41	479.58	589.38	609.34
Mean green time (s)	11.53	31.76	444.12	553.07	615.61
Max. time in system (s)	478.16	586.40	7006.16	7396.65	8565.45
Mean green time (s)	12.58	31.76	444.12	553.07	615.61

Table B.10: First set of simulation results obtained by OPS I for a two-by-two grid of intersections with the resulting mean green time values.

OPS II					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	28.72	46.60	148.64	240.99	274.80
Mean green time (s)	23.34	18.07	17.92	17.39	22.60
Max. waiting time (s)	220.00	454.00	1191.00	1877.00	3624.00
Mean green time (s)	23.34	18.07	17.92	17.39	22.60
Total mean queue lengths	17.18	52.48	262.23	438.50	559.91
Mean green time (s)	23.34	18.07	17.92	17.39	22.60
Mean time in system (s)	75.24	91.41	211.83	302.91	330.72
Mean green time (s)	23.34	18.07	17.92	17.39	22.60
Max. time in system (s)	331.72	561.22	1224.02	2781.64	2995.38
Mean green time (s)	23.34	18.07	17.92	17.39	22.60

Table B.11: First set of simulation results obtained by OPS II for a single, isolated intersection with the resulting mean green time values.

SS					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	41.83	58.54	76.96	169.08	241.72
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.05	57.59	59.15	81.72	107.98
Max. waiting time (s)	406.00	710.00	1013.00	2012.00	4414.00
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.05	57.59	59.15	81.72	107.98
Total mean queue lengths	25.97	67.58	126.38	367.01	552.25
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.05	57.59	59.15	81.72	107.98
Mean time in system (s)	89.71	105.61	124.27	229.61	300.78
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.05	57.59	59.15	81.72	107.98
Max. time in system (s)	500.26	825.16	1170.12	2146.08	4537.47
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	52.05	57.59	59.15	81.72	107.98

Table B.12: First set of simulation results obtained by the SS for a single, isolated intersection with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OFTTCA					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	76.73	75.33	99.67	131.23	169.03
Green time (s)	50.00	50.00	50.00	50.00	50.00
Max. waiting time (s)	432.00	575.00	1027.00	3810.00	5784.00
Green time (s)	50.00	50.00	50.00	50.00	50.00
Total mean queue length	60.03	82.22	139.21	223.66	320.64
Green time (s)	50.00	50.00	50.00	50.00	50.00
Mean time in system (s)	127.15	123.54	148.30	181.75	221.43
Green time (s)	50.00	50.00	50.00	50.00	50.00
Max. time in system (s)	507.45	657.81	1125.95	3855.99	5861.36
Green time (s)	50.00	50.00	50.00	50.00	50.00

Table B.13: Second set of simulation results obtained by OFTTCA for a two-by-two grid of intersections with corresponding green time values.

SOTCA I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	55.10	55.55	65.71	92.37	128.52
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	11.77	16.72	22.19	26.59	30.41
Max. waiting time (s)	393.00	756.00	756.00	1070.00	1070.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	11.41	16.05	20.27	23.71	26.63
Total mean queue lengths	42.19	59.35	89.89	159.20	248.24
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	11.77	16.72	22.19	26.59	30.41
Mean time in system (s)	102.68	102.14	112.47	144.42	182.99
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	11.77	16.72	22.19	26.59	30.41
Max. time in system (s)	485.56	790.46	989.37	1000.13	1151.10
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	11.77	16.72	22.19	26.59	30.41

Table B.14: Second set of simulation results obtained by SOTCA I for a two-by-two grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

SOTCA II					
	λ				
	0.05	0.10	0.15	0.20	0.25
Mean waiting time (s)	27.79	38.73	60.24	89.12	124.86
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	120.00	120.00	120.00	120.00	120.00
Mean green time (s)	23.08	23.34	26.16	30.61	34.55
Max. waiting time (s)	195.00	472.00	646.00	946.00	1242.00
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	120.00	120.00	120.00	120.00	120.00
Mean green time (s)	23.08	23.34	26.16	30.61	34.55
Total mean queue lengths	19.46	39.88	82.52	152.60	239.35
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	120.00	120.00	120.00	120.00	120.00
Mean green time (s)	23.08	23.34	26.16	30.61	34.55
Mean time in system (s)	73.89	84.51	106.82	140.23	178.18
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	120.00	120.00	120.00	120.00	120.00
Mean green time (s)	23.08	23.34	26.16	30.61	34.55
Max. time in system (s)	308.61	559.01	714.58	1030.46	1337.41
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	120.00	120.00	120.00	120.00	120.00
Mean green time (s)	23.08	23.34	26.16	30.61	34.55

Table B.15: Second set of simulation results obtained by SOTCA II for a two-by-two grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OPS I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	57.28	58.25	150.54	258.67	317.62
Mean green time (s)	11.12	16.60	25.81	35.18	44.38
Max. waiting time (s)	409.00	508.00	2586.00	5343.00	6015.00
Mean green time (s)	11.12	16.60	25.81	35.18	44.38
Total mean queue lengths	43.37	59.70	213.95	368.49	469.10
Mean green time (s)	11.12	16.60	25.81	35.18	44.38
Mean time in system (s)	103.82	104.71	205.32	320.09	383.56
Mean green time (s)	11.12	16.60	25.81	35.18	44.38
Max. time in system (s)	437.00	611.83	2643.53	5416.49	6073.28
Mean green time (s)	11.12	16.60	25.81	35.18	44.38

Table B.16: Second set of simulation results obtained by OPS I for a two-by-two grid of intersections with the resulting mean green time values.

OPS II					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	28.73	39.44	72.79	127.40	167.69
Mean green time (s)	23.27	23.07	21.66	20.57	20.73
Max. waiting time (s)	221.00	382.00	731.00	1174.00	1174.00
Mean green time (s)	23.27	23.07	21.66	20.57	20.73
Total mean queue lengths	20.43	38.20	104.71	209.54	292.83
Mean green time (s)	23.27	23.07	21.66	20.57	20.73
Mean time in system (s)	74.74	85.75	122.89	181.45	222.10
Mean green time (s)	23.27	23.07	21.66	20.57	20.73
Max. time in system (s)	334.42	476.39	778.16	1229.85	1229.85
Mean green time (s)	23.27	23.07	21.66	20.57	20.73

Table B.17: Second set of simulation results obtained by OPS II for a two-by-two grid of intersections with the resulting mean green time values.

SS					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	65.72	66.50	74.93	103.84	138.17
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	140.00	140.00	140.00	140.00	140.00
Mean green time (s)	116.05	97.07	82.13	77.33	76.27
Max. waiting time (s)	309.00	427.00	854.00	1016.00	1099.00
U	40.00	40.00	40.00	40.00	40.00
$U^{\max}$	60.00	60.00	60.00	60.00	60.00
Mean green time (s)	55.69	42.96	37.26	35.81	35.55
Total mean queue lengths	50.39	71.80	103.00	174.18	262.16
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	140.00	140.00	140.00	140.00	140.00
Mean green time (s)	116.05	97.07	82.13	77.33	76.27
Mean time in system (s)	115.82	115.65	123.36	154.08	191.39
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	140.00	140.00	140.00	140.00	140.00
Mean green time (s)	116.05	97.07	82.13	77.33	76.27
Max. time in system (s)	642.67	642.67	980.77	1120.05	1200.65
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	140.00	140.00	140.00	140.00	140.00
Mean green time (s)	116.05	97.07	82.13	77.33	76.27

Table B.18: Second set of simulation results obtained by the SS for a a two-by-two grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

B.3 Results obtained for a three-by-three grid of intersections

The results presented in this section were obtained for a three-by-three grid of intersections. The results shown in Tables B.19, B.20, B.21, B.22, B.23 and B.24 were obtained for a three-by-three grid of intersections for the case in which the algorithm parameters were adjusted separately for each value of λ so that an optimal performance measure value could be attained. The results shown in Tables B.25, B.26, B.27, B.28, B.29 and B.30 were obtained for a three-by-three grid of intersections for the case in which the traffic control algorithm parameters remained fixed while the value of λ varied.

OFTTCA					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	84.36	106.95	192.77	303.54	487.46
Green time (s)	40.00	40.00	40.00	60.00	60.00
Max. waiting time (s)	692.00	936.00	1520.00	6711.00	7265.00
Green time (s)	50.00	60.00	50.00	60.00	60.00
Total mean queue length	81.48	226.17	531.63	938.22	1366.68
Green time (s)	40.00	40.00	40.00	60.00	60.00
Mean time in system (s)	142.32	182.64	267.06	357.35	527.77
Green time (s)	40.00	40.00	40.00	60.00	60.00
Max. time in system (s)	686.77	960.93	1574.68	5170.91	7138.50
Green time (s)	60.00	60.00	50.00	60.00	60.00

Table B.19: First set of simulation results obtained by OFTTCA for a three-by-three grid of intersections with corresponding green time values.

SOTCA I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	66.50	76.33	117.81	251.36	397.85
U	40.00	60.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	140.00	120.00	120.00
Mean green time (s)	10.73	25.86	56.38	81.09	89.90
Max. waiting time (s)	556.00	645.00	910.00	3108.00	6049.00
U	40.00	40.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	100.00	120.00	120.00
Mean green time (s)	10.73	21.29	54.22	81.09	89.90
Total mean queue lengths	64.25	135.32	295.75	794.73	1281.47
U	40.00	60.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	140.00	120.00	120.00
Mean green time (s)	10.73	25.86	56.38	81.09	89.90
Mean time in system (s)	126.07	134.00	175.59	315.03	451.41
U	40.00	60.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	140.00	120.00	120.00
Mean green time (s)	10.73	25.86	56.38	81.09	89.90
Max. time in system (s)	661.58	700.44	959.14	3189.74	6134.41
U	40.00	60.00	80.00	100.00	100.00
$U^{\max}$	60.00	80.00	100.00	120.00	120.00
Mean green time (s)	10.73	25.86	54.22	81.09	89.90

Table B.20: First set of simulation results obtained by SOTCA I for a three-by-three grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

SOTCA II					
	λ				
	0.05	0.10	0.15	0.20	0.25
Mean waiting time (s)	47.48	70.79	116.72	248.68	402.10
U	40.00	60.00	60.00	100.00	100.00
$U^{\max}$	60.00	100.00	120.00	120.00	120.00
Mean green time (s)	22.80	21.80	33.38	81.01	89.84
Max. waiting time (s)	473.00	652.00	837.00	2309.00	5441.00
U	40.00	40.00	60.00	100.00	100.00
$U^{\max}$	60.00	80.00	120.00	140.00	120.00
Mean green time (s)	22.80	20.59	33.38	80.55	89.84
Total mean queue lengths	44.53	124.89	297.88	789.72	1295.88
U	40.00	60.00	60.00	100.00	100.00
$U^{\max}$	60.00	100.00	120.00	120.00	120.00
Mean green time (s)	22.80	21.80	33.38	81.01	89.84
Mean time in system (s)	107.04	127.42	175.12	313.00	455.85
U	40.00	60.00	60.00	100.00	100.00
$U^{\max}$	60.00	100.00	120.00	120.00	120.00
Mean green time (s)	22.80	21.80	33.38	81.01	89.84
Max. time in system (s)	472.55	651.79	808.32	1902.11	5506.14
U	40.00	40.00	60.00	100.00	100.00
$U^{\max}$	60.00	80.00	120.00	120.00	120.00
Mean green time (s)	22.80	20.59	33.38	81.01	89.84

Table B.21: First set of simulation results obtained by SOTCA II for a three-by-three grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OPS I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	74.45	76.91	647.96	868.99	950.19
Mean green time (s)	10.99	28.60	403.80	517.21	587.87
Max. waiting time (s)	691.00	653.00	14278.00	20522.00	22050.00
Mean green time (s)	10.99	28.60	403.80	517.21	587.87
Total mean queue lengths	73.25	136.78	1335.51	1780.30	1995.33
Mean green time (s)	10.99	28.60	403.80	517.21	587.87
Mean time in system (s)	134.11	135.03	723.89	948.62	1035.40
Mean green time (s)	10.99	28.60	403.80	517.21	587.87
Max. time in system (s)	651.39	770.12	9074.52	19816.38	21017.93
Mean green time (s)	10.99	28.60	403.80	517.21	587.87

Table B.22: First set of simulation results obtained by OPS I for a three-by-three grid of intersections with the resulting mean green time values.

OPS II					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	47.57	72.10	194.83	422.73	459.98
Mean green time (s)	27.87	17.20	16.12	12.23	25.73
Max. waiting time (s)	477.00	660.00	2006.00	4227.00	11154.00
Mean green time (s)	27.87	17.20	16.12	12.23	25.73
Total mean queue lengths	45.80	125.32	494.82	1057.63	1322.56
Mean green time (s)	27.87	17.20	16.12	12.23	25.73
Mean time in system (s)	107.75	127.52	254.76	468.22	501.49
Mean green time (s)	27.87	17.20	16.12	12.23	25.73
Max. time in system (s)	492.13	641.87	2018.85	4274.90	11166.09
Mean green time (s)	27.87	17.20	16.12	12.23	25.73

Table B.23: First set of simulation results obtained by OPS II for a three-by-three grid of intersections with the resulting mean green time values.

SS					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	69.47	82.51	117.52	256.26	444.77
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	51.75	54.86	57.77	81.30	107.80
Max. waiting time (s)	536.00	652.00	1054.00	2138.00	8065.00
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	51.75	54.86	57.77	81.30	107.80
Total mean queue lengths	66.18	148.62	295.94	807.39	1434.81
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	51.75	54.86	57.77	81.30	107.80
Mean time in system (s)	129.65	141.59	175.91	319.26	495.99
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	51.75	54.86	57.77	81.30	107.80
Max. time in system (s)	622.36	739.14	1066.19	2048.75	7801.67
U	40.00	60.00	80.00	100.00	120.00
$U^{\max}$	60.00	80.00	100.00	120.00	140.00
Mean green time (s)	51.75	54.86	57.77	81.30	107.80

Table B.24: First set of simulation results obtained by the SS for a three-by-three grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OFTTCA					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	106.33	119.10	135.89	177.37	232.15
Green time (s)	60.00	60.00	60.00	60.00	60.00
Max. waiting time (s)	669.00	785.00	3115.00	3115.00	4468.00
Green time (s)	60.00	60.00	60.00	60.00	60.00
Total mean queue length	127.90	200.49	285.84	446.94	658.29
Green time (s)	60.00	60.00	60.00	60.00	60.00
Mean time in system (s)	167.11	178.56	193.99	234.28	288.33
Green time (s)	60.00	60.00	60.00	60.00	60.00
Max. time in system (s)	666.52	839.88	1304.62	3020.07	3672.90
Green time (s)	60.00	60.00	60.00	60.00	60.00

Table B.25: Second set of simulation results obtained by OFTTCA for a three-by-three grid of intersections with corresponding green time values.

SOTCA I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	69.89	76.64	94.19	133.96	186.09
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	10.70	15.86	21.20	25.55	29.28
Max. waiting time (s)	606.00	622.00	782.00	1071.00	1535.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	10.81	15.37	19.66	23.16	26.02
Total mean queue lengths	86.41	124.31	197.09	342.88	527.62
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	10.81	15.37	19.66	23.16	26.02
Mean time in system (s)	131.56	133.59	151.27	192.78	244.47
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	10.81	15.37	19.66	23.16	26.02
Max. time in system (s)	689.85	689.85	804.75	1098.01	1337.86
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	10.81	15.37	19.66	23.16	26.02

Table B.26: Second set of simulation results obtained by SOTCA I for a three-by-three grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

SOTCA II					
	λ				
	0.05	0.10	0.15	0.20	0.25
Mean waiting time (s)	44.32	60.38	87.31	124.99	178.19
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	26.97	25.21	27.86	31.07	33.73
Max. waiting time (s)	331.00	558.00	830.00	963.00	1814.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	26.97	25.21	27.86	31.07	33.73
Total mean queue lengths	50.31	96.90	183.83	320.43	502.64
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	26.97	25.21	27.86	31.07	33.73
Mean time in system (s)	103.58	117.77	144.87	184.19	235.40
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	26.97	25.21	27.86	31.07	33.73
Max. time in system (s)	406.68	553.38	780.74	972.57	1485.24
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	26.97	25.21	27.86	31.07	33.73

Table B.27: Second set of simulation results obtained by SOTCA II for a three-by-three grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OPS I					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	72.37	76.56	223.44	406.52	518.48
Mean green time (s)	10.98	16.39	25.24	34.41	43.57
Max. waiting time (s)	465.00	642.00	4116.00	8149.00	13535.00
Mean green time (s)	10.98	16.39	25.24	34.41	43.57
Total mean queue lengths	87.07	126.30	501.42	897.37	1181.75
Mean green time (s)	10.98	16.39	25.24	34.41	43.57
Mean time in system (s)	131.98	135.44	291.27	477.69	594.30
Mean green time (s)	10.98	16.39	25.24	34.41	43.57
Max. time in system (s)	580.21	684.95	4046.70	8256.79	12496.00
Mean green time (s)	10.98	16.39	25.24	34.41	43.57

Table B.28: Second set of simulation results obtained by OPS I for a three-by-three grid of intersections with the resulting mean green time values.

OPS II					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	45.33	59.74	106.03	177.21	236.71
Mean green time (s)	26.29	24.04	20.56	19.31	19.99
Max. waiting time (s)	326.00	503.00	3398.00	3398.00	3398.00
Mean green time (s)	26.29	24.04	20.56	19.31	19.99
Total mean queue lengths	51.06	95.78	225.63	430.21	614.98
Mean green time (s)	26.29	24.04	20.56	19.31	19.99
Mean time in system (s)	104.42	117.00	162.25	232.39	287.47
Mean green time (s)	26.29	24.04	20.56	19.31	19.99
Max. time in system (s)	419.24	526.32	3418.23	3418.23	3418.23
Mean green time (s)	26.29	24.04	20.56	19.31	19.99

Table B.29: Second set of simulation results obtained by OPS II for a three-by-three grid of intersections with the resulting mean green time values.

SS					
	λ				
	0.05	0.1	0.15	0.2	0.25
Mean waiting time (s)	85.67	85.36	96.06	136.26	189.13
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	91.74	69.71	59.07	56.27	55.54
Max. waiting time (s)	763.00	763.00	763.00	1112.00	1615.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	91.74	69.71	59.07	56.27	55.54
Total mean queue lengths	103.80	142.70	201.36	346.15	531.20
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	91.74	69.71	59.07	56.27	55.54
Mean time in system (s)	148.24	145.88	154.30	195.07	246.06
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	91.74	69.71	59.07	56.27	55.54
Max. time in system (s)	809.32	809.32	809.32	1165.09	1665.67
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	91.74	69.71	59.07	56.27	55.54

Table B.30: Second set of simulation results obtained by the SS for a three-by-three grid of intersections with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

B.4 The Adam Tas Road & Bird Street intersection

The results shown in Tables B.31, B.32, B.33, B.34, B.35 and B.36 were obtained for the Adam Tas Road & Bird Street intersection in Stellenbosch, South Africa, for the case in which the traffic control algorithm parameters remained fixed for the duration of each simulation run. The data provided in Tables A.1, A.2, A.3 and A.4 in Appendix A were used as input to the model in order to generate the vehicle arrival distributions for each intersection approach.

CIVACR					
	λ				
	09:00	11:30	14:00	16:30	18:00
Mean waiting time (s)	121.50	93.96	83.68	76.43	84.18
Phase 1 green time (s)	10	12.38	12.38	8.11	8.11
Phase 2 green time (s)	55.14	46.75	46.75	52.84	52.84
Phase 3 green time (s)	55.79	68.88	68.88	70.74	70.74
Phase 4 green time (s)	14.71	9.75	9.75	3.79	3.79
Max. waiting time (s)	390.00	390.00	390.00	390.00	390.00
Phase 1 green time (s)	10	12.38	12.38	8.11	8.11
Phase 2 green time (s)	55.14	46.75	46.75	52.84	52.84
Phase 3 green time (s)	55.79	68.88	68.88	70.74	70.74
Phase 4 green time (s)	14.71	9.75	9.75	3.79	3.79
Total mean queue length	115.93	70.86	60.36	53.74	60.81
Phase 1 green time (s)	10	12.38	12.38	8.11	8.11
Phase 2 green time (s)	55.14	46.75	46.75	52.84	52.84
Phase 3 green time (s)	55.79	68.88	68.88	70.74	70.74
Phase 4 green time (s)	14.71	9.75	9.75	3.79	3.79
Mean time in system (s)	142.05	117.75	108.68	101.99	108.95
Phase 1 green time (s)	10	12.38	12.38	8.11	8.11
Phase 2 green time (s)	55.14	46.75	46.75	52.84	52.84
Phase 3 green time (s)	55.79	68.88	68.88	70.74	70.74
Phase 4 green time (s)	14.71	9.75	9.75	3.79	3.79
Max. time in system (s)	493.00	493.00	493.00	493.00	493.00
Phase 1 green time (s)	10	12.38	12.38	8.11	8.11
Phase 2 green time (s)	55.14	46.75	46.75	52.84	52.84
Phase 3 green time (s)	55.79	68.88	68.88	70.74	70.74
Phase 4 green time (s)	14.71	9.75	9.75	3.79	3.79

Table B.31: Simulation results obtained by CIVACR with corresponding green time values.

SOTCA I					
	λ				
	09:00	11:30	14:00	16:30	18:00
Mean waiting time (s)	102.06	94.79	82.65	84.69	90.10
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	24.00	23.08	23.28	23.87	24.91
Max. waiting time (s)	460.00	460.00	460.00	460.00	460.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	24.00	23.08	23.28	23.87	24.91
Total mean queue lengths	113.65	70.87	61.09	61.31	68.07
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	24.00	23.08	23.28	23.87	24.91
Mean time in system (s)	121.61	117.48	106.50	108.84	113.34
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	24.00	23.08	23.28	23.87	24.91
Max. time in system (s)	556.14	556.14	556.14	556.14	556.14
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	24.00	23.08	23.28	23.87	24.91

Table B.32: Simulation results obtained by SOTCA I with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

SOTCA II					
	λ				
	09:00	11:30	14:00	16:30	18:00
Mean waiting time (s)	104.55	82.33	67.67	67.34	72.65
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	36.74	33.24	32.48	32.42	33.06
Max. waiting time (s)	470.00	470.00	470.00	470.00	470.00
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	36.74	33.24	32.48	32.42	33.06
Total mean queue lengths	114.13	61.44	49.02	46.82	55.41
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	36.74	33.24	32.48	32.42	33.06
Mean time in system (s)	124.57	105.51	91.88	91.88	96.29
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	36.74	33.24	32.48	32.42	33.06
Max. time in system (s)	574.11	574.11	574.11	574.11	574.11
U	80.00	80.00	80.00	80.00	80.00
$U^{\max}$	100.00	100.00	100.00	100.00	100.00
Mean green time (s)	36.74	33.24	32.48	32.42	33.06

Table B.33: Simulation results obtained by SOTCA II with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

OPS I					
	λ				
	09:00	11:30	14:00	16:30	18:00
Mean waiting time (s)	126.73	109.81	96.07	92.52	101.20
Mean green time (s)	67.16	35.27	32.03	30.33	34.61
Max. waiting time (s)	509.00	509.00	509.00	509.00	509.00
Mean green time (s)	67.16	35.27	32.03	30.33	34.61
Total mean queue lengths	140.30	82.64	70.77	67.73	80.52
Mean green time (s)	67.16	35.27	32.03	30.33	34.61
Mean time in system (s)	149.01	134.04	120.90	117.62	125.79
Mean green time (s)	67.16	35.27	32.03	30.33	34.61
Max. time in system (s)	613.90	613.90	613.90	613.90	613.90
Mean green time (s)	67.16	35.27	32.03	30.33	34.61

Table B.34: Simulation results obtained by OPS I with the resulting mean green time values.

OPS II					
	λ				
	09:00	11:30	14:00	16:30	18:00
Mean waiting time (s)	122.01	85.48	74.23	71.54	76.45
Mean green time (s)	25.89	30.90	30.23	29.46	28.47
Max. waiting time (s)	505.00	505.00	505.00	505.00	505.00
Mean green time (s)	25.89	30.90	30.23	29.46	28.47
Total mean queue lengths	111.83	62.36	52.06	49.03	53.88
Mean green time (s)	25.89	30.90	30.23	29.46	28.47
Mean time in system (s)	140.71	107.67	97.26	94.77	98.66
Mean green time (s)	25.89	30.90	30.23	29.46	28.47
Max. time in system (s)	599.16	599.16	599.16	599.16	599.16
Mean green time (s)	25.89	30.90	30.23	29.46	28.47

Table B.35: Simulation results obtained by OPS II with the resulting mean green time values.

SS					
	λ				
	09:00	11:30	14:00	16:30	18:00
Mean waiting time (s)	105.56	96.02	84.65	85.06	90.94
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	35.02	43.18	44.74	44.24	42.08
Max. waiting time (s)	482.00	482.00	482.00	482.00	482.00
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	35.02	43.18	44.74	44.24	42.08
Total mean queue lengths	114.65	74.25	64.49	62.55	69.57
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	35.02	43.18	44.74	44.24	42.08
Mean time in system (s)	125.90	119.42	109.04	109.84	114.80
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	35.02	43.18	44.74	44.24	42.08
Max. time in system (s)	581.05	581.05	581.05	581.05	581.05
U	60.00	60.00	60.00	60.00	60.00
$U^{\max}$	80.00	80.00	80.00	80.00	80.00
Mean green time (s)	35.02	43.18	44.74	44.24	42.08

Table B.36: Simulation results obtained by the SS with corresponding values of U and $U^{\max}$ and the resulting mean green time values.

APPENDIX C

Contents of the accompanying compact disk

This appendix contains a brief description of the compact disc included with this thesis. The compact disk contains an electronic version of the thesis itself in ".pdf" format, the AnyLogic code associated with the model used to simulate the single, isolated intersection of §5.3.1 in ".html" format, and executable versions of the traffic simulation model for a single, isolated intersection, a two-by-two grid of intersections and a three-by-three grid of intersections. The traffic simulation model was created in AnyLogic version 6.5.0 and may be executed in this version or later. There are three directories on the compact disc and their contents are described here by their directory names.

Thesis. This directory contains an electronic copy of this thesis in ".pdf" format.

Code. This directory contains each piece of implementation code associated with the traffic simulation model used to simulate the single, isolated intersection of §5.3.1 in ".html" code.

Traffic simulation model. This directory contains the traffic simulation model for a single intersection, a two-by-two grid of intersections and a three-by-three grid of intersections as AnyLogic script files (".alp" format). The script for the single intersection, the two-by-two grid of intersections and the three-by-three grid of intersections is labelled "SI.alp", "2x2.alp" and "3x3.alp", respectively. To execute any of these traffic simulation models, the relevant file should be opened from AnyLogic. Once opened in AnyLogic the user is required to select the desired model in the "Projects" window in AnyLogic. The user should then build the model, either by clicking the "Build model" button in AnyLogic or by pressing F7. Once the model has compiled successfully, the user is required to run the model either by clicking the "Run" button in AnyLogic, or by pressing F5. Following this, a window will appear with the title of the simulation being run, and a button which reads "Run the model and switch to main view". The user is required to click this button at which time the execution of the simulation will commence. While the simulation is running the user may alter the arrival rates of vehicles to the system using the slider labelled "Arrival rate," as well as select the traffic control algorithm to be implemented and their associated parameters using the appropriate check boxes.