

Tuning Feedback-Based Traffic Signal Controls

Simon Paulsson



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6067
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2018 by Simon Paulsson. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2018

Abstract

A growing problem in many cities is that the traffic demand fluctuates heavily during the day and that it grows larger as the years go by. This calls for, and the technology to easier measure traffic enables, more efficient traffic light controls that can handle a greater traffic demand but also be adaptive. Centralised control of traffic light might, even though it can be both adaptive and efficient, be costly to install and expand due to the required communication between junctions. Decentralised control however, might be cheaper to install and maintain while still being adaptive and efficient. This thesis explores the Proportional Allocation controller with regards to its tuning parameter under conditions imposed by e.g., network size and topology, and queue sensor length. Two aspects of its performance were measured: optimality and fairness. In addition to the original controller two modifications are proposed and evaluated. It was found that when simulating, smaller networks could be used to approximate bigger ones, and that the queue sensor length is of critical importance not to be too small. The proposed modifications were found to perform similarly to the original one even though it in principal should not. The results herein should give a good starting point of the optimal value of the tuning parameter in a real-world implementation.

Acknowledgements

I would like to thank my supervisors Associate Professor Giacomo Como and Ph.D. student Gustav Nilsson for guiding me during the work on this thesis. I would also like to thank Christine Sjölander for being a ball plank, and Ph.D. Nils Paulsson for proofreading and helping out with the report.

Contents

1. Introduction	9
2. Models of Traffic Networks and Controllers of Traffic Signals	13
2.1 Mathematical theory	13
2.2 Macro and Micro Simulations	18
2.3 Proportional allocation controller	21
2.4 Back-pressure control	23
3. Tuning of the Proportional Allocation Controller	24
3.1 Optimality	24
3.2 Fairness	30
4. Required sensor length	36
4.1 Two analytic approaches	36
4.2 Simulations in MATLAB	40
4.3 Simulations in SUMO	42
5. Performance comparison with Back-pressure control	46
6. Discussion	52
7. Future work	55
Bibliography	56
A. Simulation scenarios	58
A.1 Phase setup	58
A.2 Traffic generation	59
B. Optimality results	62
C. Fairness results	71
D. Stability results	78
E. Performance	80

1

Introduction

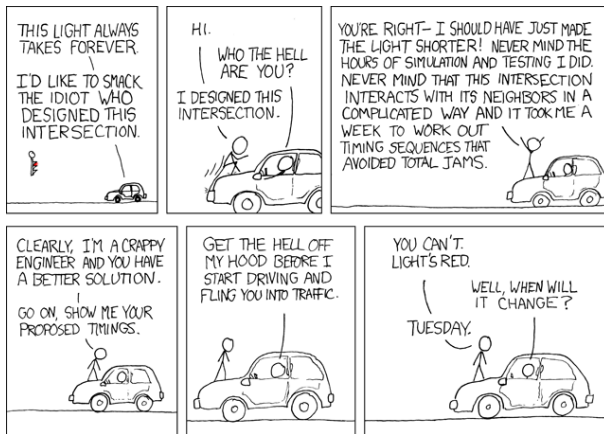


Figure 1.1 Image from [*Long light*]. Randall Munroe, *Long light*.

On the 9th of December 1868 the first traffic light in the world was installed. It was situated just outside the Houses of Parliament in London and was a gas-lit contraption showing green light in one direction and red in the other. It soon exploded, injuring the policeman who operated it. However, during its brief lifetime it is said that it successfully controlled the traffic and lowered the congestion. The idea was fruitful and just after the turn of the century, traffic signal systems like the one in London were in place all over the United States of America. As time went by, with the help of electric lights, the traffic signals in the same junction were made to function from a central control box. Then several junctions were put together to be controlled from the same control system. After the second world war, with the rise of computers, the traffic lights started to be controlled by computers together with a pressure plate sensor, to see if there was a queue waiting at a red light. For example,

in Denver 1952, a computer controlled 120 lights with a total of 6 pressure plates as the input signal. [Wikipedia, 2018]

As traffic demand, computer power and interconnectedness has increased, *centralised* control has been implemented. Centralised control is when a control unit controls the traffic lights in several junctions simultaneously. This enables inventions such as green waves, which is when a series of traffic lights, are coordinated to allow continuous traffic flow over several junctions in one main direction.

The problem with centralised control is that the infrastructure to implement it can be rather resource demanding because communication systems need to be set in place. In recent years therefore, with more sophisticated sensors, and research done in queue theory in computer networks, research has been done on *decentralised* controllers. This means that the traffic light control system in each junction only has information of that junction.

Most notably among the proposed decentralised controllers is the Max pressure controller by [Varaiya, 2013] that is an adaptation of earlier works for communications networks in [Tassiulas and Ephremides, 1992]. However, this controller needs information about the traffic turning ratios beforehand, i.e., what fraction of vehicles that are going where. This type of controller is also called a Back-pressure controller. In [Gregoire et al., 2014] it has been adapted to function without prior known turning ratios. However, this adaptation is no longer throughput optimal. Simulations done by the authors of that article shows that the loss of optimality is around 20%.

The main problem of traffic junctions as pointed out in [Roess et al., 2011] is that vehicles have to accelerate. This means that when a queue gets a green light it takes some time to get up to full speed. The consequence of this is that each time the lights switches, because of non-optimal flow in the beginning, some time that could be used to let cars drive, is lost. This is called lost time. If drawn to the extreme: if the lights are switched very frequently (a very short cycle length), i.e. disco-lights, no vehicle has enough time to drive through the junction. On the other hand, if the lights are switched every other day (a very long cycle length), in some sense it is no longer a traffic light. Somewhere in between is the optimum between not having too long to wait for green light and having enough time to drive, while still maintaining traffic capacity.

The Max pressure/Back pressure controller does take this into consideration. However, only with a static cycle time. In the last few years work has been done to try to set the cycle length of the traffic lights in a decentralised control adaptively, with the queue lengths as inputs, so as to keep optimal control even with varying traffic demand. Higher traffic demand requires a longer cycle time because of the aforementioned time to switch lights. In [Kovacs et al., 2016] it is proposed to change the cycle length dynamically and to set it to be proportional to the square root of the sum of queues. However, the intent in that paper, is to change the cycle times in an iterative learning control manner, using the traffic from previous days to determine the cycle times of the upcoming day. This means that there is an as-

sumption of traffic being very similar day to day, and that the adaptiveness is slow. In [Nilsson and Como, 2018] the proposition is a cycle time that is proportional to the sum of the queue lengths and inversely to a tuning parameter, which will be more clarified later in Chapter 2.2. For now, it can be said that this controller works independently of previous information in time.

All of the previously mentioned control strategies have been formulated and explored in a highly theoretical way, with simplistic models to be able to prove stability and to be able to understand them better. Some of them have been simulated, where [Nilsson and Como, 2018] has gone the furthest. In that paper simulations have been done with the proportional allocation controller with dynamic cycle lengths in a microscopic traffic simulator, SUMO.

Aim of thesis

The main problem that this thesis tries to answer is how to *tune* the Proportional Allocation controller. This, in essence, is an optimisation problem - to minimise or maximise something. Here the main points of interest are to minimise the amount of traffic in the network, in this thesis called optimality, and to do this in a fair way. Both of these points will be presented later in Chapter 3. However, to achieve optimality in a certain city might require different settings than in another city. The main interest then becomes to see how different operation conditions affect the tuning of the controller. Two modifications to the controller are also suggested and the same sort of results are produced for these. Performance comparison is also carried out to a smaller extent. The different areas that are investigated are summarised in the following list:

- Controller modifications
- Network particulars
 - Size
 - Topology
 - Traffic demand
- Sensor length
- Performance

Outline of this Master's thesis

This thesis first goes on to explain the traffic models and controllers along with modifications in Chapter 2. In Chapter 3 the "Network particulars" in the list are investigated as well as some introductory investigations into the affects of the sensor length. The sensor length, and its effects on stability and tuning, is more closely examined in the next chapter, Chapter 4. Lastly, a comparison between the proportional allocation controller and the Back-pressure controller is carried out in Chapter 5.

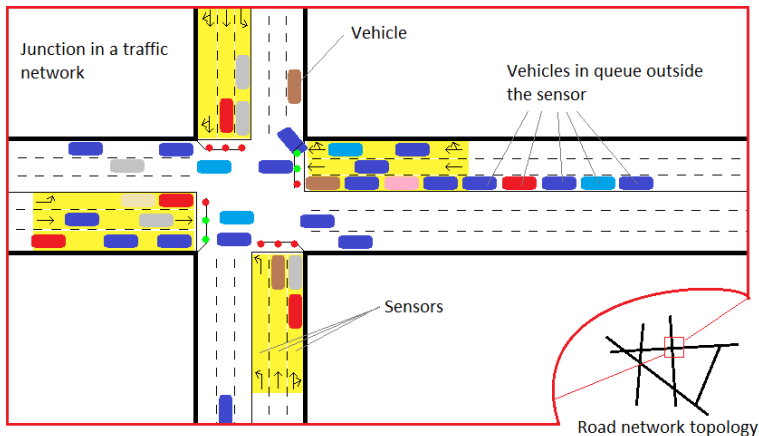


Figure 1.2 An illustration of the problem setting. This is a junction in a bigger network of roads that are connected in a certain way. In the picture the lanes that allow horizontal vehicles to drive straight and to turn right receive green light. Sensors are used to detect queues, however, they may be limited in length and therefore might not be measuring the whole queue.

Summary of introduction

To summarise: the problem at junctions is to control the traffic flow via traffic lights to achieve better traffic flow. Reference Figure 1.2 for an image. This is not easy because vehicles can collide and need to accelerate. Collision possibilities means that not all vehicles can drive at the same time. Therefore, the lights need to switch between green, yellow and red. Because of the need for acceleration, the switching incurs overhead switching times. The need to switch also begs the question of how long time to receive green light. This is determined by some controllers by having sensors in the road to detects vehicles. However, the sensors have a limited length. When a control strategy is decided the following problem is to tune it, if necessary. This is not easy either since it can depend a great deal on a variety of things, e.g. network topology and network size.

2

Models of Traffic Networks and Controllers of Traffic Signals

In this chapter first a theoretical understanding of general traffic models will be explained, then the macroscopic model used in MATLAB simulations will follow. Lastly, the microscopic simulator SUMO and its associated tools will be described.

2.1 Mathematical theory

Traffic networks as graphs

Road networks can be described with mathematical graph theory and the mathematical tools therein. Where different roads meet, i.e. an intersection or a junction, is a *node*. Here the term *junction* will be used to mean the physical junction (or a simulated one) and node will be used where a more mathematical language is needed. Roads can consist of one or more *lanes*. The lanes will be represented in a mathematical language by *edges*. Edges connect nodes. Edges are for the purpose of this thesis *directional*, so the vehicles on a road can only go one way. To accommodate for travelling both ways between nodes, and multiple lane roads, more than one edge needs to be in place. We define the *network topology* as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where \mathcal{V} denotes the set of nodes (\mathcal{V} stands for vertices) and \mathcal{E} denotes the set of edges connecting these nodes.

Nodes as junctions

For every node (or junction) $v \in \mathcal{V}$ we can define the lanes where vehicles enters it as input lanes $i \in I_v$ and the lanes where traffic is leaving it as output lanes $m \in O_v$. A junction may be traffic light controlled and in the context of this thesis they always are. Every lane has a traffic light that can allow the traffic from that lane to drive or not.

Continuous model of flow

The states in the model used in this thesis consist of the queue lengths on every lane. The queues are measured in number of vehicles. They are denoted $x_i(t)$ for every lane i at time t , and $x(t)$ is used to denote the vector containing all the queues at time t . To model how the vehicles move through the network on a statistical level, the *routing matrix* $\mathbf{R} \in \mathbb{R}^{\mathcal{E} \times \mathcal{E}}$ is created. The entries R_{ij} in the matrix denotes what fraction of incoming traffic on lane i that goes to lane j . These are called *turning ratios*. For example: if incoming traffic from lane turns right 60% of the time, goes straight 40% of the time and left 0% of the time, the turning ratios would be (left, straight, right) = (0, 0.4, 0.6). With these turning ratios on all the incoming lanes in the junction in Figure 2.1, the resulting routing matrix would have the following values, given that the junction is the whole network:

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 0 & 0.6 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.1)$$

By necessity of mass conservation, so that the routing matrix does not add vehicles to the network, the sum of every row has to be smaller or equal to 1. Furthermore every entry naturally has to be 0 or larger. A value smaller than 0 would mean that vehicles are flowing backwards. An entry R_{ij} can only be larger than 0 if there is an edge from i to j .

Along with the turning ratios, there may also be exogenous inflow to a subset of the lanes. Exogenous inflow is flow that comes from outside the network. This is denoted λ_i for a lane i . There is also outflow from the lane that goes to other lanes according to the aforementioned turning ratios. Flows are denoted z_i for lane i . The full balance equation for a given lane i becomes:

$$\dot{x}_i = \lambda_i + \sum_j R_{ji} z_j - z_i. \quad (2.2)$$

The change in the queue is equal to the exogenous flow plus the inflow from other lanes minus the outflow from the lane itself. The outflow is limited because of physical constraints, mostly by the *saturated flow rate* which is a number indicating the maximum flow on a lane in veh/s (vehicles per second). This is here denoted s_i and by necessity $z_i \leq s_i$. Moreover, the challenge for a junction with traffic lights is to give a control signal $u_i \in [0, 1]$ that determines how large the fraction of time is used to *serve* (give green light to) a queue x_i such that the queues are kept *stable*. With the control action the outflow z_i is limited so that $z_i \leq u_i s_i$.

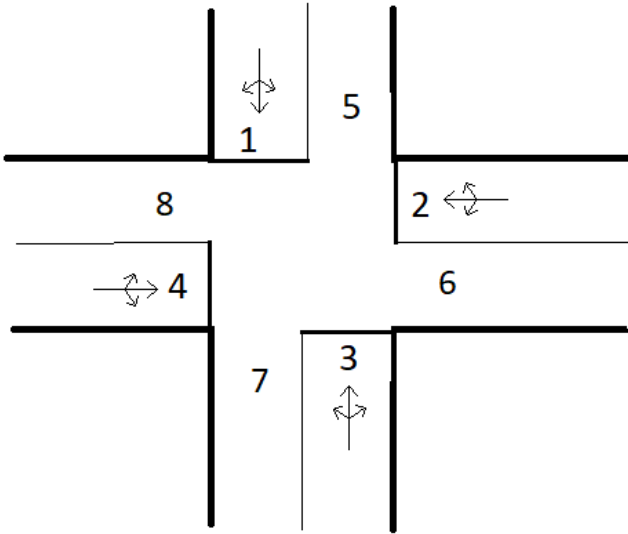


Figure 2.1 A junction that is helpful when trying to understand the turning ratios in (2.1).

In the context of this thesis a queue is said to be *stable* if the length of the queue is bounded. And since the traffic cannot be negative this also means that the limit

$$\lim_{t \rightarrow \infty} x_i(t) \quad (2.3)$$

is bounded for every i . This, in turn means, that $\int_0^\infty \dot{x}_i(t) \leq \varepsilon$ for an arbitrarily big value of ε and in turn that

$$\int_0^\infty (\lambda_i + \sum_j R_{ji} z_j) dt \leq \int_0^\infty z_i dt + \varepsilon \quad (2.4)$$

which in words can be expressed as: the inflow over time must be less than or equal to the outflow over time to keep the queue stable.

Introduction to control of Traffic lights

By necessity, to avoid collisions and so to make traffic light controlled junctions a viable option instead of having a junction without traffic lights, the control signal needs to be discretized so that the control can be divided into phases. A *phase* consists of one or more lanes that receive green light simultaneously. Usually a

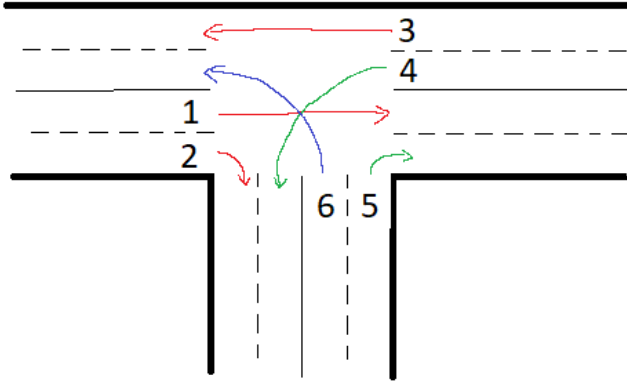


Figure 2.2 A junction with phases to reference when trying to understand the phase matrix construction. Lanes are numbered for reference.

phase matrix is constructed so that the problem can be understood and used in a more mathematical way. A phase matrix $\mathbf{P}^{(v)}$ is a binary $p \times l$ matrix where p is the number of possible and l is the number of lanes. Every junction has its own phase matrix associated with it as indicated by the index v . The phase matrix may be unique for every junction depending on what lanes are included in what phase, but in this thesis the phase matrix is the same for every junction in the whole network and might only change when the network topology changes. Every row in the phase matrix indicates a phase. Every lane l that is executed during phase i has a 1 on row i and column l . All other entries are zero. So the single junction in figure 2.2 would have the phase matrix

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (2.5)$$

indicating that the red phase allowing vehicles from lanes 1, 3, and 4 to drive is the first phase, followed by blue and then green. If every lane l belongs to one and only one phase, the phases are said to be *orthogonal*, which they are in this example.

The main challenge of every controller is to choose the desired phase at every time. This affects if a lane can actually empty. The dynamical flow equation 2.2 now has to be rewritten as

$$\dot{x}_i(t) = \lambda_i + \sum_j R_{ji} z_j - u_i s_i \quad (2.6)$$

where $f_i(u, x)$ is the flow from lane i but now depending on the queues and control signal. Moreover, the assumption is that the outflow from a lane is at the maximum when vehicles are present:

$$x_i(t) > 0 \Rightarrow z_i = u_i s_i. \quad (2.7)$$

Usually, the control strategy is decided at discrete time intervals. This interval is called a cycle. In classical fixed control strategy this cycle time is fixed and all the phases have to be executed in it, see [Roess et al., 2011]. It can also be dynamically set as done in [Nilsson and Como, 2018] such that the next point of control is decided upon from the queues as

$$t_{k+1}^{(v)} = t_k^{(v)} + T(x^{(v)}(t_k)) \quad (2.8)$$

where $t_k^{(v)}$ are the points in time at a certain junction (indicated by the indices v) when the control action up to the next point is calculated, and $T(x^{(v)}(t_k))$ is a function of the queues that determines how long the upcoming cycle length will be. Now the full problem formulation can be described as trying to achieve stability with control under the influence of discretization. Different control strategies will be more explained in the next chapter, for now it will suffice to say that discretization of the control (and the real world) bears with it artefacts (start-up and clearance lost time) that needs to be accounted for in the control strategies.

Modelling of junction lost time

The fundamental problems of traffic at junctions are that vehicles need to accelerate and that they can collide. The collision issue is what calls for traffic lights in the first place. When a queue is waiting for the green light the vehicles in it stands still. After some reaction time when the light turns to green, the first vehicle starts to accelerate and pass the stop line to the junction. After some time again the second vehicle passes. This time between the vehicles is called the *headway*. The first headway is in relation to the rest long, due to reaction time and acceleration. The second headway is shorter because the vehicle has had time before the junction to accelerate. The headways are progressively shorter until a certain time called *saturated flow headway* is reached when the queue has come up to speed. Here denoted h , with unit s/veh since it measures the time between vehicles (how long time it takes for a vehicle to pass), it is the minimum average time between vehicles in a stable moving queue that can be achieved. See [Roess et al., 2011, Figure 20-1]. If the extra time used to get the queue up to speed is taken away, the queue would move at the saturated headway all the time from the beginning of receiving green light. This extra time is called the *start-up lost time* and is here denoted l_1 . The problem of acceleration also works in the opposite direction when the phase is about to switch. This means that yellow lights have been introduced so that the vehicles have a fair chance of stopping when the light turns to red.

To stop vehicles from colliding in the junction, beside the necessity of having the control action working in phases, the traffic lights needs to have all red time to allow vehicles to safely leave the junction. All red means that all the lights into the junction show red light. This is more time lost that could have been used to allow a queue to drive. This is called *clearance lost time* and is denoted l_2 or T_w . The cycle lost time L_{tot} is the total start-up lost time and clearance loss over the entire cycle.

Cycle times and capacity

Every time a phase switch occurs time that could have been used to serve a queue is then lost. Since a cycle is made up of one or more phase switches, in every cycle, time is lost. This is important to know as it heavily affects the *capacity* of the junction, the volume of traffic that the junction can process. If the cycle time is very short so that the vehicles have less time than the start-up lost time to move, not many vehicles can pass the junction. The capacity is very low. On the other hand, if the cycle time is made very long it has very high capacity, but a vehicle might have to wait too long of a time to pass it. Somewhere in between is the cycle time that is just enough to process the traffic demand but not so long so that the vehicles have to wait longer than necessary. Traffic demand can loosely be defined as the volume of traffic that the network, road, or junction needs to handle. In other words, the number of cars that have a route through the defined time period and place.

Critical lane volume

The *critical lane volume* is used as a measurement of traffic demands. It is the measurement of the lane in a phase that has the most traffic demand. The unit is veh/s. This measurement is really related to how much time is needed to process the traffic in that lane, so modifiers need to be added for e.g. left-turning vehicles that take longer time to flow through the junction, uphill queues and downhill queues. The measurement that is used to define total traffic demand is called the *sum of critical lane volume*, which is rather obviously the sum of the critical lane volumes over all the phases in a junction. It is denoted V_c . The critical lane volume differs from the capacity in that it does only count the lane in each phase that has the highest traffic demand.

2.2 Macro and Micro Simulations

When simulating, in essence two approaches can be made. Macro, where traffic is modelled as a continuous mass, where only the traffic volume is known, and Micro simulation where every vehicle is simulated individually, perhaps with some dynamic to mimic driver behaviour. The reason to use a macro model is that it is almost necessary to be able to do anything theoretical. It is also a lot less computationally heavy than a microscopic simulation since every vehicle does not need simulating. The downside is that when using a simpler model, it is simpler and therefore the conclusions that can be drawn from the results of the simulations are fewer. However, since a microscopic model might to some extent model driver behaviour and have more built in dynamics, even if the conclusions from the micro simulator might be greater in number they may be of lower certainty and more care must be taken to ensure the causal links when analysing the results. For the most part in this thesis the Microscopic simulator SUMO is used - the Macroscopic model is only used to test stability in Chapter 4.

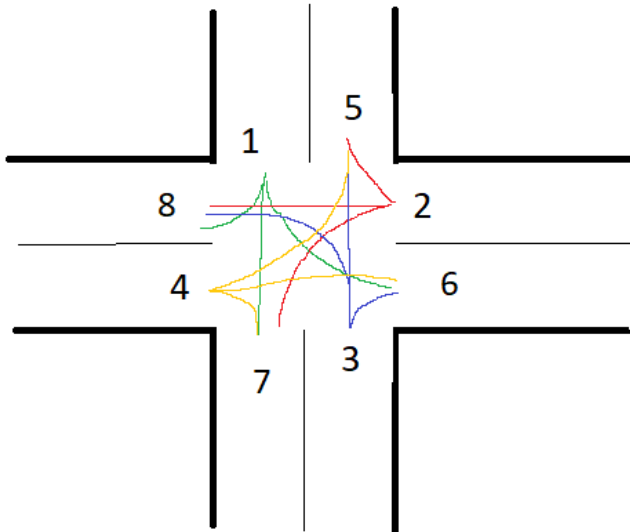


Figure 2.3 The single junction with the lanes numbered for reference and the different phases draw with different colours.

Macroscopic model used in the MATLAB simulations

A very simple model was used in MATLAB to test some of the more theoretical work in this paper, namely investigations into stability. It was chosen in this case because it can be made to simulate in a more deterministic way than in SUMO. The microscopic model used in SUMO to some extent models user behaviour and this induces a stochastic element to the simulation, granting less control over the outcome. Coding from scratch in MATLAB was thus chosen as a complement. The network model used in MATLAB uses infinite-buffer-point-queues. This means that the storing capacity of the queues are not formally upwards limited (the computer hardware might still impose restrictions), and that there is no simulated travelling time between junctions.

Two networks were used in the MATLAB simulations. The first one was the single junction in Figure 2.3, and the second network was the simple 2 by 2 junction network in Figure 2.4.

The model furthermore works as follows. Each simulation step all the queues in which vehicles are allowed to drive are emptied at a rate of the saturated flow rate. Vehicles are allowed to drive during a green light after start-up lost time has passed. The vehicles that leave a queue are then added to other queues according to

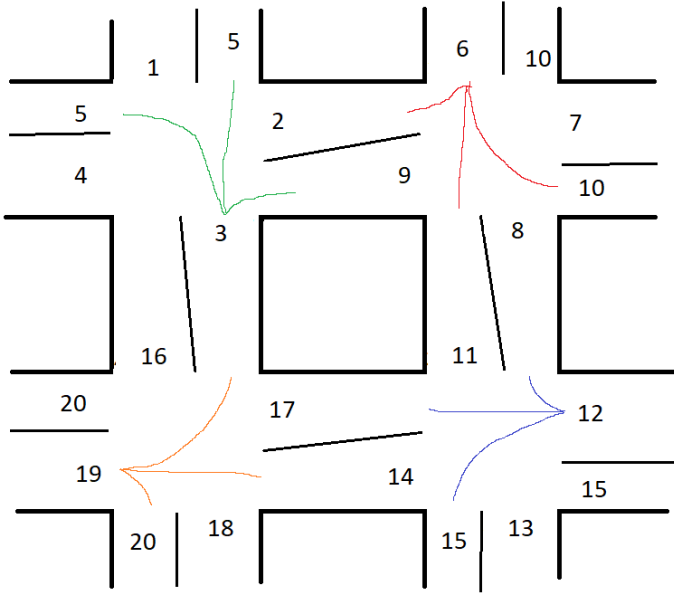


Figure 2.4 The simple 2 by 2 junction network with the different coloured phases. The phases have been drawn on different junctions for ease of seeing them even though all phases exist on all junctions. The different numbers are numbers used for reference.

which queues are connected to which, and turning ratios. The queues $x(t)$ updates according to

$$x_i^{n+1} = x_i^n + h(\lambda_i^n + \sum_j R_{ji} z_j^n - z_i^n) \quad (2.9)$$

commonly known as the Euler method of the first order, where h is the step length, and the superscript n is the time step number. Furthermore,

$$z_i^n \leq u_i^n(t_k^v) s_i^n \quad (2.10)$$

for all queues in the network where k is the maximum value such that $h \cdot n \geq t_k^v$. Remember that t_k^v are the times where the control is calculated at a junction v . In all MATLAB simulations a step size of 0.05 was used.

This model is not different from a time average model in theory, however, it was implemented actually executing the phases. A time average model instead of executing phases, lets the queues empty at a rate that is proportional to the fraction of allocated time in a cycle for the phases.

Even though the queues are not upwards limited in size, the measurements of them were saturated to simulate limited detector length.

The SUMO microscopic traffic simulator

SUMO - Simulation of Urban MObility is a suite of tools and programs developed partly as open source on the initiative by the Institute of Transportation under DLR - Deutsches Zentrum für Luft- und Raumfahrt. The suite is centered around main tools for creating networks and traffic, then simulating these. Via an API, programs can interface with a variety of things inside the simulation. The actual simulator, SUMO, is a so called microscopic simulator, simulating every vehicle independently.

The SUMO simulator uses the Krauss vehicle following model which is a model that in essence allows the vehicles to drive as fast as possible within the speed limits, as long as they can break to avoid a collision safely should the vehicle in front of them break [SUMO wiki 2018]. The vehicle following model is based on the accelerations of the vehicles.

ACTIVITYGEN and DUAROUTER are two tools that were extensively used during the simulations. First a population statistics file has to be created. This file specifies statistically where workplaces and homes are placed and is used by ACTIVITYGEN to create a file specifying where each person lives and works. Then DUAROUTER is used to create routes from this information. DUAROUTER creates routes according to shortest way and can be called again with the created routes to iteratively approach dynamic user equilibrium. However, the iterative approach was not done for this thesis. DUAROUTER was only run once to obtain routes.

In this chapter the different controllers used in this thesis are explained. They are the proportional allocation controller (with dynamic cycle length) using different mathematical norms (ways to measure the size of a set of values) and the Back-pressure controller. In the explanation of the controller the phase matrix is used extensively for ease of notation. \mathbf{P}_i is used to mean a row i in the phase matrix, and, therefore means the set of lanes that have green light during phase i . Since the phase matrix is a binary matrix all the matrix calculations boils down to simple sums. $\mathbf{P}_i x(t)$ means the sum of the queues in phase i . The notation $|\mathbf{P}_i|$ is the L^1 norm over row i in the phase matrix (the Taxi or Manhattan norm) and since it is a binary row it boil down to the number of queues in phase i . Another notation that is used hereafter is $A \odot B$. In this thesis it means element wise multiplication of vectors i.e. that $c_k = a_k \cdot b_k$, where c is the resulting vector from the operation.

2.3 Proportional allocation controller

The proportional allocation controller is a decentralised controller that does not need a priori information. It instead relies on measuring the queues that are queuing to the different phases. It, depending on the queue lengths, dynamically determines the cycle length and phase time allocation. The cycle time T as a function of the

queues $x(t)$ to the junction is determined by

$$T(x(t)) = T_w \left(1 + \frac{\sum_{1 \leq i \leq n_l} (x_i(t))}{\kappa} \right) \quad (2.11)$$

where κ is the design parameter that needs to be tuned, T_w is the total clearance lost time, and n_l is the number of lanes. This controller works both with orthogonal phases and with non-orthogonal phases, but for the sake of simplicity of calculation, orthogonal phases are always used in this thesis. The interested reader that wants to know how to make it work with non-orthogonal phases, should read [Nilsson and Como, 2018]. The proportion of time p_i allocated to a phase i is calculated from

$$p_i(x(t)) = \frac{\mathbf{P}_i x(t)}{\kappa + \sum_k x_k(t)} \quad (2.12)$$

where \mathbf{P}_i means the row i in the phase matrix corresponding to phase i . The nominator is simply the sum of the queues in the phase. Since $p_i(x(t))$ is the fraction of time that phase i receives service during the next cycle this corresponds to the control signal u_i in the more general case.

The proportional allocation controller as evaluated by [Nilsson and Como, 2018] and explained here, measures the sum of all the lanes included in the phase. This could be problematic since it is not taking the critical traffic volume into consideration but rather the total traffic volume. Think of a simple example: a junction with a one-lane phase and a six-lane phase. If there are six cars queuing for each phase they will get allocated the same amount of time even though the six lane phase would require a lot less time to empty the queues (if the cars on the six-lane phase are distributed evenly).

Therefore, two other ways of measuring the queues to the controller are here proposed using different norms. It should be noted that stability for these have not yet been proven. The two different norms are the mean norm and the max norm. The mean norm takes the mean queue length of every queue as the regulator parameter. It can be expressed as

$$T(x(t)) = T_w \left(1 + \frac{\sum_{1 \leq i \leq n_p} (\mathbf{P}_i x(t) / |\mathbf{P}_i|)}{\kappa} \right) \quad (2.13)$$

with n_p being the number of phases, for cycle time, and

$$p_i(x(t)) = \frac{\mathbf{P}_i x(t) / |\mathbf{P}_i|}{\kappa + \sum_{1 \leq i \leq n_p} (\mathbf{P}_i x(t) / |\mathbf{P}_i|)} \quad (2.14)$$

for the fraction of time allocated to phase i . The max norm uses the common norm $\|\cdot\|_\infty$ as the argument resulting in the cycle time being calculated as

$$T(x(t)) = T_w \left(1 + \frac{\sum_{1 \leq i \leq n_p} \max(\mathbf{P}_i \odot x(t))}{\kappa} \right) \quad (2.15)$$

with phase time allocation according to

$$p_i(x(t)) = \frac{\max(\mathbf{P}_i \odot x(t))}{\kappa + \sum_{1 \leq i \leq n_p} \max(\mathbf{P}_i \odot x(t))} \quad (2.16)$$

again with the same notation as in expressing the mean norm modification. These two different approaches are two attempts to combat the initial problem posed here regarding critical traffic volume.

The model for which the proportional allocation controller is proven stable

The model in which the proportional allocation controller is proven stable is a macroscopic, time-average, infinite-buffer-point-queues model. This means that, as usual for macroscopic models, the vehicles are not modelled individually, but rather as a continuous mass of vehicles. Time-average means that during control execution all queues get serviced at the same time, completely disregarding that vehicles in real life can collide. This works because of the fact that they get a fractional rate of service of the saturated flow rate, corresponding to the fraction amount of time of the cycle that has been allocated by the controller. For example: The controller decides to allocate 1/5 of the cycle time to a queue. In this model then, the queue gets to empty at a rate of 1/5 of the saturated flow rate, the entire cycle.

2.4 Back-pressure control

It might be argued that the Back-pressure controller is not decentralized in the sense that it needs information at surrounding junctions to function properly. It is decentralized in the sense that there is a separate controller for every junction. The Back-pressure controllers most basic component is the time slot that has a length t in time which can be said to be the cycle. During this time slot the phase in which the queues exerts the most pressure on the junction gets serviced.

The weight w_i of the queue x_i on the lane i is dependent also on the destination lanes j and the turning ratio R_{ij} as

$$w_i = x_i - \sum_j R_{ij} x_j. \quad (2.17)$$

So the weight is the difference between the queue length and the weighted average queue lengths of its destinations according to turning ratios. The pressure exerted by a phase i is then calculated as

$$\gamma(i) = \mathbf{P}_i(\mathbf{s} \odot \mathbf{w}) \quad (2.18)$$

where \mathbf{s} is the vector with the saturated flow rates and \mathbf{w} is a vector containing the different weights of the lanes. In conclusion, each time slot the phase with the largest pressure gets all the service. Hence, the control u_i is during a cycle the control signal is binary with the phase exerting the largest pressure receiving all the service.

3

Tuning of the Proportional Allocation Controller

To see how different operation conditions of the proportional allocation controller might affect the optimal value of κ , three different networks were simulated. They were the Manhattan network, see Figure 3.1, the Brickwall network which is a network with only T-junctions, see Figure 3.2, and a single junction network. The Manhattan network was varied in size, and traffic demand for the Manhattan and Brickwall networks was varied with the help of `ACTIVITYGEN` and `DUAROUTER`. The single junction networks had arriving traffic with a Poisson distributed arrival rate, that varied from a nominal value which is presented as the sum of critical lane volume (see Appendix A for details). Throughout this thesis the name of the network is used to specify the topology, it is then followed by the size specifications, and after a "-" the population is specified. For the Manhattan and Brickwall networks the size is the number of junctions on a side. The single junction networks have instead the number of avenues followed by the number of streets, and, since the nominal traffic demand for the single junction was not varied, no traffic demand specification is necessary aside from in Table 3.1. Avenues are vertical roads and streets are horizontal.

As an example: Manhattan 8 - 4000, would mean a grid type network that is 8 by 8 junctions in size, with a population of 4000. In Table 3.1 all the simulation scenarios carried out in this thesis are presented. More detailed information about the simulations and network setups can be found in Appendix A. From there the most valuable things to know is that the phases for all the networks are orthogonal, and that the phase setup is different depending on the topology of the network. Furthermore, the queue sensor length is the length of the road in most cases.

3.1 Optimality

To be able to say anything about the optimal value of κ , first optimality in the context of this thesis needs to be defined. Out of a pure network capacity perspective

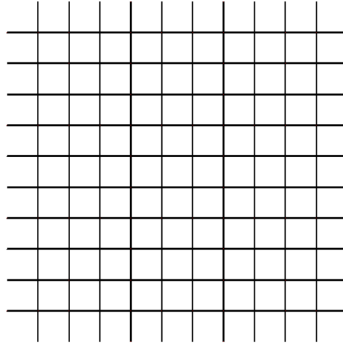


Figure 3.1 The Manhattan network used in simulations. Size of the depicted one is 10 by 10 junctions.

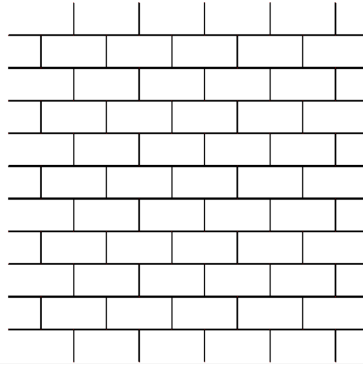


Figure 3.2 The Brickwall network used in simulations. Size of the depicted one is 10 by 10 junctions.

it would make sense to be able to process vehicles as fast as possible, to get them to their destination and therefore out of the network. This would mean that vehicles stay a shorter time in the network, which in turn means fewer vehicles present at every time instance. If vehicles on roads travelling between junctions are neglected this would also mean shorter queues. The most optimal κ therefore, would result in a control signal that produces the lowest integral of the sum of queues,

$$\min\left(\int_{start}^{finish} \sum_i x_i(t) dt\right) \quad (3.1)$$

where the queue vector x here includes all the queues in the network. However, since this approach might mean that the optimal way to control a junction would be

Table 3.1 Summation of the different simulation scenarios. *Junction with 80 meters sensor length.

Name	Type	Population	V_c
Manhattan 10 - 10000	10 by 10 junction grid	10000	
Manhattan 10 - 5000	10 by 10 junction grid	5000	
Manhattan 10 - 2500	10 by 10 junction grid	2500	
Manhattan 5 - 5000	5 by 5 junction grid	5000	
Manhattan 5 - 2500	5 by 5 junction grid	2500	
Manhattan 5 - 1250	5 by 5 junction grid	1250	
Brickwall 10 - 10000	10 by 10 junction brickwall	10000	
Brickwall 10 - 5000	10 by 10 junction brickwall	5000	
Brickwall 10 - 2500	10 by 10 junction brickwall	2500	
Junction 2-3	junction, 2 by 3 incoming lanes		1326
Junction 2-3*	junction, 2 by 3 incoming lanes		1326
Junction 2-6	junction, 2 by 6 incoming lanes		1178
Junction 2-6*	junction, 2 by 6 incoming lanes		1178

to always have green light to a heavily trafficked phase and almost never to a phase that has almost no traffic, drivers might feel like they are being treated unfairly. Fairness will be treated in the next section. All results in this section come from SUMO.

As can be seen from the plots in Appendix B and from Table 3.2 the most optimal κ for the Manhattan type networks seems to be very stable with regards to traffic demand and network size. However, the single junctions simulations do not appear to have a clear minimum as can be seen in e.g. Figure 3.7 and the rest of the figures in Appendix B. This might be due to the smallness of the network. A single junction wont produce spillover effects when the road is full. That is, the queue cannot go backwards past a previous junction (spill over) as it can in bigger networks.

A visual comparison between the different Manhattan networks indicates that a reasonable hypothesis, if a smaller network is used to approximate a larger one, is that the queue behaviours are the same for half the population in a half as big a network. However, this is not a completely satisfactory fit and it might be because of the fact that the traffic mostly commute in the east-west direction, thereby, in effect seeing the network as half the size. If the traffic were to have a more complicated driving pattern the results should change, with the same reasoning. This, however, remains to be verified.

The optimal value of κ for the Brickwall networks varies greatly with different traffic demand. Even though the general picture is the same with very long queues for lower κ , a minimum somewhere and increasing queues at higher values, the optimal κ seems to be higher for higher traffic demand.

One might question the validity of these results due to there being a small

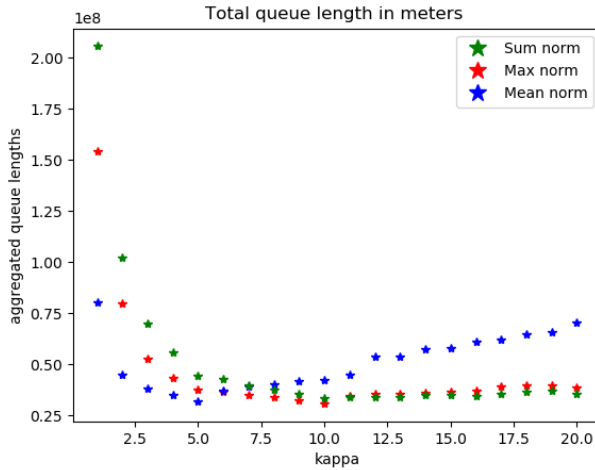


Figure 3.3 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 10 - 5000 network with a lane area detector covering the whole road.

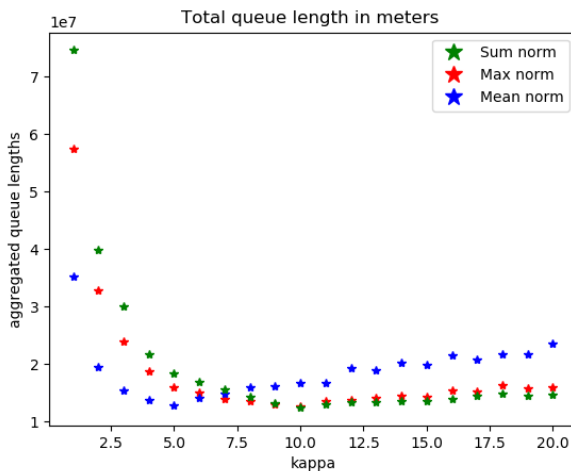


Figure 3.4 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 5 - 2500 network with a lane area detector covering the whole road.

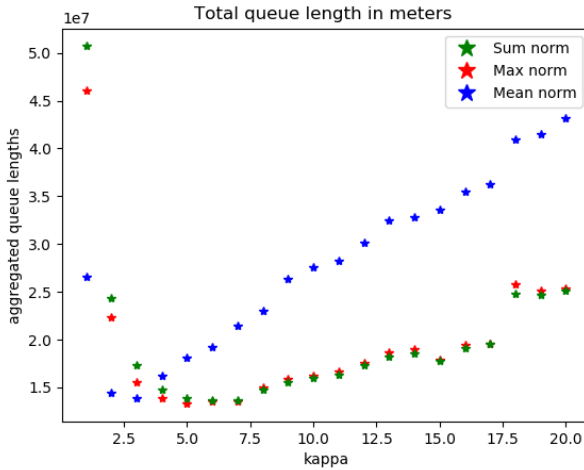


Figure 3.5 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Brickwall 10 - 2500 network with a lane area detector covering the whole road.

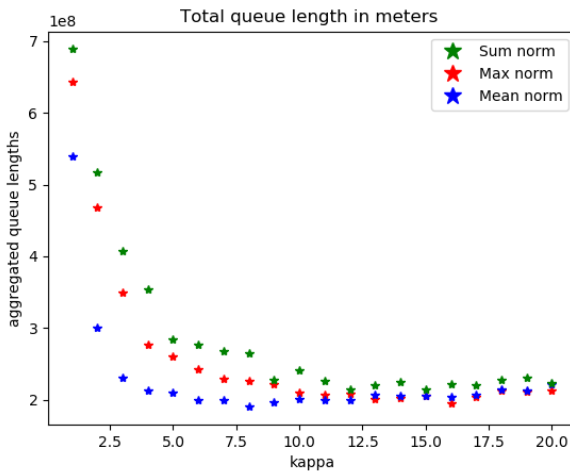


Figure 3.6 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Brickwall 10 - 10000 network with a lane area detector covering the whole road.

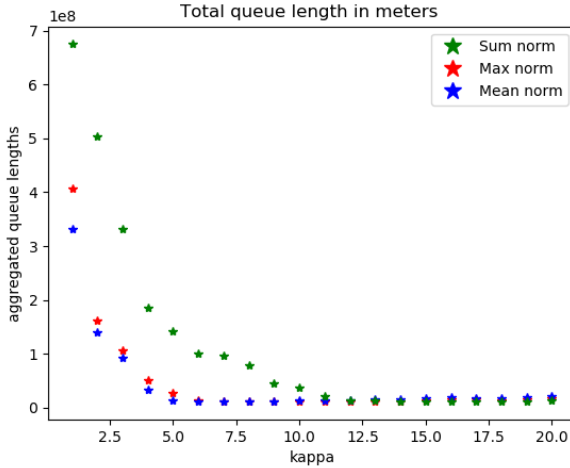


Figure 3.7 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Junction 2-3 network with a lane area detector covering the whole road, and a sum of critical lane volume of 1326.

Table 3.2 Summation of the results from the simulations. *Single junction networks with shorter sensor lengths. Max means when using equations (2.13) and (2.14). Mean means when using equations (2.15) and (2.16). Sum means when using equations (2.11) and (2.12).

Network	Optimal κ		
	(max)	(mean)	(sum)
Manhattan 10 - 10000	10	5	15
Manhattan 10 - 5000	10	5	10
Manhattan 10 - 2500	10	5	10
Manhattan 5 - 5000	10	5	10
Manhattan 5 - 2500	10	5	10
Manhattan 5 - 1250	9	5	10
Brickwall 10 - 10000	16	8	15
Brickwall 10 - 5000	15	3	15
Brickwall 10 - 2500	5	3	6
Junction 2-3	7	9	18
Junction 2-3*	9	7	15
Junction 2-6	12	5	14
Junction 2-6*	9	5	15

stochastic element to the creation of traffic. To produce confidence intervals on all the results would simply take too much time. However, box plots of the Manhattan 5 - 2500 network run 39 times can be found in Appendix B to make the results believable.

3.2 Fairness

Since optimal control of a junction might result in an unfair waiting time for some vehicles, the fairness of the control will here be examined. The main interest will be between the three different norms, the second point of interest how the most fair κ correlate with the most optimal κ for the three different norms. First Jain's fairness index [Jain et al., 1998] will be presented as the tool to measure fairness together with what should be fair. After that, fairness measurements from the previous simulations in Table 3.1 will follow. Lastly a small discussion of the results will ensue. All results in this section come from SUMO.

Jain's fairness index

Like optimality, the concept of fair needs to be defined. Here the mean speed is chosen as a measurement to make comparisons with, because this is probably what the commuters want to be equal. This would mean that it takes the same time to drive the same distance no matter in what direction the commute is desired. This would mean that a commute is predictable in an easier manner. The mean speed was calculated by adding the commute time and the waiting time to start the commute due to the road being blocked by a queue, then dividing by the commute length. To compare the mean speed of the vehicles, Jain's fairness index is used. Jain's fairness index is

$$\text{fairness} = \frac{\sum d_i^2}{M(\sum d_i)^2} \quad (3.2)$$

where M is the number of arguments, here vehicles, and d_i are the arguments, which here, as mentioned, are the mean speed of the vehicles. If all vehicles drive at the same speed the fairness would be 1. If instead the fairness is 0.8, this would mean that 80% have the same speed while 20% have no speed at all.

When looking at the results (all results are presented in Appendix C) the most fair κ seems to be quite consistent for the Manhattan networks. The value shifts slightly to lower values as the traffic demand increases as can be seen in Table 3.3 created from the figures in Appendix C. Between the different norms there seem to be no significant difference in fairness achieved (with different values of κ for the different norms). The Brickwall networks show results that are changing a lot with the highest traffic demand compared to the other two. This might have to do with the capacity of the network. Since roads have been removed in the Brickwall network, the capacity in this network is lower. This does not explain that the value of κ becomes higher however, since a higher value means shorter cycle times. For

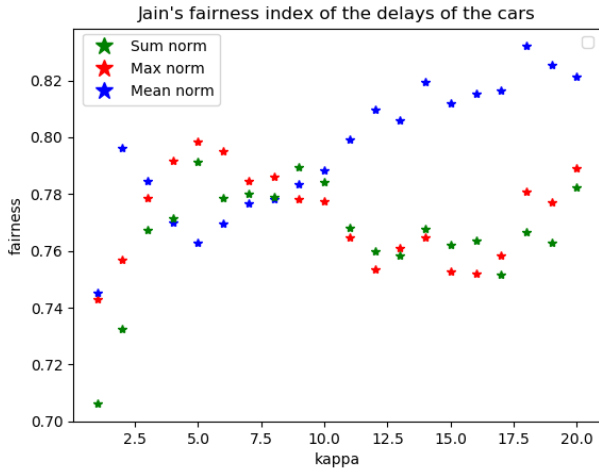


Figure 3.8 The Jain's fairness index for the Brickwall 10 - 10000 network.

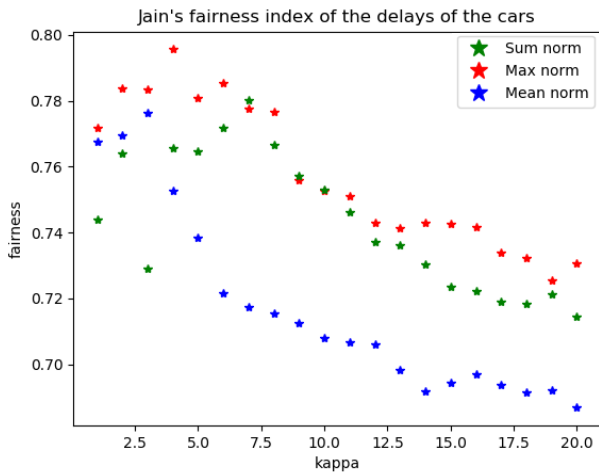


Figure 3.9 The Jain's fairness index for the Manhattan 5 - 5000 network.

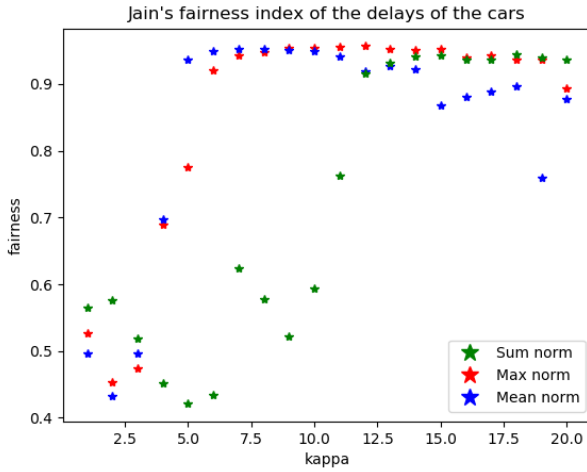


Figure 3.10 The Jain's fairness index for the Junction 2-3 network.

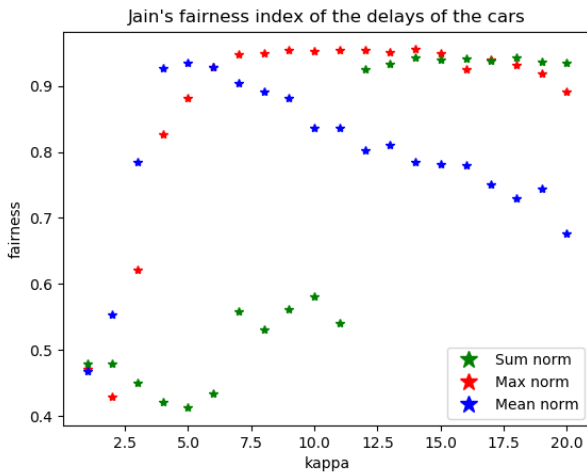


Figure 3.11 The Jain's fairness index for the Junction 2-6 network.

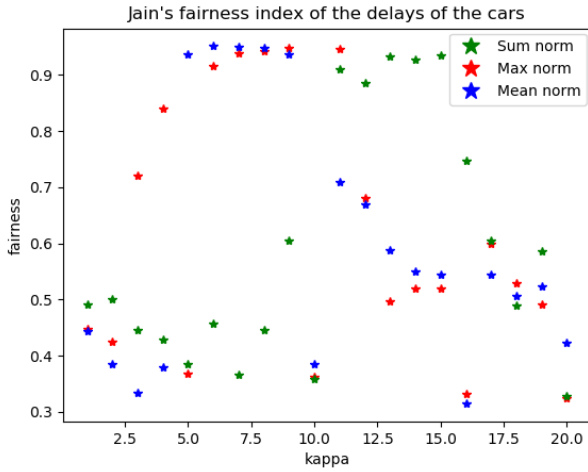


Figure 3.12 The Jain's fairness index for the Junction 2-3 network with the short-ened queue length.

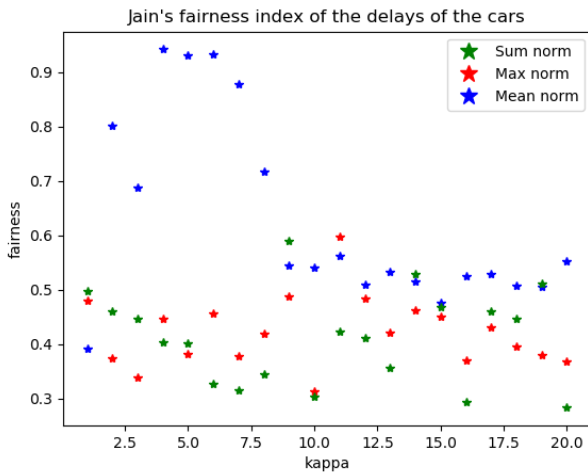


Figure 3.13 The Jain's fairness index for the Junction 2-6 network with the short-ened queue length.

Table 3.3 Summation of the fairness of from the simulations. *Results from the shortened sensor length to 80 meters. Max means when using equations (2.13) and (2.14). Mean means when using equations (2.15) and (2.16). Sum means when using equations (2.11) and (2.12).

Network	Most fair κ		
	(max)	(mean)	(sum)
Manhattan 10 - 10000	5	4	7
Manhattan 10 - 5000	6	3	9
Manhattan 10 - 2500	7	3	7
Manhattan 5 - 5000	4	3	7
Manhattan 5 - 2500	6	3	8
Manhattan 5 - 1250	8	4	8
Brickwall 10 - 10000	5	18	5
Brickwall 10 - 5000	6	2	5
Brickwall 10 - 2500	5	2	5
Junction 2-3	12	7	18
Junction 2-3*	9	6	15
Junction 2-6	14	5	18
Junction 2-6*	4	11	9

Table 3.4 Summation of the fairness from the simulations. *Results from the shortened sensor length to 80 meters. Max means when using equations (2.13) and (2.14). Mean means when using equations (2.15) and (2.16). Sum means when using equations (2.11) and (2.12).

Network	Most fair value		
	(max)	(mean)	(sum)
Manhattan 10 - 10000	0.884	0.860	0.858
Manhattan 10 - 5000	0.966	0.959	0.950
Manhattan 10 - 2500	0.986	0.985	0.984
Manhattan 5 - 5000	0.795	0.776	0.780
Manhattan 5 - 2500	0.920	0.903	0.910
Manhattan 5 - 1250	0.969	0.961	0.971
Brickwall 10 - 10000	0.789	0.832	0.791
Brickwall 10 - 5000	0.923	0.913	0.918
Brickwall 10 - 2500	0.983	0.981	0.982
Junction 2-3	0.956	0.952	0.944
Junction 2-3*	0.947	0.951	0.936
Junction 2-6	0.955	0.935	0.944
Junction 2-6*	0.597	0.943	0.590

the one junction networks with shortened sensor length there seem to be significant fairness loss for the sum norm, but a little unexpected also for the max norm, since the max norm was an attempt at producing fair results. See Figure 3.13 for reference. The mean norm far outperforms the two others. However, as can be seen in figures 3.10 and 3.11, the unfairness may, to some extent be countered by a longer sensor length. Since the waiting time to enter the network is accounted for in the fairness calculations, spillover effects are somewhat taken into consideration.

4

Required sensor length

So far the lane detector area, the sensor, has covered almost the whole of the length of the road (more than 95% of the length). Since this might be a bad reflection of reality, this chapter will delve into what sensor length is required to be able to use the optimal values of κ that was previously obtained. It will also investigate a little bit how that value might change as the sensor length gets shorter. Only the sum norm will be investigated here since it is proven to be stable in previous work and of demarcation reasons, to keep the work and results manageable. Similar calculations for the other norms should be rather straightforward.

4.1 Two analytic approaches

This section tries to come up with an analytical approach to establish the hard limitations on the guaranteed stable flow in a junction. First an approach without the start-up lost time is tried and then an approach where start-up lost time is considered as well. Since the clearing time is accounted for in the proportional allocation controller this need not be thought of.

Without start-up lost time

Let f denote the expected traffic demand on a single lane in veh/sec. The saturated headway is as usual denoted by h , and therefore the capacity of a single lane is h^{-1} . The fraction of the total time the lane receives service when controlled by the proportional allocation controller is

$$\frac{T(x(t)) \cdot p_i}{T(x(t))} = \frac{\mathbf{P}_i x(t)}{\kappa + \sum_k x_k(t)}, \quad x_m \leq L \quad (4.1)$$

where L is the lane detector length measured in vehicles. The notation $x_m \leq L$ and serve as a reminder that the sensor length is limited on all lanes. The fraction f/h^{-1} of flow has to be lower than the fraction of allocated time to the phase (and therefore

the lane). It follows therefore that the allowed flow becomes

$$fh \leq \frac{\mathbf{P}_i x(t)}{\kappa + \sum_k x_k(t)}, \quad x_m \leq L \quad (4.2)$$

which can be rewritten as

$$fh \leq \frac{\mathbf{P}_i x(t)}{\kappa + \mathbf{P}_i x(t) + \bar{\mathbf{P}}_i x(t)}, \quad x_m \leq L \quad (4.3)$$

to aid following arguments. The $\bar{\mathbf{P}}$ (it really is a d, but ignore that) is here used to mean the complement to \mathbf{P} if viewed as a boolean matrix, i.e. every 0 has been replaced by a 1 and vice versa. If the lane area detector is not full, the controller will not allocate the full amount of time it is able to do. Hence the queue will grow until the controller allocates enough time to stabilise the queue. The problem is as mentioned before that the lane area detector can only measure up to a certain length. Because of these two reasons it is of interest to see what constraints are imposed when one of the sensors is fully saturated. Let one of the sensors for a lane in a phase be saturated, i.e. that e.g. lane 1 is saturated ($x_1(t) = L$). Also, for notations sake, let $\hat{\mathbf{P}}$ be the modified phase matrix with the elements telling if $x_1(t)$ is in the phase be set to 0. This lane will be the investigated lane, i.e. the lane of which the capacity is of interest. The equation becomes

$$fh \leq \frac{L + \hat{\mathbf{P}}_i x(t)}{\kappa + L + \hat{\mathbf{P}}_i x(t) + \bar{\mathbf{P}}_i x(t)}, \quad x_m \leq L \quad (4.4)$$

which has minimum with respect to the queues not in the phase, when these queues are L long and maximum when they are empty. Since it is the minimum that is of interest to calculate allowed flow, let these lane area detectors be saturated and a number n such lanes. It follows that $\bar{\mathbf{P}}_i x(t) = nL$. The equation now becomes

$$fh \leq \frac{L + \hat{\mathbf{P}}_i x(t)}{\kappa + (n+1)L + \hat{\mathbf{P}}_i x(t)}, \quad x_m \leq L \quad (4.5)$$

To find minimum the derivative of the equation is taken and then set equal to zero. For readability let $X = \hat{\mathbf{P}}_i x(t)$.

$$0 = \frac{d}{dX} \frac{L+X}{\kappa + (1+n)L + X} = \quad (4.6)$$

$$\begin{aligned} &= \frac{1}{\kappa + (1+n)L + X} - \frac{L+X}{(\kappa + (1+n)L + X)^2} = \\ &= \frac{\kappa + (1+n)L + X - L - X}{(\kappa + (1+n)L + X)^2} = \frac{\kappa + nL}{(\kappa + (1+n)L + X)^2}, \quad x_m \leq L \end{aligned} \quad (4.7)$$

This clearly cannot be zero within the limits of $x_m(t)$. The boundary then has to have the minimum and maximum. For $\hat{\mathbf{P}}_i x(t) = 0$ the previous equation (4.5) becomes

$$fh \leq \frac{L}{\kappa + (1+n)L} \quad (4.8)$$

and for saturated measurements of the other queues in the phase $\hat{\mathbf{P}}_i x(t) = |\hat{\mathbf{P}}_i|L$ the equation becomes

$$fh \leq \frac{aL}{\kappa + (a+n)L} \quad (4.9)$$

where a is the number of lanes in the phase including the lane of interest. As can be seen, the value of equation (4.8) is clearly smaller than the value of equation (4.9). This means that the minimum is when the rest of the queues in the phase in which the lane of interest is in are empty ($\hat{\mathbf{P}}_i x(t) = 0$). The capacity of this lane then is guaranteed to be

$$f \leq \frac{L}{h(\kappa + (1+n)L)}. \quad (4.10)$$

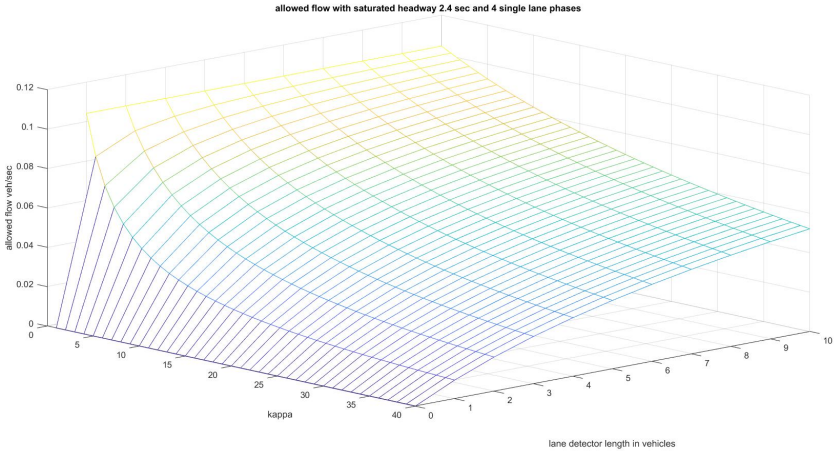


Figure 4.1 The allowed flow on one lane in a four phase junction with one lane in each phase. Saturated headway is 2,4.

With start-up lost time

Let f once again denote the expected traffic demand on a single lane in veh/sec, and the saturated headway is again denoted by h . Now, l_1 is the start-up lost time for a cycle. The time lost per hour due to the start-up lost time is therefore l_1 times the number of cycles. The number of cycles in an hour is the inverse of the cycle length.

Therefore an additional fraction $l_1/T(x(t))$ needs to be accounted for. The equation to investigate now becomes

$$fh \leq \frac{\mathbf{P}_i x(t)}{\kappa + \sum_k x_k(t)} - \frac{l_1}{T_w(1 + \sum_k x_k(t)/\kappa)}, \quad x_m \leq L \quad (4.11)$$

which can be rewritten to

$$fh \leq \frac{T_w \mathbf{P}_i x(t) - l_1 \kappa}{T_w(\kappa + \mathbf{P}_i x(t) + \bar{\mathbf{P}}_i x(t))} := E, \quad x_m \leq L \quad (4.12)$$

and called E to make the following calculations simpler. The minimum with regards to the queues not in the phase of interest is once again obvious to be $x(t) = L$. Let there be n such queues. The new equation is now called E' .

$$\min(E)_{\bar{\mathbf{P}}_i x(t)} = \frac{T_w \mathbf{P}_i x(t) - l_1 \kappa}{T_w(\kappa + \mathbf{P}_i x(t) + nL)} := E', \quad x_m \leq L \quad (4.13)$$

Remaining is now to minimise with respect to the queues in the phase and the queue of interest. The queue of interest is called queue a . For the same reasons as before with no start-up lost time, let queue a grow until saturated measurement. Thus, this queue becomes L long.

$$E' = \frac{T_w(\hat{\mathbf{P}}_i x(t) + L) - l_1 \kappa}{T_w(\kappa + \hat{\mathbf{P}}_i x(t) + (n+1)L)} \quad (4.14)$$

For readability let $\hat{\mathbf{P}}_i x(t) = X$.

$$\min(E')_X = \min\left(\frac{T_w(X+L) - l_1 \kappa}{T_w(\kappa + X + (n+1)L)}\right) \quad (4.15)$$

To find minimum differentiate with respect to X and set equal to 0.

$$\begin{aligned} \frac{dE'}{dX} &= \frac{d}{dX} \frac{T_w(X+L) - l_1 \kappa}{T_w(\kappa + X + (n+1)L)} = \frac{1}{\kappa + X + (n+1)L} - \frac{T_w(X+L) - l_1 \kappa}{T_w^2(\kappa + X + (n+1)L)^2} = \\ &= \frac{T_w^2(\kappa + X + (n+1)L) - T_w(X+L) + l_1 \kappa}{T_w^2(\kappa + X + (n+1)L)^2} = 0 \end{aligned} \quad (4.16)$$

Solved for X this becomes:

$$X = \frac{T_w L - l_1 \kappa - T_w^2(\kappa + (n+1)L)}{T_w^2 - T_w} \leq \{T_w > 1, \text{all constants positive}\} \leq 0 \quad (4.17)$$

The greatest limitation is therefore when all the queues not in the phase are full and all the other lanes in the phase are empty. The flow therefore is limited by equation (4.18).

$$f \leq \frac{T_w L - l_1 \kappa}{h T_w(\kappa + (n+1)L)} \quad (4.18)$$

4.2 Simulations in MATLAB

To verify the results obtained in the theoretical work MATLAB, simulations were performed using the model described in the model chapter (Chapter 2). During the simulations the incoming flow rate was kept constant while the detector length and value of κ were varied. This requires a rearranging of equation (4.10) to

$$\kappa \leq \left(\frac{1}{fh} - (1+n) \right) L \quad (4.19)$$

and (4.18) to

$$\kappa \leq \frac{T_w(1 - fh(n+1))}{fhT_w + l_1} L. \quad (4.20)$$

To simulate the lane detector length the measurements of the queues were saturated at the desired number before control signals were calculated. As a measurement of under what circumstances a network becomes unstable, the external net flow to the network was calculated as the main indicator. The net flow is the flow out from the network subtracted from the flow into the network. Therefore, when the external net flow is larger than zero the queues in the network are unstable. As complementary information the aggregated queue lengths of the simulations are presented in Appendix D.

Single junction network

To get the simplest test of how the stability is limited under the circumstances layed out in the theoretical section, a single junction was simulated. To test both approaches, simulations with a 2 seconds start-up lost time and no start-up lost time was simulated. Clearing time for both simulations was 2 seconds. The junction is depicted in Figure 2.3 with the lanes numbered for reference. Incoming traffic was here added on only lane 1 with a rate of 0.052 vec/s (1/8 of the saturated flow rate). However, the controller was made to believe that the other incoming lane detectors were always full. Lane detector length was varied from 1 to 12 vehicles in steps of 1 and κ was varied from 1 to sufficiently large to get unstable, in steps of 1. Only the sum norm was used. Both the aggregated queue lengths and the net flow to the network was plotted to get indications for the stability. The simulation was run for 10000 seconds in steps of 0.05. The calculation of net flow was started after 2500 seconds to accommodate for transients in the system.

The resulting figures showing net flow are figures 4.2 and 4.3 for the no start-up lost time and 2 seconds start-up lost time respectively. The aggregated queue lengths can be found in Appendix D.

As can be seen in the results the relationship between the lane detector length and the last stable κ seems to be linear for both simulations. This is confirmed by linear fits of the last stable κ , against the corresponding lane detector length. The linear fits can be seen in Figure 4.4 together with predictions from the analytical

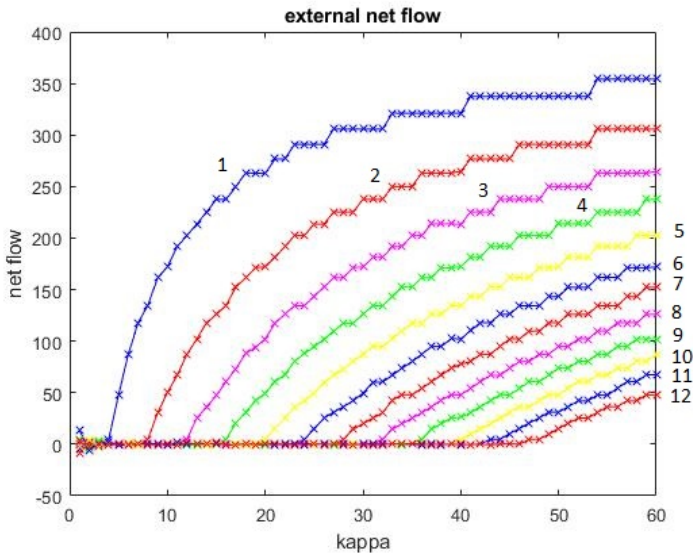


Figure 4.2 Net flow in the 1 junction network with no start-up lost time. Measured after 2500 seconds. Numbers next to the coloured lines are lane detector lengths in vehicles.

expressions. As can be seen the prediction when no start-up lost time was simulated or accounted for, is rather good, however, when start-up lost time is accounted for the prediction is poor.

Simple 2 by 2 junction network

Here a 2 by 2 junction network was simulated to see if the results from the simulation follow the same pattern as before. In Figure 2.4 depicting the network, the phases have been drawn on different junctions for ease of seeing them. All phases exist on all junctions though. The lanes are numbered for reference.

Incoming traffic was here added on all incoming lanes to the network (lanes (1, 4, 6, 7, 12, 13, 18, 19)) at a rate of 0.052 vec/s (1/8 of the saturated flow rate, the same as in the single junction network). The turning ratios was for all lanes (left, straight, right) = (0.2, 0.5, 0.3). Both start-up lost time and clearing time was 2 seconds. Lane detector length was varied from 1 to 12 vehicles in steps of 1 and κ was varied from 1 to 40 in steps of 1. Only the sum norm was used. Both the aggregated queue lengths and the net flow to the network was plotted to get indications for the stability. The simulation was run for 10000 seconds in steps of 0.05. The calculation of net flow was started after 2500 seconds to accommodate for transients in the system. The results for this simulation is shown in Figure 4.5 and complementary in Appendix D.

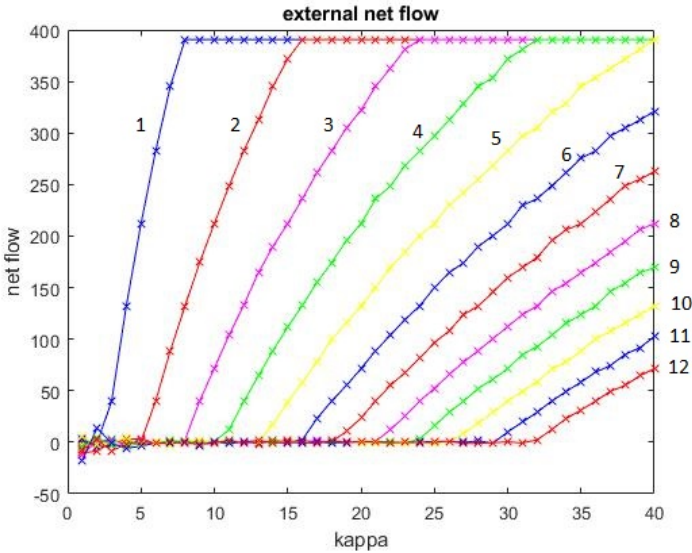


Figure 4.3 Net flow in the 1 junction network with 2 seconds start-up lost time. Measured after 2500 seconds. Numbers next to the coloured lines are lane detector lengths in vehicles.

The relationship for the 2 by 2 network seems to be linear as well which is confirmed by the plot in figure 4.6.

4.3 Simulations in SUMO

To test how the optimal κ would depend on lane detector length in a more realistic setting, a Manhattan 5 network was simulated in SUMO with varying sensor lengths. First a population of 5000 was simulated going to work at the same time, for all three norms, and then a population of 10000 using only the sum norm, for reasons of limited time to run simulations. Both with a probability 0.58 of owning a car. The optimal kappa with respect to aggregated queue lengths was then found and plotted against detector length. In SUMO the detector length is in meters. The results from these were complemented with the results from the Manhattan 5 - 5000 with 375 meter lane detector length (length enough to cover the whole length of the road, same as in the simulations mentioned in Chapter 3). They can be seen in Figure 4.7. As can be seen in the figure, the optimal value of κ doesn't change when the sensor length is above a certain length. It might be that the lane detector was not fully utilised throughout the simulations due to not high enough traffic demand.

The results here have shown that there exists a linear relationship between the

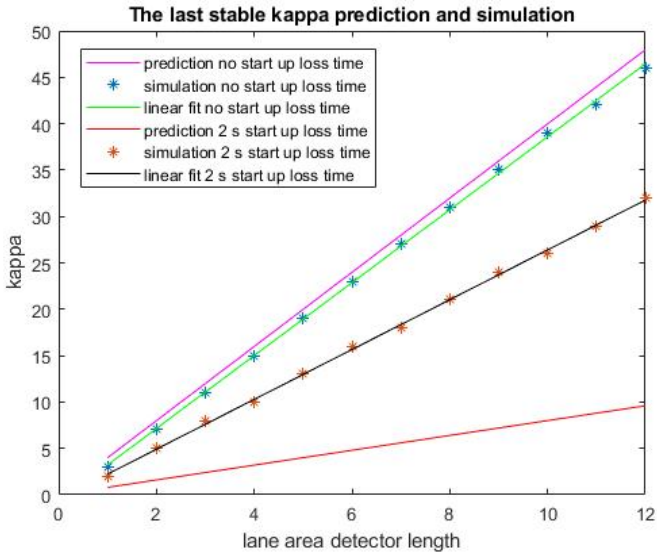


Figure 4.4 Linear fit of the last stable kappa in the single junction network with respect to lane detector length. The analytical prediction for the two different cases with and without start-up lost time is also plotted. Sensor length is in vehicles.

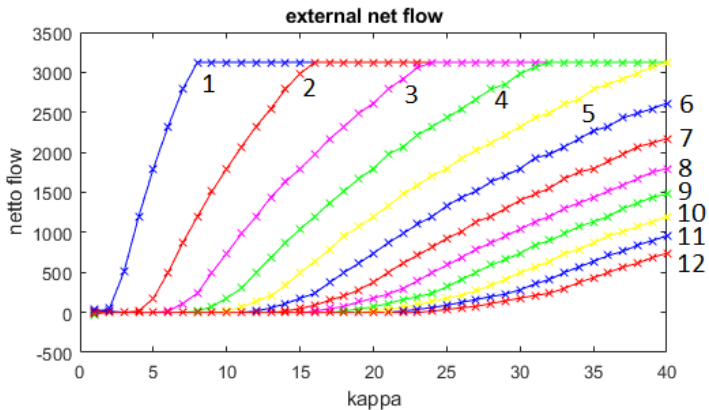


Figure 4.5 Net flow in the simple 2 by 2 junction network. 2 seconds start-up lost time. Measured after 2500 seconds. Numbers next to the coloured lines are lane detector lengths in vehicles.

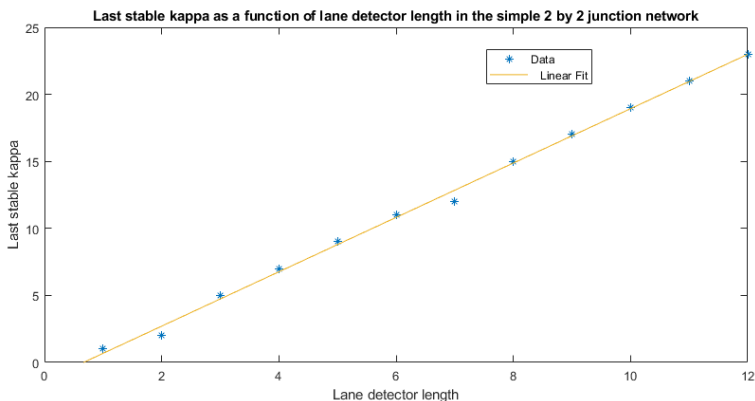


Figure 4.6 Linear fit of the last stable kappa in the simple 2 by 2 junction network with respect to lane detector length. Sensor length in vehicles.

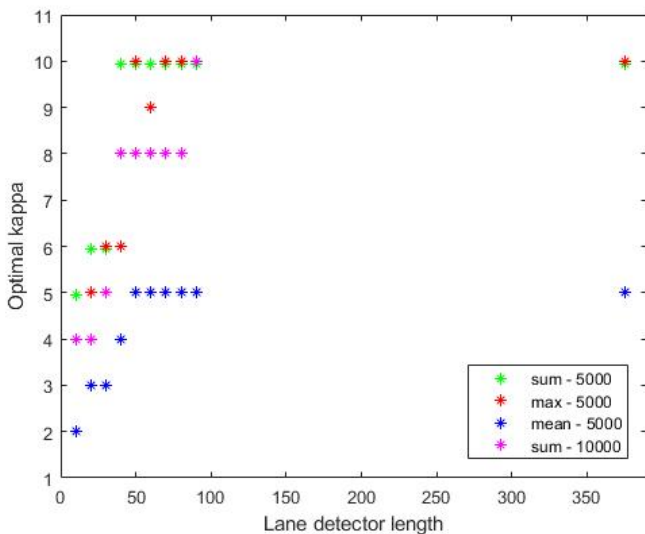


Figure 4.7 Optimal κ from the Manhattan 5 - 5000 network with varying lane detector length, and the from the sum norm from a Manhattan 5 - 10000 network. The green data points from the sum norm are shifted 0.05 down for visibility. Sensor length in meters.

sensor length and the last stable κ given a certain flow. This relationship is for one junction almost correctly predicted by an analytical deduction, given a very simple model. When a more complex model is used the analytical approach predicts stability issues earlier than the simulations show. With this said, since the optimality results rarely show the optimal κ above 10 (there is an outlier for the Manhattan 10 - 10000 network, in Figure B.7 but the results for $\kappa = 10$ and $\kappa = 15$ are very similar), from the results in Figure 4.7 it would seem that a sensor length above 100 meters in implementation in the real world, would be of little use. If a shorter sensor length is required for other reasons, the value of κ should be chosen lower as can be seen in Figure 4.7. This is however dependent on the assumed traffic demand as a lower traffic demand would have the same optimal κ until the detector length becomes smaller.

The stability issues would in real life most likely be encountered at a slightly higher κ except in more extreme cases with single lane phases competing with heavily trafficked multi-lane phases. In this case the max or mean norm should be considered since they in theory should produce more fair phase allocation. However, there exists no stability proof of these norms as of yet. Furthermore, the simulations with the single junction networks show that if the sensor length is long enough, the sum norm does not become unfair.

5

Performance comparison with Back-pressure control

To do a quick comparison between the Proportional Allocation and the Back-pressure controller a Manhattan 5 network was simulated. However, since turning ratios are required for the Back-pressure controller to work, traffic was created a little bit differently. Instead of a population statistics file, 6 sources and destinations were set up, reference Figure 5.2 for more details. Routes were then created with the help of turning ratios and the sumo tool JTRROUTER. The turning ratios were the same for all junctions and they are presented in Table 5. The turning ratios were the same for the two horizontal directions and different from the two vertical directions. The two vertical approaches had the same turning ratios. The amount of traffic that was added varied throughout the simulation time according to the numbers in Table 5.2. The simulation was stopped after 8000 seconds. Time slot lengths varying from 5 seconds to 90 seconds were tried.

With the Back-pressure controller there is a problem with input lanes that allow vehicles to drive to multiple output lanes, if there is another input lane that simultaneously allows vehicles to drive to one or more of those output lanes. The fundamental problem is that it is somewhat undetermined how to calculate the pressure of a lane in these cases. In this thesis the pressure calculations have been implemented as follows. If there are multiple input lanes that share an output lane, the input lanes are treated as a single lane. This will be called a *combined lane* and the lanes that make it up will be called *sublanes*. The queue that is on a combined input lane is calculated as the total queue of the sublanes. Sublanes on a combined output lane gets, in calculations, a queue that is the average of the sublanes. There is nothing that can prevent a combined lane to only consist of a single sublane.

Now reference Figure 5.1. As can be seen it is rather straight forward to calculate the final turning ratios of the different output lanes once the approximation of combined lanes has been done. With mathematics all this can be expressed as the following. Let \hat{x}_i denote the queue of a combined input lane and \bar{x}_j the queue of

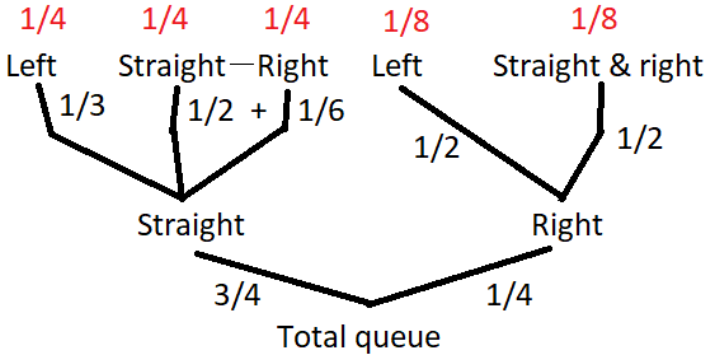


Figure 5.1 How the statistical distribution of vehicles will be downstream from the go straight or turn right horizontal phase in the Manhattan networks used in this thesis (the blue phase in Figure A.1). Final turning rates are in red. Reference Appendix A for more details on this phase setup.

Table 5.1 The turning ratios depending on the approach to the junction.

	right	straight	left
Horizontal	0.166	0.50	0.333
Vertical	0.25	0.25	0.50

Table 5.2 The amount of traffic added to every source of traffic.

time interval	number of vehicles
0-500	50
500-1000	300
1000-1500	100

sublanes of a combined output lane. The calculations of the pressure now become

$$w_l = \hat{x}_l - \sum_m R_{lm} \bar{x}_m \tag{5.1}$$

which gives the following expression (which is simplified further because of the standardised saturated flow rate s in the simulations) for the pressure exerted by a phase i as

$$\gamma(i) = \mathbf{P}_i \mathbf{s} \mathbf{w}. \tag{5.2}$$

The results are hard to interpret due to the fact the the simulation time is too small to let all the vehicles finish their route. This is a problem. First of, one observation is that the minimum of the proportional allocation controller seems to be reasonably consistent with previous results in this thesis even though it is somewhat skewed

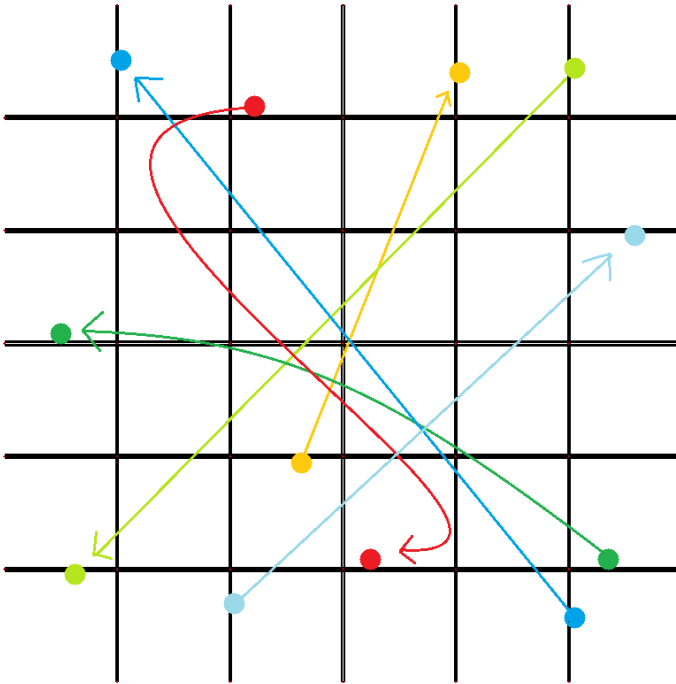


Figure 5.2 The sources and destinations of the traffic used in the comparison between the Back-pressure controller and the proportional allocation controller.

because of the simulation time constraint. Secondly, it can from first glance seem like the Back-pressure controller produces shorter aggregated queue lengths if the slot time is correct. This is confirmed by the figures 5.4 and 5.6 that indicates that the Back-pressure controller indeed is able to process the traffic at a higher rate.

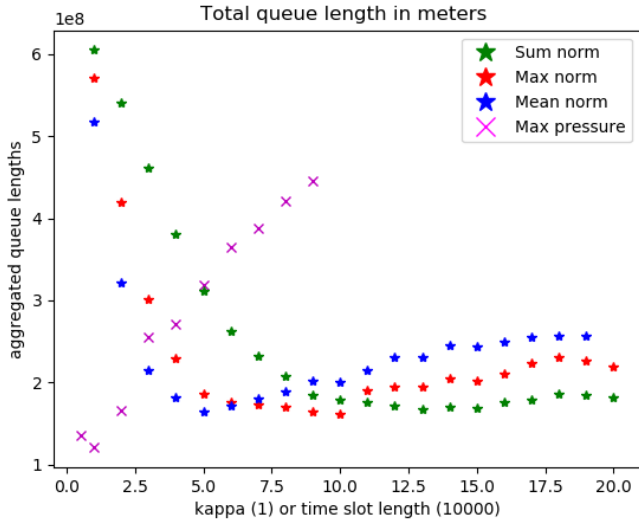


Figure 5.3 Aggregated queue lengths from the different controllers with different settings. These results cannot be interpreted straight off since all the vehicles have not left at the simulation end time. Slot time is in milliseconds.

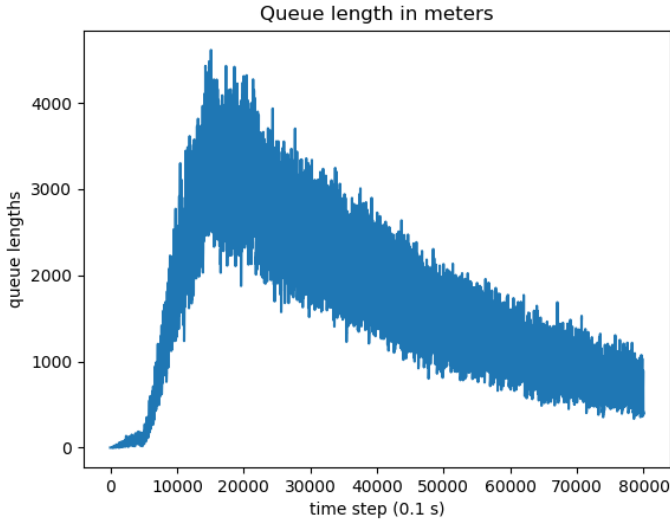


Figure 5.4 Queue lengths over time during the simulation. Back pressure controller. Slot time 10 seconds.

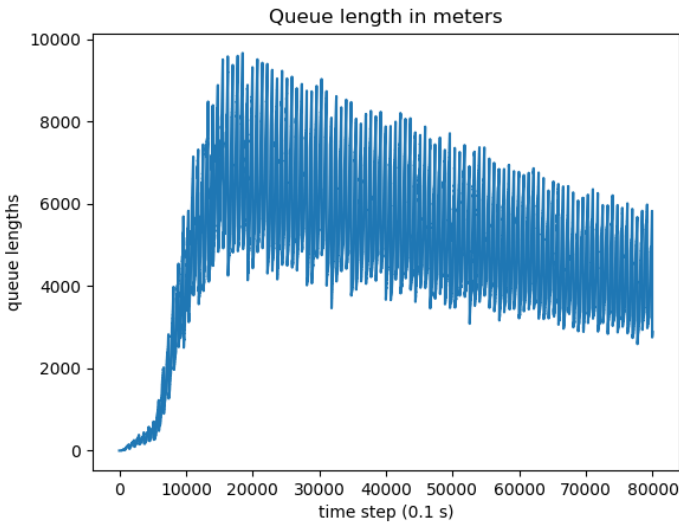


Figure 5.5 Queue lengths over time during the simulation. Back pressure controller. Slot time 70 seconds.

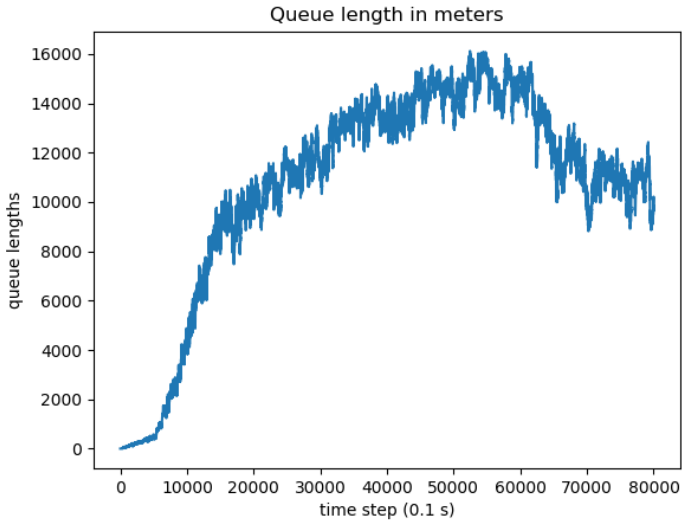


Figure 5.6 Queue lengths over time during the simulation. Proportional allocation controller, sum norm, $\kappa = 10$.

6

Discussion

The results herein do depend to some extent on a stochastic element which is the populace probability of owning a car. The consequence of this is that many more sets of simulations need to be produced to be able to set confidence intervals on the results. With this said, it might still be possible to draw some general conclusions. First of all the results obtained from the optimality agrees with the results in [Nilsson and Como, 2018] in as so far as the optimal κ for the sum norm is around 10 (see Table 3.2). Next thing to say is that the optimal κ does not seem to depend on the traffic demand in a Manhattan type network, with regards to the results as can be seen in Table 3.2. Therefore, to ease the burden on the CPU and speed up simulations, networks can be made smaller in size when simulating uniform grid networks. Not too small though, as the results from the Junction 2-3 network are different than the bigger networks. Spillover effects might be a big error factor here but if these can be accounted for, single junction simulations might prove useful. One way might be to see if a shorter sensor length might emulate spillover effects, but the results showed that it cannot, as the results in Table 3.2 show different optimal κ than the bigger networks. Another way could be to count the number of cars waiting to enter the junction as additional queue. More extreme traffic demands might prove that smaller networks cannot be used instead of bigger.

A thing that might be a big affector of the choice of κ is the saturated headway that is remarkably small in SUMO (1.1 from [Wikipedia, 2018]) compared to what is cited in [Roess et al., 2011] (2.4). This should have a great impact on how one would choose κ since the saturated headway directly affects how long it will take to empty a queue. If the case is that saturated headway is of great importance, weights might also need to be added to junctions that are placed in such a way that the saturated headway is dependent on from which direction the vehicles arrive. The required time to empty a queue, if to believe [Roess et al., 2011] and the saturated headway cited in [Wikipedia, 2018], should be a little bit more than twice the time in reality which would mean that all values of κ in the SUMO simulations should be halved.

When the network topology and phase setup is changed, this heavily affects the optimal κ . If it is the phase change or the network topology that influences this

change is hard to say, a guess is that it is both, but this needs more investigation. One thing to note is that the Brickwall network of a certain size has roughly only 3/4 the capacity of a similar sized Manhattan network due to the removal of one of the roads in each city block.

Among the norms, for the multi-junction networks, over all they perform similarly for their respective optimal κ . When looking for fairness, the max norm performs best by a small margin (see figures in Chapter 3). However, for another κ than the optimal one. This produces a problem of compromise between fairness and optimality.

From figures 3.13 and 3.12 for the fairness it can clearly be seen that in an unfair junction with short sensor length, the mean norm produces almost unchanged fairness results between a fair and an unfair junction, while the other norms are severely worsened compared to a more fair junction. However the unfairness effect seems to be compensated for with a longer sensor length as can be seen in figures 3.10 and 3.11. This in turn might only work if the traffic demand is low enough since the controller is stabilising. The stabilising property means, as mentioned before, that a queue will grow until the allocated time is enough to stabilise the queue. This means that the queues themselves might be very uneven, but that the service time as a result of this becomes more fair. This of course does not work if the traffic demand is so large that it will saturate the sensor, which will happen if the traffic demand is too large or the sensor too short.

Over all the mean norm's fairness is more sensitive to the value of κ compared to the others. This might not be a problem however, since the most fair value of κ when using the mean norm is very close to the optimal value.

It has been shown in Chapter 4 that the sensor length has a big impact on the choice of κ for the controller to remain stable. The relationship is shown to be linear in a simple model, but the relationship might prove to be more complicated in a more complex model. The important thing to take into consideration is how a very short sensor length (<40m) might affect the value of κ . For such short sensor lengths it seems like the most optimal thing for the controller to do is to allocate more time than necessary for the measured queue, so that the vehicles in the unmeasured queue has a chance to drive, congruent with standard static control practices (see for example [Roess et al., 2011]), to over allocate cycle time so that the peak hour traffic can be processed. To over allocate time in turn might prove to be unoptimal if the measured queue in fact is the whole queue.

The results in this paper would suggest that if not much resources are available when tuning traffic lights controlled by the proportional allocation controller, the mean norm should be used with $\kappa = 4$ and a sensor length of at minimum approximately 60 meters. Higher traffic demands would require a longer sensor length. If more resources are available, first the compromise between optimality and fairness should be taken into consideration and if possible the junction in itself should be made more or less fair depending on the expected traffic. Then simulations should be carried out to determine the optimal κ . In this case the max norm with values

from 5 to 11 can be used as a starting point. Here, κ should also be checked so that the value is well inside the stable region with regards to sensor length and estimated maximum traffic demand. In both cases, if guaranteed stability is required, proofs of the stability of these norms first needs to be provided.

As a final note it can be said that the Back-pressure controller achieves better results under similar circumstances, however, it can be argued that it is not a truly decentralised controller.

7

Future work

To construct a decentralised self tuning controller might prove tricky to do. The only information available would be the queue lengths from the sensors, but as have been discussed here, the optimal κ would be one that is over-allocating time. One way might be to check if any significant queue is left after phase execution. If there is, decrease κ . However, to make this tuner stable might prove difficult if done in a sophisticated way. The simplistic way would be to have the value of κ saturated between, let's say 4 and 15. This was not explored during the work here, but remains as a future challenge.

Even though the exploration in this thesis is quite extensive it still leaves many variables' impact on the optimal value unanswered. Most notably could be if it is the topology or phase setup that affects the optimal value to such a large extent. Why is the value in the Brickwall network fluctuating so much while it is stable in the Manhattan network, and what is the impact of varying saturation headways?

Bibliography

- Gregoire, J., E. Frazzoli, A. de La Fortelle, and T. Wongpiromsarn (2014). “Back-pressure traffic signal control with unknown routing rates”. *IFAC Proceedings Volumes* **47**:3, pp. 11332–11337. ISSN: 1474-6670. DOI: 10.3182/20140824-6-ZA-1003.01585. URL: <https://www.sciencedirect.com/science/article/pii/S1474667016434178>.
- Jain, R., D. M. Chiu, and H. William R (1998). “A quantitative measure of fairness and discrimination for resource allocation in shared computer systems”. **cs.NI/9809099**.
- Kovacs, P., T. T. Le, R. Núñez Queija, H. H. Vu, and N. Walton (2016). *Proportional green time scheduling for traffic lights*. URL: <https://www.narcis.nl/publication/RecordID/oai:cwi.nl:25248>.
- Munroe, R. *Long light*. Here used under CC BY-NC 2.5. URL: <https://xkcd.com/277/>.
- Nilsson, G. and G. Como (2018). “Evaluation of decentralized feedback traffic light control with dynamic cycle length”. In: *Proceedings of the 15th IFAC Symposium on Control in Transportation Systems*. BuildSys ’15. Savona, Italy. URL: https://ifac.papercept.net/conferences/conferences/CTS18/program/CTS18_ContentListWeb_3.html.
- Roess, R., E. Prassas, and W. McShane (2011). *Traffic engineering, fourth edition*. Pearson.
- SUMO wiki* (2018). URL: http://sumo.dlr.de/wiki/Definition_of_Vehicles,_Vehicle_Types,_and_Routes#Default_Krauss_Model_Description.
- Tassioulas, L. and A. Ephremides (1992). “Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multihop radio networks”. In:

- Varaiya, P. (2013). “Max pressure control of a network of signalized intersections”. *Transportation Research Part C: Emerging Technologies* **36**, pp. 177–195. ISSN: 0968-090X. DOI: 10.1016/J.TRC.2013.08.014. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X13001782>.
- Wikipedia (2018). *Traffic light*. URL: https://en.wikipedia.org/wiki/Traffic_light#History (visited on 2018-06-04).

A

Simulation scenarios

A number of different simulation scenarios were tried in simulation. In general three main components were varied: traffic volume in most cases in the form of population, size of the network, and network topology. In all networks vertical roads will be called *avenues* and horizontal roads will be called *streets*. All networks where it is not explicitly written otherwise, the networks had 2 lanes per direction on avenues and 3 lanes in each direction on streets. The centre of the junctions are in all networks placed 400 meters apart creating road lengths of a minimum of 375 meters. These 375 meters are, where not written otherwise, covered with a sensor counting the number of vehicles in the queue (if the vehicle is moving it is not in the queue). When simulating all the bigger networks (Manhattan 10 and 5, and Brickwall 10) the populace was set to start working 1000 seconds after simulation start, and the simulation was stopped after 4000 thereby allowing almost all the vehicles to finish their commute.

A.1 Phase setup

In this section the different networks phase set up will be presented.

Manhattan networks

The cycle is divided into 4 orthogonal phases that do not allow the vehicles to collide. They are the same as used in [Nilsson and Como, 2018] and are depicted in Figure A.1. They are always executed in the same order starting with the phase allowing traffic flowing east to west and west to east, the blue phase in the figure.

Brickwall network

The phases in this network will understandably be different because of being T-junctions. Still orthogonal, the number of phases is reduced to 3. The order of execution is always the same but different for T-junctions that are oriented in the opposite way. Refer to Figure A.2 for a clear layout of the phases.

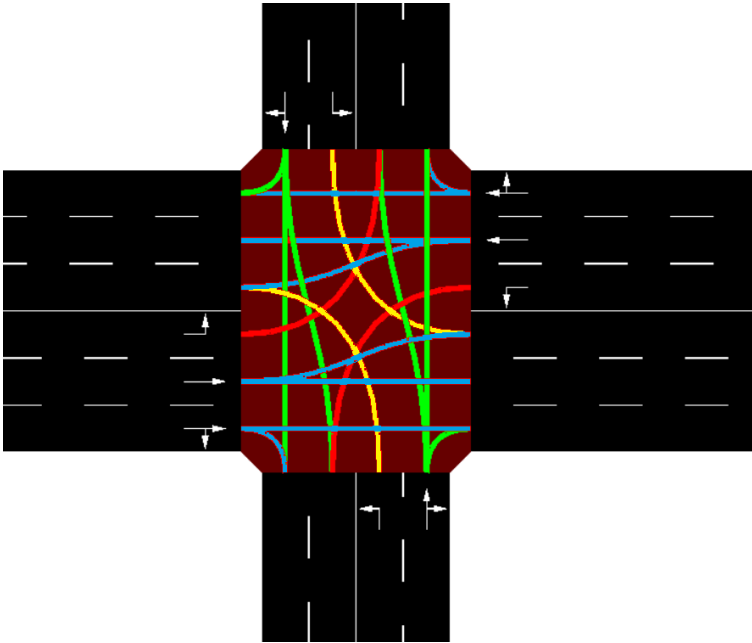


Figure A.1 The phase setup of the junction used for simulation in all of the grid networks. The phase execution order is always the same starting with the blue one.

Single junction network

Of a smaller but still noticeable importance two different single junctions were simulated. A junction that looks exactly as in Figure A.1 and a junction that looks like the one in Figure A.3. This second junction has 2 lanes in each direction of the avenues and 6 lanes in each direction on each street. The phases are still orthogonal and are executed in the same order every time. The junctions are the *Junction 2-3* and the *Junction 2-6* respectively congruent with the number of lanes on the avenues and streets.

A.2 Traffic generation

Mahattan 10 and 5, and Brickwall 10

For all networks bigger than a single network the previously mentioned AVTIVITY-GEN and DUAROUTER was used with different population sizes to create routes and thereby traffic demands. For the networks Manhattan 10 and Brickwall 10, the area east of Avenue 6 was designated the living zone where there are 10 times more people living than working. West of Avenue 6 was the working zone with the re-

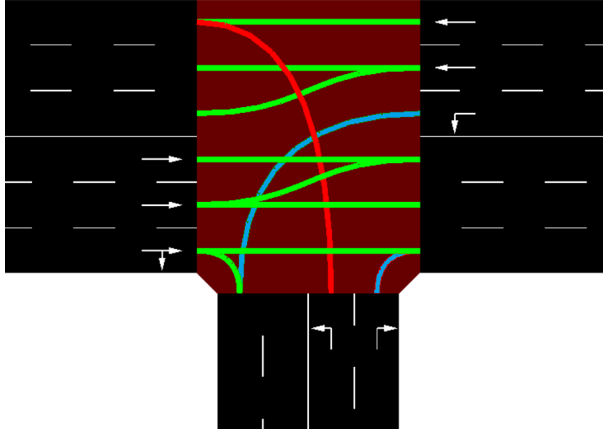


Figure A.2 The phases used in the junctions in the Brickwall network. If the junction is rotated 180 degrees the phases stay true with the junction.

versed relation. For the Manhattan 5 network this break point of relation between work and living numbers was at Avenue 2. There is a small stochastic element between sets of simulation - the probability to own a car. The same routes were used in the same set of simulations, so the same plot in the results has no stochastic element within itself.

Junction 2-3 and Junction 2-6

For the single junction networks there is no viable way of generating traffic with a population file. Instead traffic with a Poisson distributed arrival rate was created since this was simpler. The total traffic volume were the same in both the junctions throughout the set of simulations. However, because of the difference in the number of lanes the sum of critical lane volumes was different. The incoming traffic was produced at equal intensity from all 4 directions and the turning ratio was the same (left, straight, right) = $(1/4, 1/2, 1/4)$, for all directions. Sum of critical lane volume were 1326 for the 2 by 3 lane junction and 1178 for the 2 by 6 junction. This was only the nominal value though. At different time steps this was changed to simulate different traffic demands throughout the day. The relative change in traffic demand is shown in Table A.1. Vehicles had a probability of being created until time 10000 and the simulation was finished when no vehicles remained in the network. To investigate fairness and scalability these networks were also simulated with a shorter sensor length.

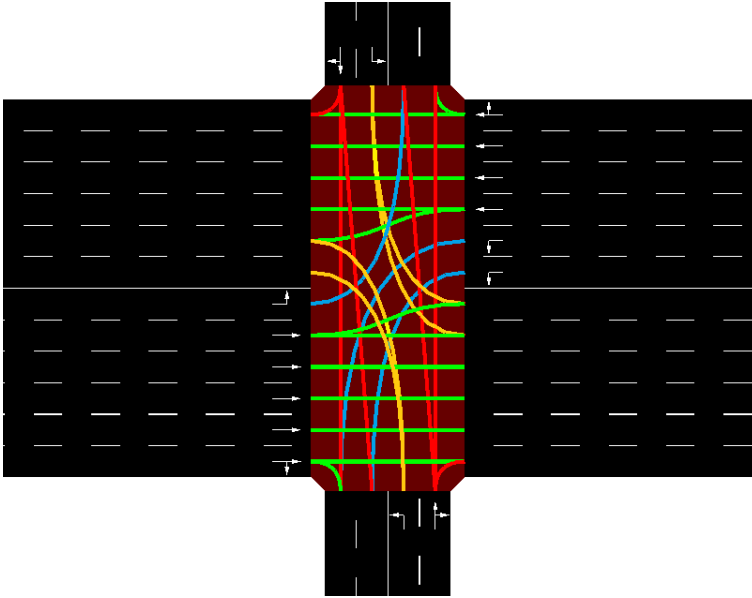


Figure A.3 The phase setup of the 2 by 6 lanes junction used for simulation. The phase execution order is always the same.

Table A.1 The different modifiers at the different times as compared to the starting intensity.

Time (s)	1500	3000	6000	9000
Modifier from nominal traffic arrival intensity	0.5	1	1.2	0.996

B

Optimality results

The results from the simulations on the Manhattan 10 network are presented in figures B.1, B.2, and B.3; the results from the Manhattan 5 network in figures B.4, B.5, and B.6; the results from the Brick-wall 10 network in figures B.7, B.8, and B.9; and the results from the Junction 2-3 and Junction 2-6 in figures B.10 and B.12 respectively. The results from the single junction networks are shown in figures B.11 and B.13. The results are summarised in table 3.2. Box plots from 39 different simulation sets of the Manhattan 5 - 2500 are presented in figures B.14, B.15, and B.16.

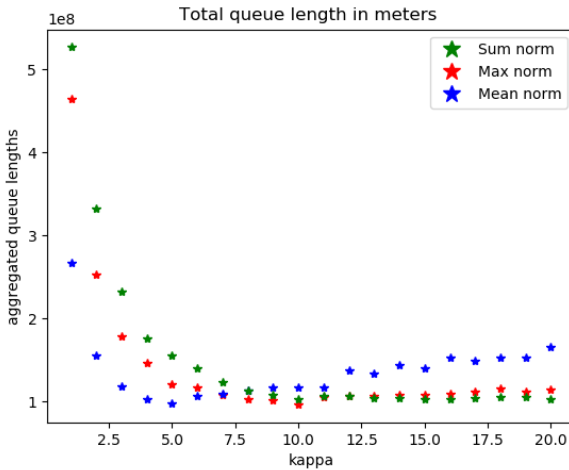


Figure B.1 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 10 - 10000 network with a lane area detector covering the whole road.

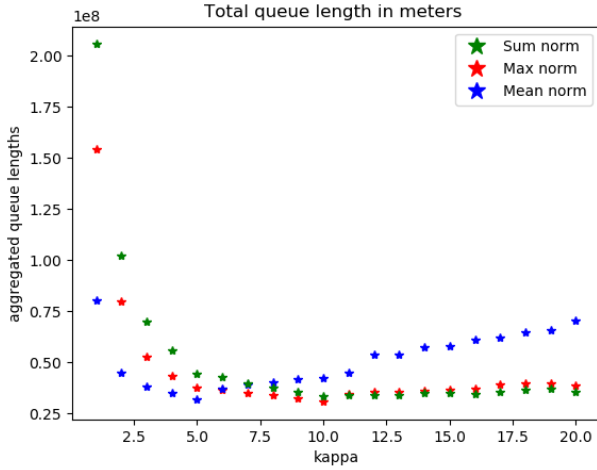


Figure B.2 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 10 - 5000 network with a lane area detector covering the whole road.

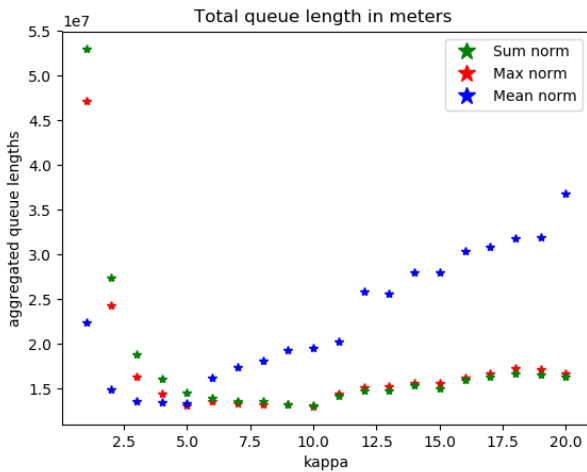


Figure B.3 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 10 - 2500 network with a lane area detector covering the whole road.

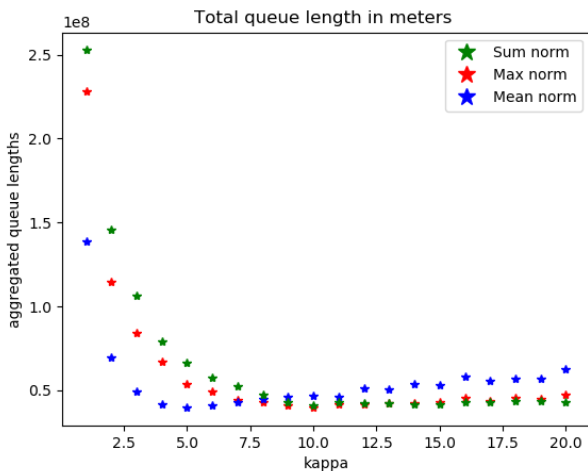


Figure B.4 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 5 - 5000 network with a lane area detector covering the whole road.

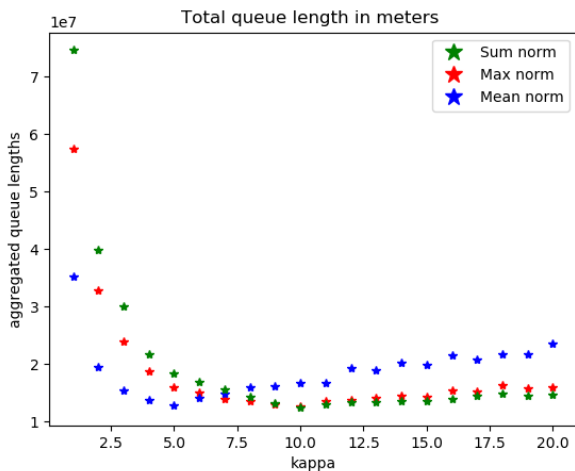


Figure B.5 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 5 - 2500 network with a lane area detector covering the whole road.

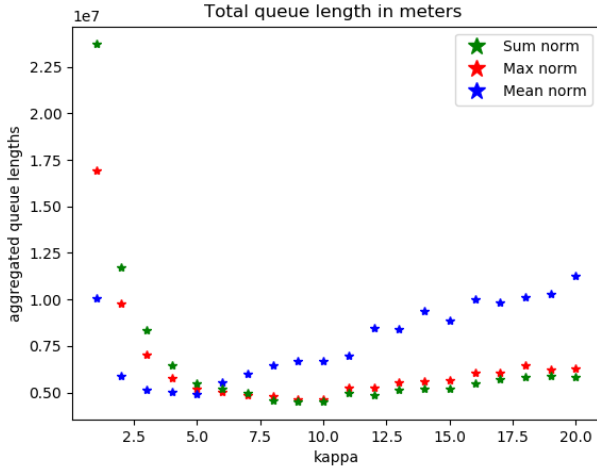


Figure B.6 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Manhattan 5 - 1250 network with a lane area detector covering the whole road.

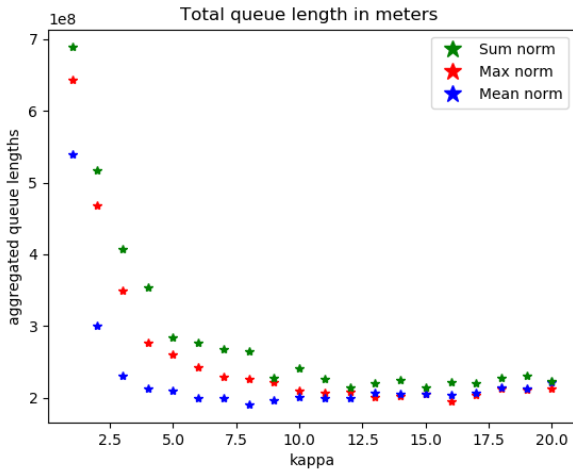


Figure B.7 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Brickwall 10 - 10000 network with a lane area detector covering the whole road.

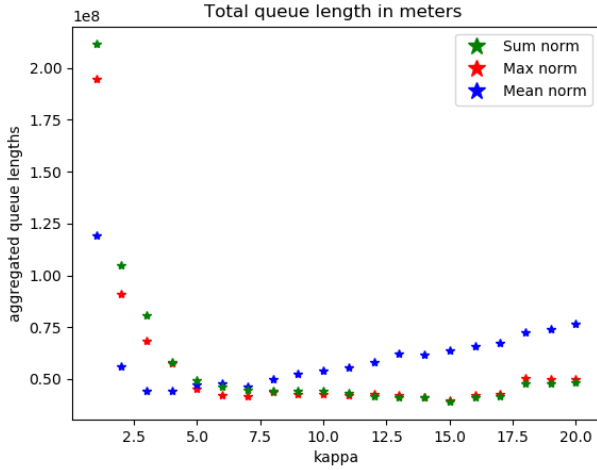


Figure B.8 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Brickwall 10 - 5000 network with a lane area detector covering the whole road.

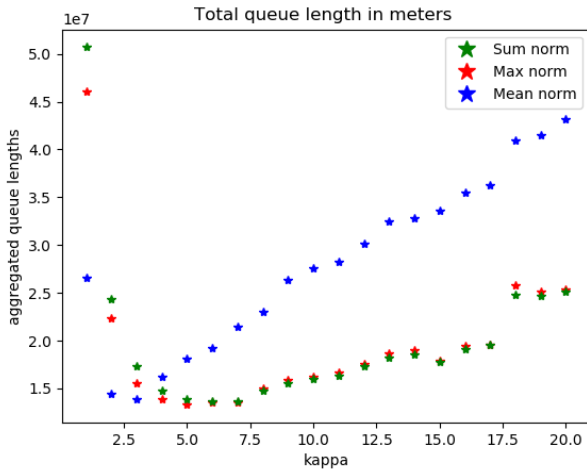


Figure B.9 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Brickwall 10 - 2500 network with a lane area detector covering the whole road.

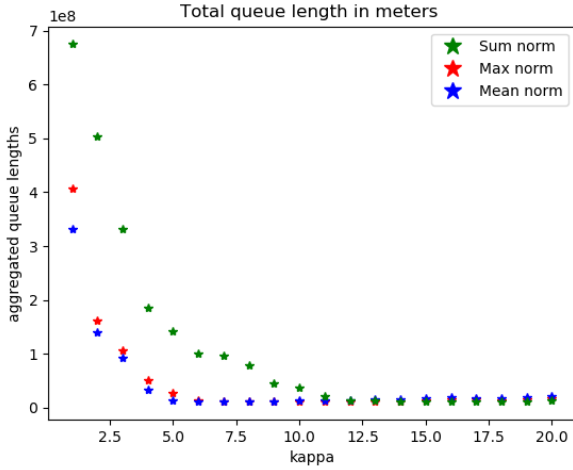


Figure B.10 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Junction 2-3 network with a lane area detector covering the whole road, and a sum of critical lane volume of 1326.

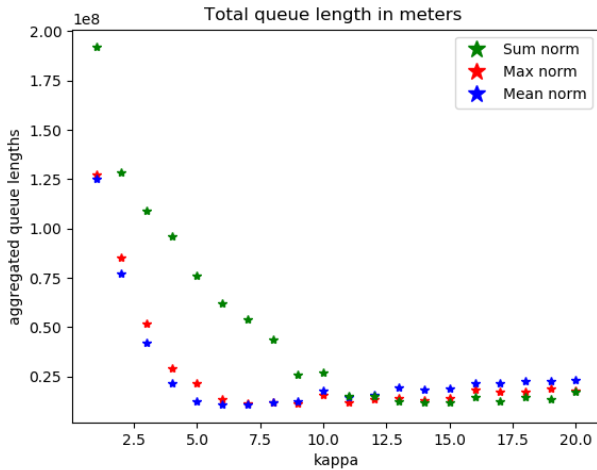


Figure B.11 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Junction 2-3 network with a lane area detector 80 meters long, and a sum of critical lane volume of 1326.

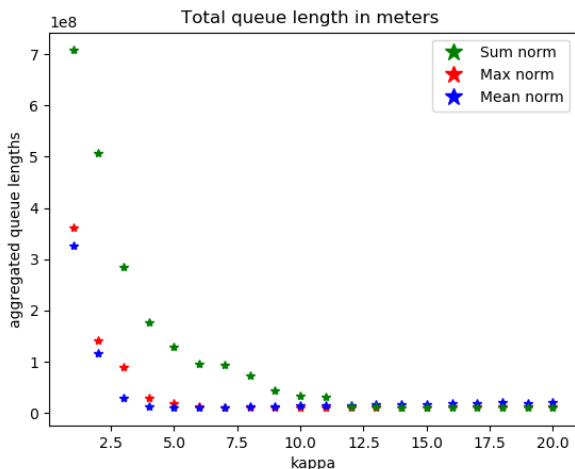


Figure B.12 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Junction 2-6 network with a lane area detector covering the whole road, and a sum of critical lane volume of 1178.

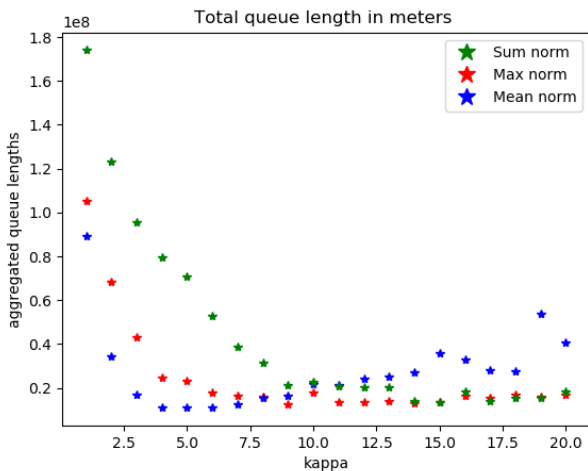


Figure B.13 The aggregated queue lengths for values of $\kappa \in [1, 20] \in \mathbb{N}$, and for the 3 different norms. These are the results from the Junction 2-6 network with a lane area detector 80 meters long, and a sum of critical lane volume of 1178.

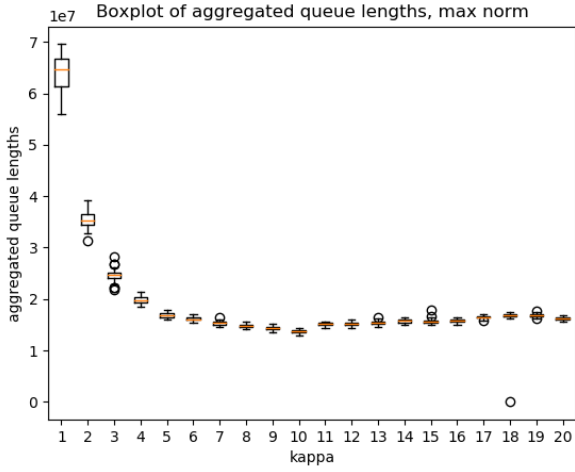


Figure B.14 Box plot from 39 runs of the Manhattan 5 - 2500 network with the max norm.

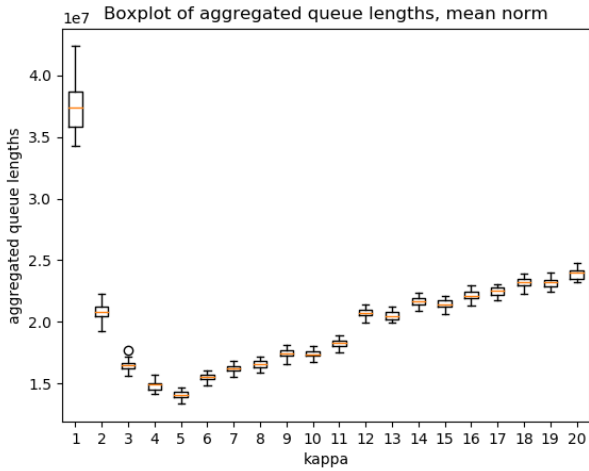


Figure B.15 Box plot from 39 runs of the Manhattan 5 - 2500 network with the mean norm.

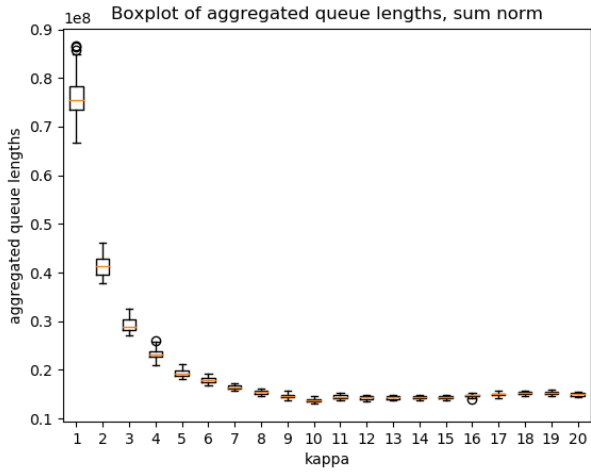


Figure B.16 Box plot from 39 runs of the Manhattan 5 - 2500 network with the sum norm.

C

Fairness results

The fairness results from the simulations on the Manhattan 10 network are presented in figures C.1, C.2, and C.3; the fairness results from the Manhattan 5 network in figures C.4, C.5, and C.6; the fairness results from the Brick-wall 10 network in figures C.7, C.8, and C.9; and the fairness results from the Junction 2-3 and Junction 2-6 in figures C.10 and C.11 respectively, and in figures C.12 and C.13 with shortened sensor length. The results are summarised in tables 3.3 and 3.4.

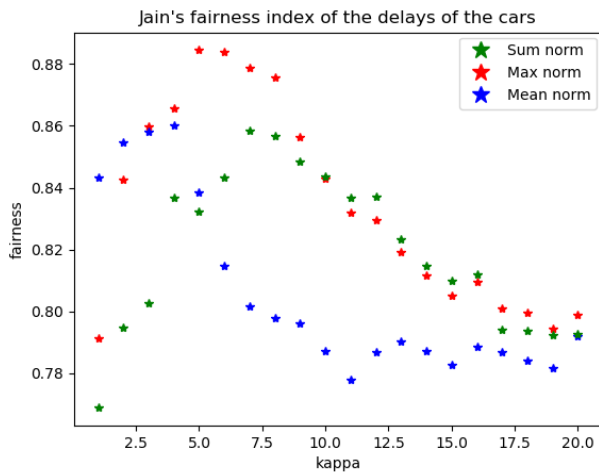


Figure C.1 The Jain's fairness index for the Manhattan 10 - 10000 network.

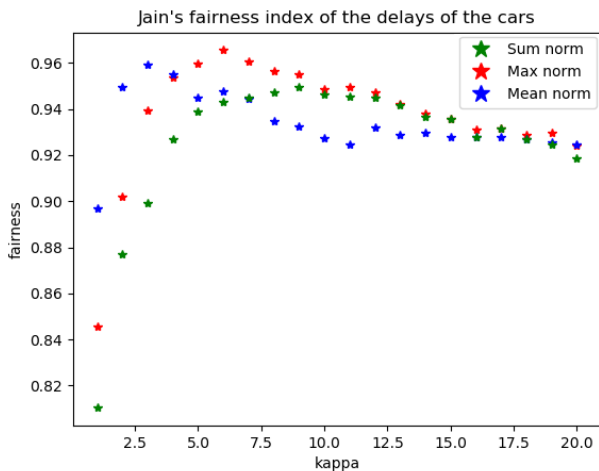


Figure C.2 The Jain's fairness index for the Manhattan 10 - 5000 network.

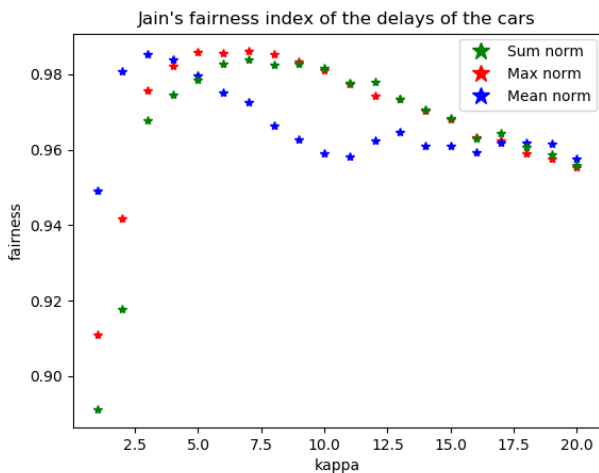


Figure C.3 The Jain's fairness index for the Manhattan 10 - 2500 network.

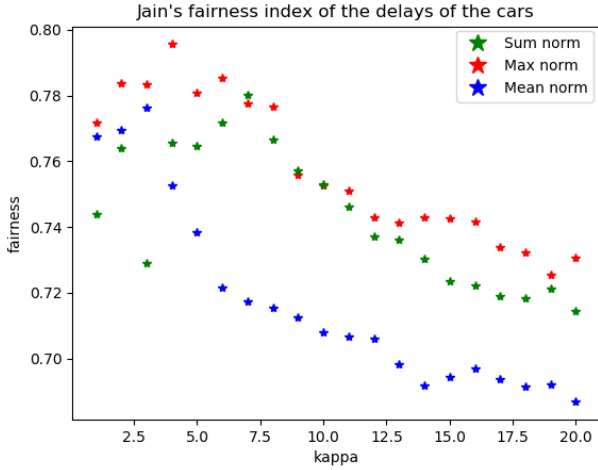


Figure C.4 The Jain's fairness index for the Manhattan 5 - 5000 network.

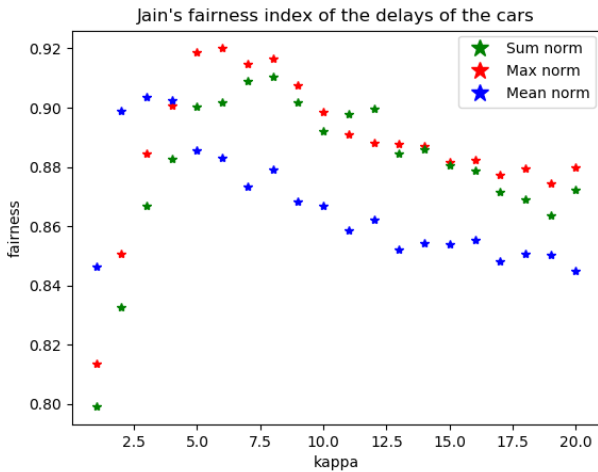


Figure C.5 The Jain's fairness index for the Manhattan 5 - 2500 network.

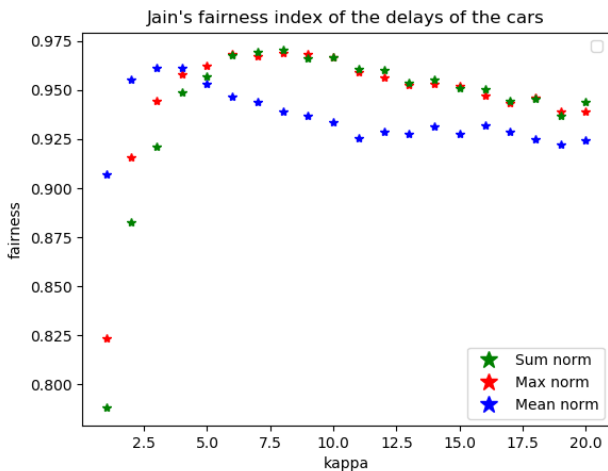


Figure C.6 The Jain's fairness index for the Manhattan 5 - 1250 network.

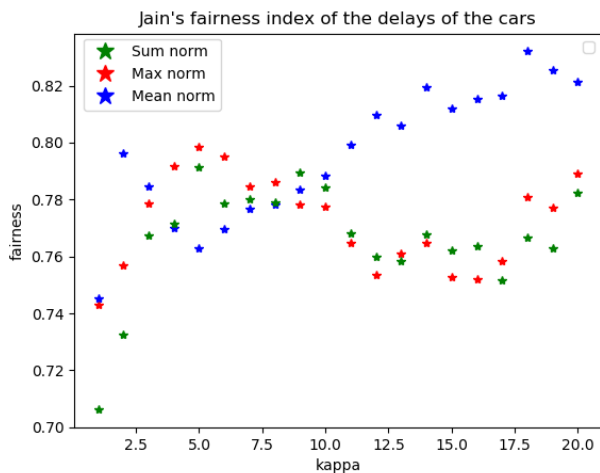


Figure C.7 The Jain's fairness index for the Brickwall 10 - 10000 network.

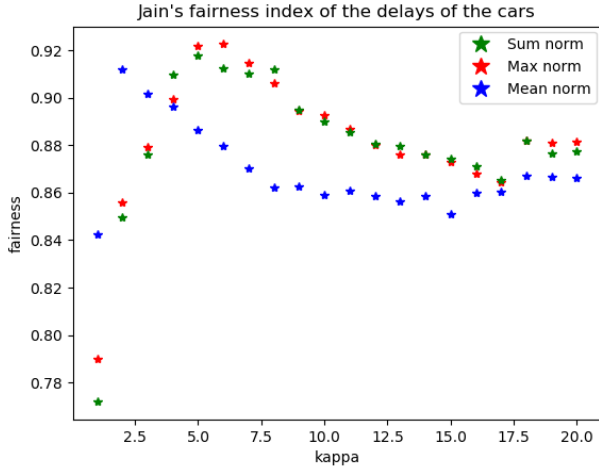


Figure C.8 The Jain's fairness index for the Brickwall 10 - 5000 network.

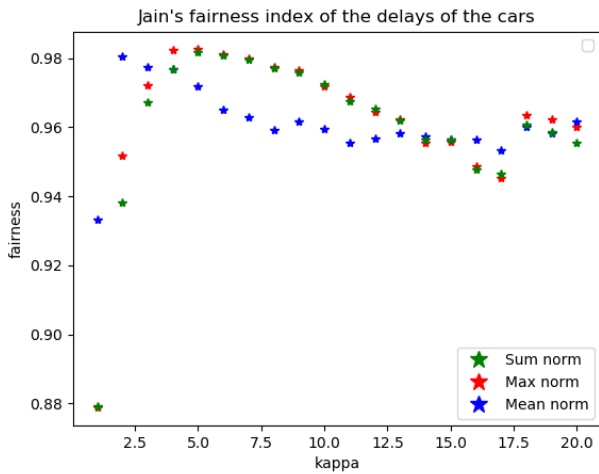


Figure C.9 The Jain's fairness index for the Brickwall 10 - 2500 network.

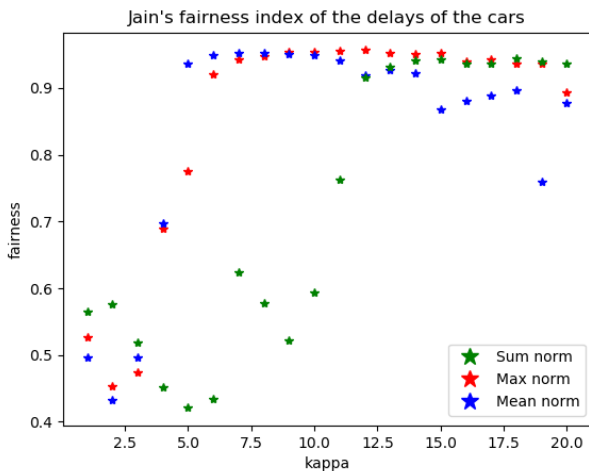


Figure C.10 The Jain's fairness index for the Junction 2-3 network.

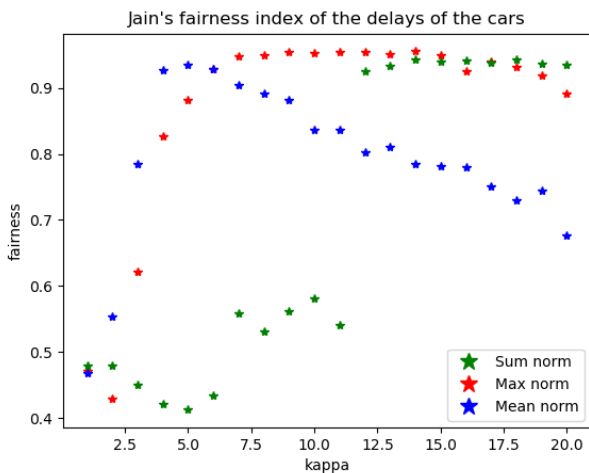


Figure C.11 The Jain's fairness index for the Junction 2-6 network.

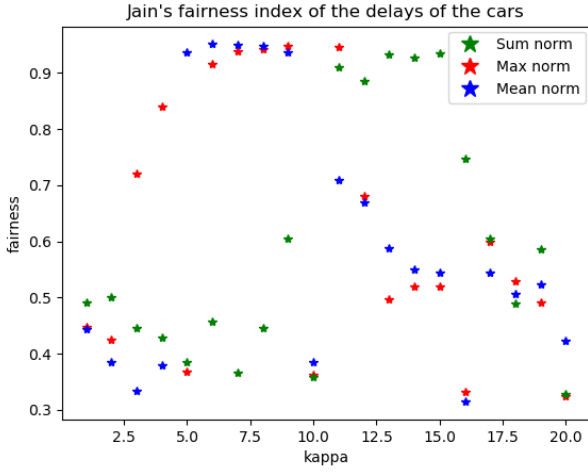


Figure C.12 The Jain's fairness index for the Junction 2-3 network with the shortened queue length.

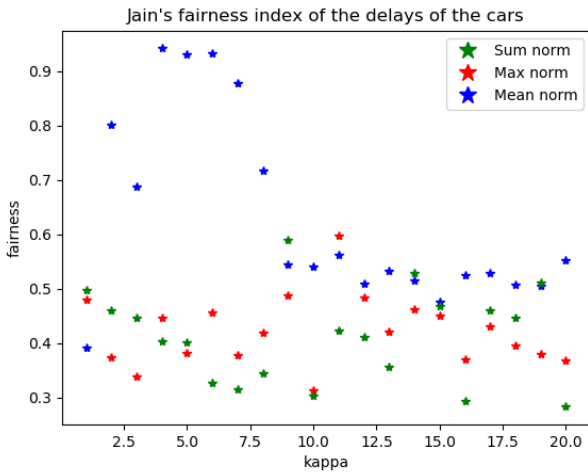


Figure C.13 The Jain's fairness index for the Junction 2-6 network with the shortened queue length.

D

Stability results

In this appendix the aggregated queue lengths of the MATLAB simulations are presented.

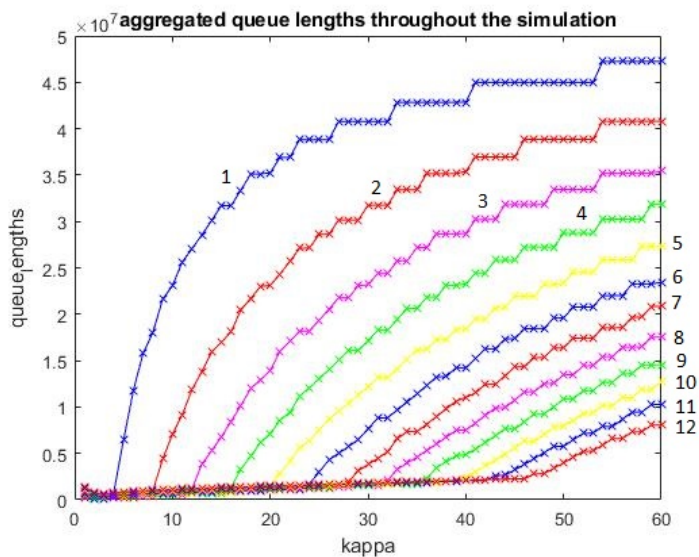


Figure D.1 Aggregated queue lengths in the 1 junction network with no start-up lost time. Numbers next to the coloured lines are lane detector lengths in vehicles.

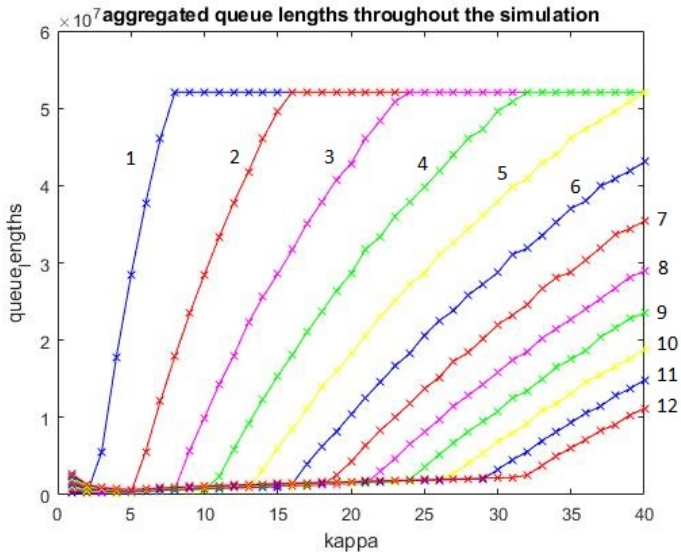


Figure D.2 Aggregated queue lengths in the 1 junction network with 2 seconds start-up lost time. Numbers next to the coloured lines are lane detector lengths in vehicles.

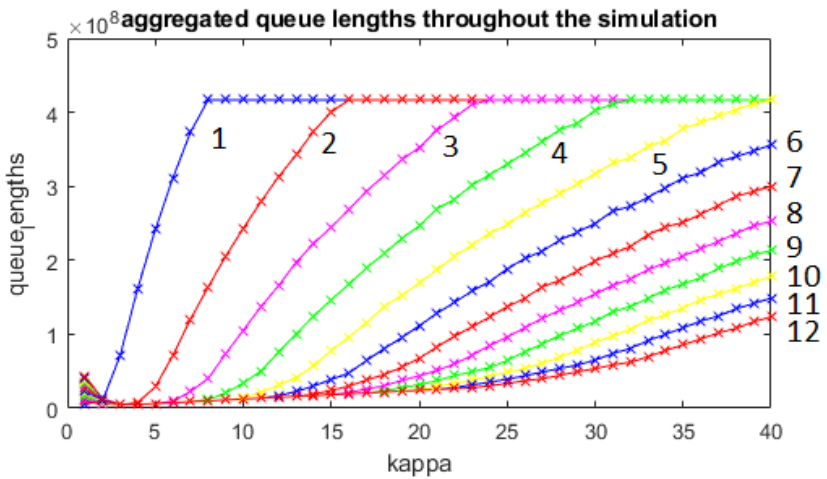


Figure D.3 Aggregated queue lengths in the simple 2 by 2 junction network. Numbers next to the coloured lines are lane detector lengths in vehicles.

E

Performance

In this appendix more results from the performance comparison are presented. First, in Figure E.1 the summarised results are presented again. After that, in the rest of the figures, some of the data points in Figure E.1 are displayed. The data points displayed are selected based on the best performance from the different controllers and some of the worst performance.

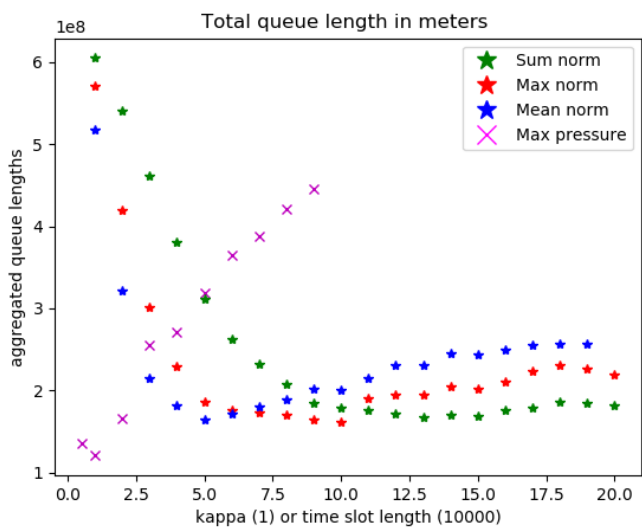


Figure E.1 Aggregated queue lengths from the simulations.

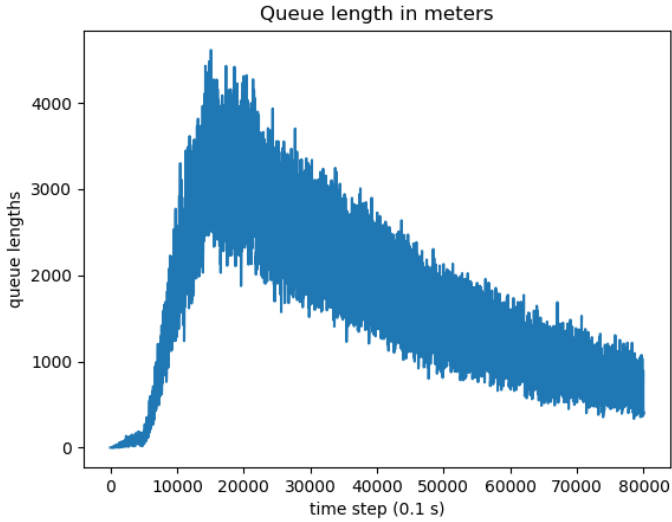


Figure E.2 Queue lengths over time during the simulation. Back pressure controller. Slot time 10 seconds.

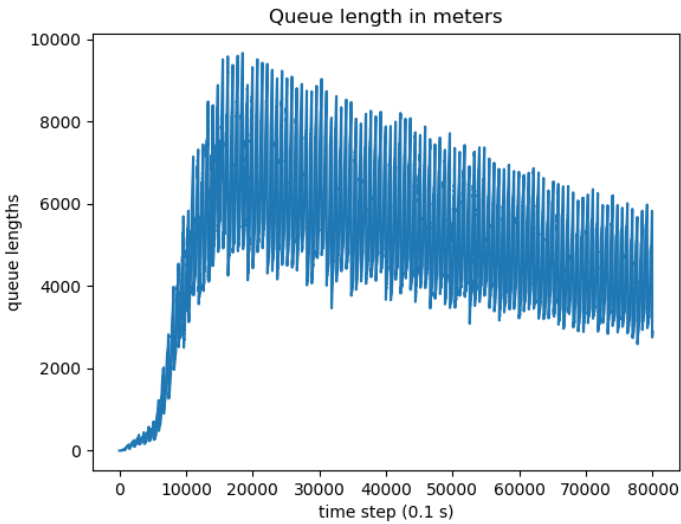


Figure E.3 Queue lengths over time during the simulation. Back pressure controller. Slot time 70 seconds.

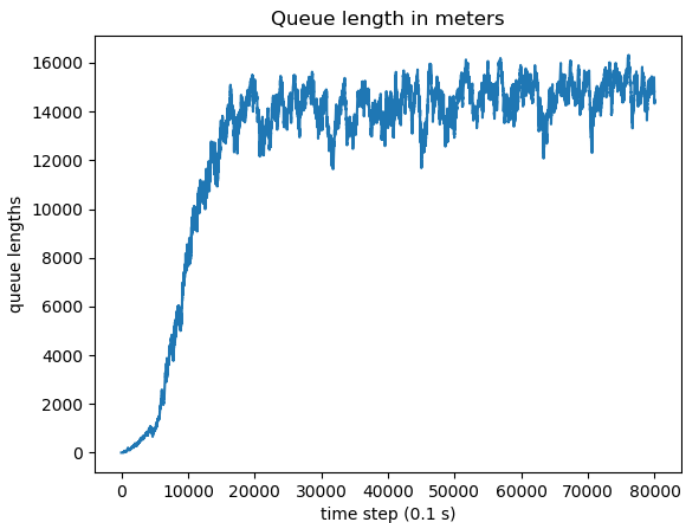


Figure E.4 Queue lengths over time during the simulation. Proportional allocation controller, max norm, $\kappa = 1$.

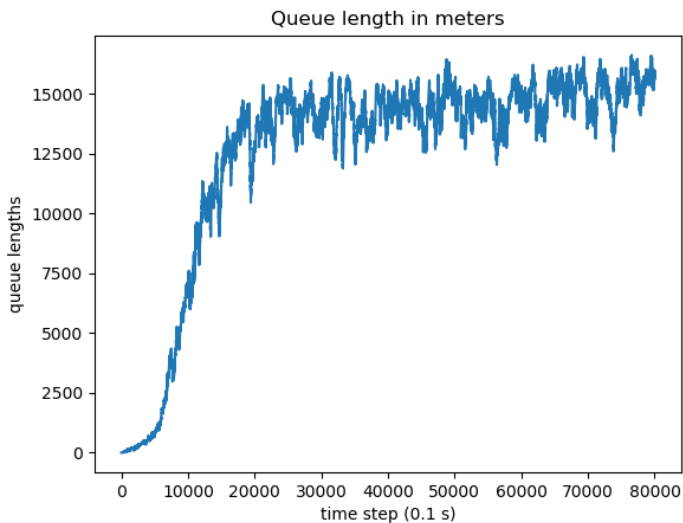


Figure E.5 Queue lengths over time during the simulation. Proportional allocation controller, mean norm, $\kappa = 1$.

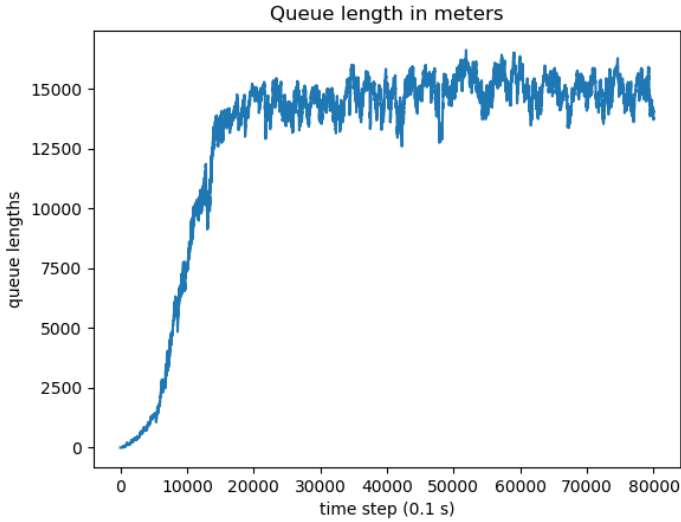


Figure E.6 Queue lengths over time during the simulation. Proportional allocation controller, sum norm, $\kappa = 1$.

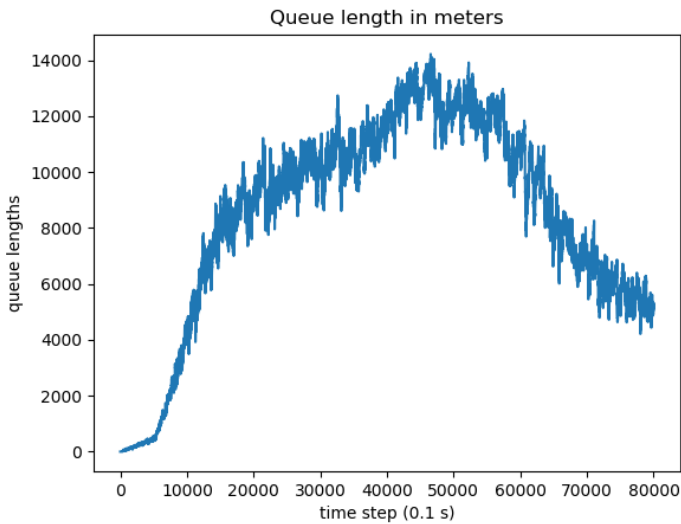


Figure E.7 Queue lengths over time during the simulation. Proportional allocation controller, max norm, $\kappa = 10$.

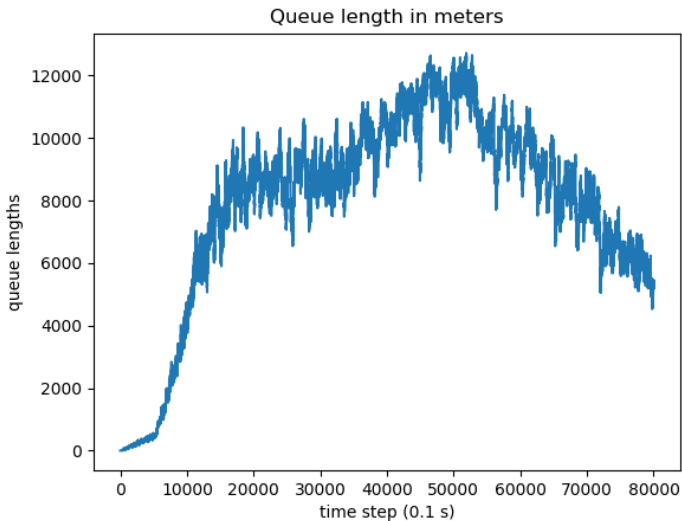


Figure E.8 Queue lengths over time during the simulation. Proportional allocation controller, mean norm, $\kappa = 5$.

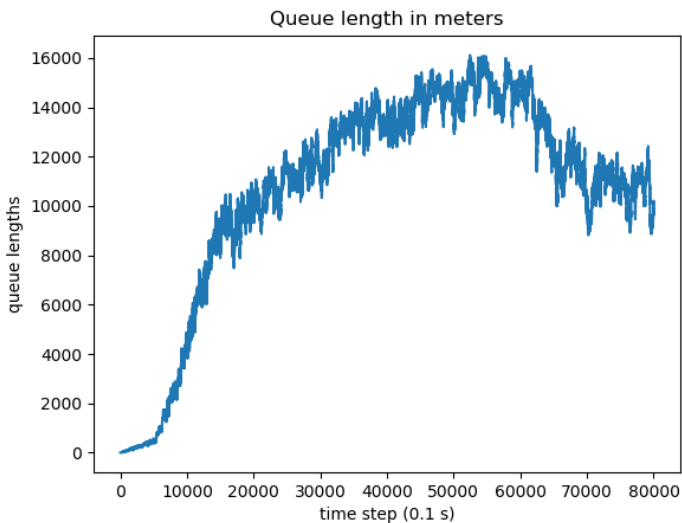


Figure E.9 Queue lengths over time during the simulation. Proportional allocation controller, sum norm, $\kappa = 10$.

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS
		<i>Date of issue</i> September 2018
		<i>Document Number</i> TFRT-6067
<i>Author(s)</i> Simon Paulsson		<i>Supervisor</i> Gustav Nilsson, Dept. of Automatic Control, Lund University, Sweden Giacomo Como, Dept. of Automatic Control, Lund University, Sweden Anton Cervin, Dept. of Automatic Control, Lund University, Sweden (examiner)
<i>Title and subtitle</i> Tuning Feedback-Based Traffic Signal Controls		
<i>Abstract</i> <p>A growing problem in many cities is that the traffic demand fluctuates heavily during the day and that it grows larger as the years go by. This calls for, and the technology to easier measure traffic enables, more efficient traffic light controls that can handle a greater traffic demand but also be adaptive. Centralised control of traffic light might, even though it can be both adaptive and efficient, be costly to install and expand due to the required communication between junctions. Decentralised control however, might be cheaper to install and maintain while still being adaptive and efficient. This thesis explores the Proportional Allocation controller with regards to its tuning parameter under conditions imposed by e.g., network size and topology, and queue sensor length. Two aspects of its performance were measured: optimality and fairness. In addition to the original controller two modifications are proposed and evaluated. It was found that when simulating, smaller networks could be used to approximate bigger ones, and that the queue sensor length is of critical importance not to be too small. The proposed modifications were found to perform similarly to the original one even though it in principal should not. The results herein should give a good starting point of the optimal value of the tuning parameter in a real-world implementation.</p>		
<i>Keywords</i>		
<i>Classification system and/or index terms (if any)</i>		
<i>Supplementary bibliographical information</i>		
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 1-84	<i>Recipient's notes</i>
<i>Security classification</i>		