MASTER'S THESIS

Adaptive and Scalable Urban Traffic Control in an Indian Metropolis

ARS | Traffic & Transport Technology

Author:

Tommi Jaakko Cristian Sisso

Supervisors:

Dr. ir. A. Al-Hanbali

Dr. J.C.W. van Ommeren

Ir. W. van Nifterick

MSc. Anna Oblakova

Author

T.J.C. (Tommi) Sisso

University

University of Twente

Master programme

Industrial Engineering & Management

Specialization

Production and Logistics Management

Graduation Date

29.8.2016

Graduation Committee

Dr. Ir. A. Al-Hanbali
University of Twente

Dr. J.C.W. van Ommeren
University of Twente

Ir. W. van Nifterick
ARS Traffic and Transport Technology BV

MSc. Anna Oblakova
University of Twente

Management Summary

Motivation

The increasing urbanization and the simultaneous growth of economic prosperity and population cause huge traffic problems in the developing countries. The congestion of traffic taxes the economy and the society through increased pollution and decreased productiveness. Individual people and companies suffer alike, insufficient road capacity contributes to the already high traffic mortality rates and commuting takes a considerable amount of daily free time. As the traffic infrastructure is updated slower than the number of road users increases, in near future the problems are only going to get worse.

The only short term approach to relieve the problems related to traffic is to increase the utilization rate of the existing traffic infrastructure. ARS Traffic & Transport Technologies is currently installing a traffic light system in the city of Patna in India. Patna suffers from problems caused by severe traffic congestion and traffic lights are installed to help relieve the adverse effects. A well-controlled traffic light network can help utilize the existing road infrastructure better, thus increasing the available capacity.

Research objective

The objective of this research is to develop a control algorithm that makes traffic light control decisions:

Design an adaptive network management algorithm that aims at minimizing the average travel time in the network that is applicable to the city of Patna

The goal of the algorithm is to reduce the average delay vehicles face in the system. The algorithm has to be scalable in the number of intersections in the network, robust against communication and intersection infrastructure failures, stable in the sense of providing network optimal throughput under heavy traffic conditions, and able to make real-time online control decisions.

Traffic and traffic infrastructure in Patna

The traffic in Patna behaves differently than the traffic in the western countries. Firstly, the vehicular composition of the traffic is different, consisting of mostly two and three-wheeled vehicles. Secondly, the drivers behave differently. Lane markings are often not followed and the concept of a queue of vehicles is often ambiguous. This phenomenon, called heterogeneity, places limitations on the traffic control approach. The problem of measuring and controlling the traffic in Patna is hampered also by the limited capabilities of the detection cameras used in Patna. The system can only detect the presence of vehicles, i.e. occupancy, close to the stop line of the intersection. In traditional traffic control approaches detectors are placed upstream to provide predictive measurements and are able to recognize and count individual vehicles.

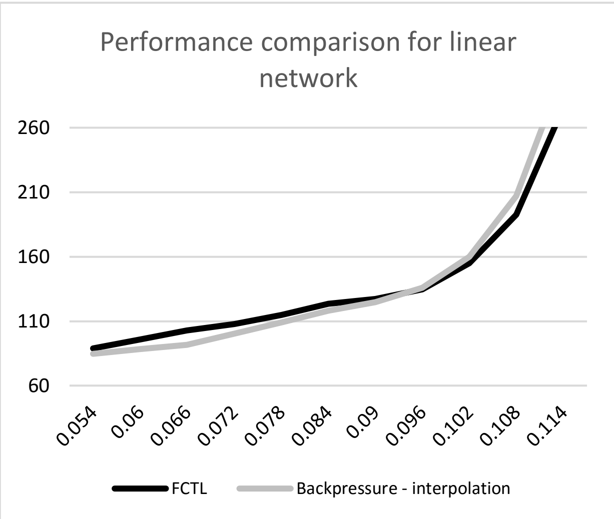
Traffic control algorithm

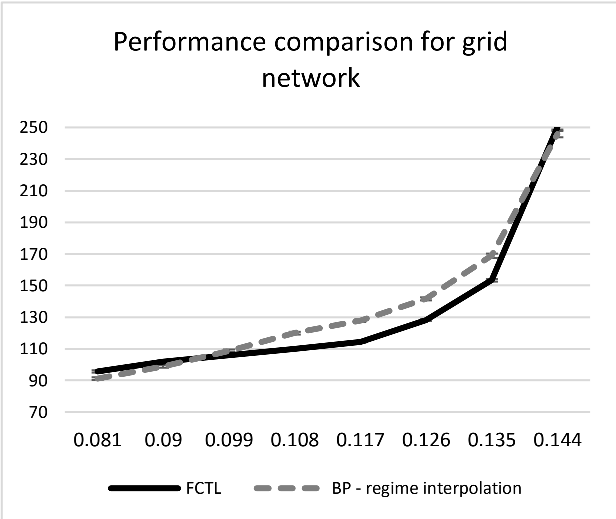
We propose a control method based on a stochastic routing algorithm. The algorithm computes for each service phase the difference between the queue lengths at the intersection and the queue lengths at the target downstream intersections. This difference is interpreted as a “pressure” that the intersection wants to discharge during the next traffic cycle. The algorithm is decentralized meaning that each intersection in the network has authority over the local traffic control decisions. Beneficial network behavior arises implicitly as a result of simple decision policies being applied on local level. Each junction is controlled by an intersection agent, an individual controller that also has the responsibility of gathering and processing the sensory information produced at the intersection. The intersection controllers communicate with each other to share the measurement data. By doing this, the system is able to build more complete picture of the prevailing traffic load in the system. In addition, the state of downstream intersections is used by the control algorithm to reduce the myopia of the decentralized control.

We solve the problem of heterogeneity of the traffic and the limited capabilities of the detection cameras by proposing a measurement algorithm that allows the system to estimate the queue lengths at the intersection from historical measurements. The algorithm solves the problem related to the short-sightness of the detector cameras by detecting the end of a moving queue and by translating this to an estimation of the number of vehicles that were queueing when the green light was initiated. The algorithm uses multiple measurements from prior traffic light cycles to construct an estimate of the current queue length. Smoothing is used to remove short term variations and capture long term trends in the estimates.

Results

We conduct simulation experiments to measure the average delay of vehicles in the network when traffic lights are controlled by our traffic control algorithm. We follow current literature in using a fixed cycle controller as a benchmark. We compare the performance of our algorithm to the performance of this fixed controller. We demonstrate a simulation-based procedure of determining near-optimal values for the control parameters and establish that the values depend on the prevailing intensity of the traffic.





We simulate two different networks, a linear chain of intersections and a square grid-like network. We find that in the linear network and for light and medium traffic intensities our control algorithm performs up to 10.8 % better than a fixed cycle controller. For very heavy traffic intensities, our controller performs worse than a fixed controller, with the overall average delay reduction of 1.77 %. In the grid network, our control algorithm fails to decrease the average delay. In average, our control algorithm causes 4.3 % longer delays in the network. The algorithm performs better than the fixed controller for three traffic intensities, two light traffic demands, and the heaviest traffic demand.

In addition to analyzing the average delay and the performance, we establish that the key reason our algorithm fails to improve the average delay for the heavy traffic demands is the lack of proper information. The virtual loop detectors used in Patna are located on the stop lines of the intersections and fail to provide good quality information. The algorithm we propose for estimating queue lengths using these detectors fails to accurately estimate the number of vehicles for high traffic intensities. We also establish that with additional investments in the detector infrastructure the performance of the control algorithm can be improved significantly.

Recommendations for implementation

The control algorithm proposed in this research serves a good starting point in developing an urban traffic control system in Patna. Given an amount of total cycle time in an intersection the algorithm can take current traffic conditions into account and distribute the green time to the service phases. We have shown in this research that with certain types of network and traffic demand our control algorithm can perform better than a fixed green time allocation.

We recommend that the system is implemented as a multi-agent system. By nature, our control algorithm is decentralized, and the most logical way to implement it is to build an autonomous controller for each of the intersections. In this research, we have proposed a structure for the intersection controller and the control algorithm proposed in this research serves as the decision making module of the controller.

Before the actual implementation of the urban traffic control system the intersection controller needs to be supplemented with some additional functionality. The intersection controller has to be able to analyze the required length of the service cycle. The queue length estimation algorithm proposed in this research can be used to do this. The intersection controller should communicate their needs for the cycle length to each other. The communication can be used to create sub-networks with shared cycle length to improve synchronization. In addition, the intersection controller should communicate with other nearby intersections to compute the optimal offsets values to improve the synchronization and coordination between the traffic lights.

Preface

I am proud to present you my MSc. thesis "*Adaptive and Scalable Urban Traffic Control in an Indian Metropolis*". This research marks the end of my studies at the University of Twente. The time in Enschede and in Den Haag has sure been the most interesting, stressful and rewarding two years of my life. This research is the end of a six-month struggle of trying to figure out how to guide notoriously chaotic traffic flows through a large network. Although initially the project seemed very unorthodox for Industrial Engineering, it was satisfying see how, in the end, I could use the skills I acquired during my studies.

This research would not have been finished without the support of many people. My special thanks go to my company supervisor Wim van Nifterick. He gave me the chance to work on the thesis at ARS. And although he is often very busy, he was always supportive, able to squeeze out some time for a short meeting and always ready to challenge my ideas. I'd like to thank also the numerous other people at ARS, who have introduced me to the world of traffic management. Most notably, I would like to thank Edwin van Endhoven the fellow thesis writer, who, being more experienced with traffic management, helped me to get started with the thesis and helped to reflect the choices, decisions and conclusions I made during this project. In the end, I liked my time and the people at ARS so much, that, I am proud to say, I start there with my first real job after this project.

My special thanks go also to my supervisor Ahmad al Hanbali. First, he helped me find a thesis assignment and end up at ARS in the first place. After this, during the thesis his support and feedback was invaluable. He was always available for a meeting and his guidance helped me through the rather slow start of the project. I would also like to thank the other members of graduation committee J.C.W van Ommeren and Anna Oblakova for their constructive feedback during the later parts of the project.

Obviously, the most special thanks go to my girlfriend Britt. Without her support and motivation, you would not be reading this research today. Her support was especially invaluable during the "rough start" of my studies when I faced problems choosing the direction I wanted to go with my Master's. Then, during this project her support and motivation kept me going. Thanks for bringing me to this wonderful country; I am looking forward to our future here. Lastly, I would like to thank my parents for understanding my choice to move to the Netherlands and follow a study abroad. I would also like to thank them for helping me solve some resource related problems that inherently come with such a decision.

Tommi Sisso

Table of Contents

1 Introduction

Traffic lights are a key part of everyday urban life and the key method of trying to align the capacity of the limited traffic infrastructure with the ever increasing amount of traffic. This problem is especially pronounced in developing countries, where population is growing and urbanizing rapidly but investments in the infrastructure are lacking. Traffic management, realized through the control and coordination of traffic lights in urban areas, is the key method to alleviating congestion and the related problems. This research supports ARS Traffic & Transport Technologies, a Dutch company currently installing a network of traffic lights in the city of Patna in India, to control the traffic lights effectively. We propose an adaptive algorithm that can be used to make real-time decisions concerning the amount of green light shown to different roads at the intersections of the network.

In this chapter, we briefly introduce ARS Traffic & Transport Technologies and discus the problem topic, traffic light management and consider the problem motivation and background. After this, in Section 1.4 we identify the key problem that we solve in this research. In Section 1.5, this problem is translated into a more concrete research goal. Later, in Section 1.6 we define the scope and key limitations of the research, and finally in Section 1.7 we use the Managerial Problem Solving Approach to define the problem solving approach that defines the structure of this thesis. We also define the research questions that lead to the fulfilment of the research goal and that are answered in this research.

1.1 Company Introduction

ARS T&TT (Traffic & Transport Technologies), later ARS in this report, based in The Hague, The Netherlands, provides consultancy, products, design, development, operation and financing on intelligent transport systems (ITS). The company was founded in 1998 by Dr. Jan Linssen, and currently has around 200 employees in the Netherlands and in India. The expertise areas of the company are, for example, strategic, tactical, and operational consultancy on ITS issues, optimization of existing traffic information and traffic management centers, dynamic travel information for road traffic and public transport, dynamic guidance systems for bus stations and car parks, speed limit enforcement, access control systems and enforcement, environmental zone enforcement, traffic planning systems, road pricing and toll systems, fleet management, and international monitoring and operation of ITS systems. (ARS T&TT, 2016)

1.2 Problem topic

The key idea of traffic lights is to separate conflicting traffic movements in the time domain. Without traffic signalization conflicting traffic flows occupy the same space at the same time. In these cases, the driver must apply known traffic rules to determine the priorities of traffic movements. When traffic flows do not conflict, as in Figure 1.1, traffic signalization is not necessary. But in the situation shown in Figure 1.2, when traffic flows do conflict, signalization can improve both traffic safety and the flow of the traffic.

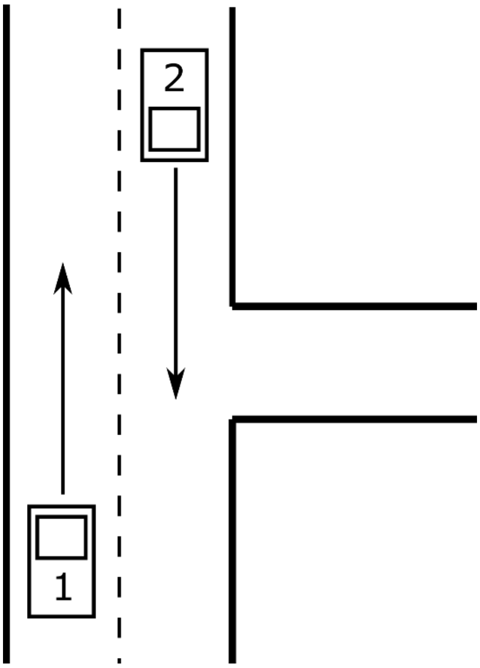


Figure 1.1 Non-conflicting traffic flow

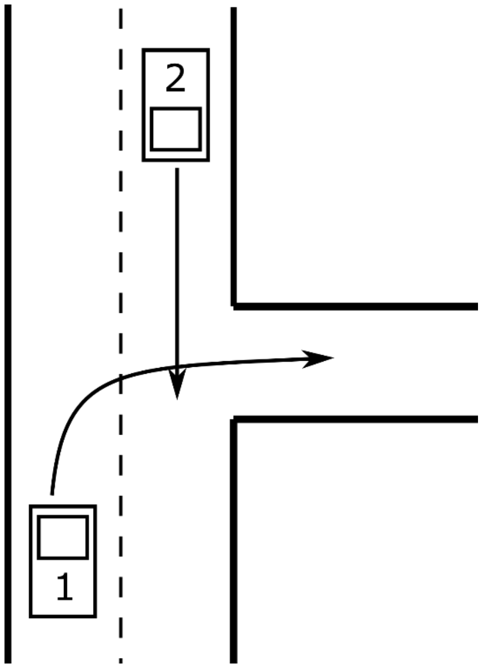


Figure 1.2 Example of conflicting traffic movements

Without traffic control, as in Figure 1.2, an increased risk of an accident is involved. In order to make a decision to either wait or drive, driver 1 must estimate the distance to driver 2, the speed of driver 2, and the time he or she needs to clear the conflicting area of the junction. Wrongly estimating any of these factors and making a decision based on them can lead to an accident, say, driver 1 decided to cross the junction but driver 2 crashes into driver 1. Assuming both drivers respect traffic signalization, this problem can be encountered by installing traffic lights that separate the conflicting traffic flows like in Figure 1.3.

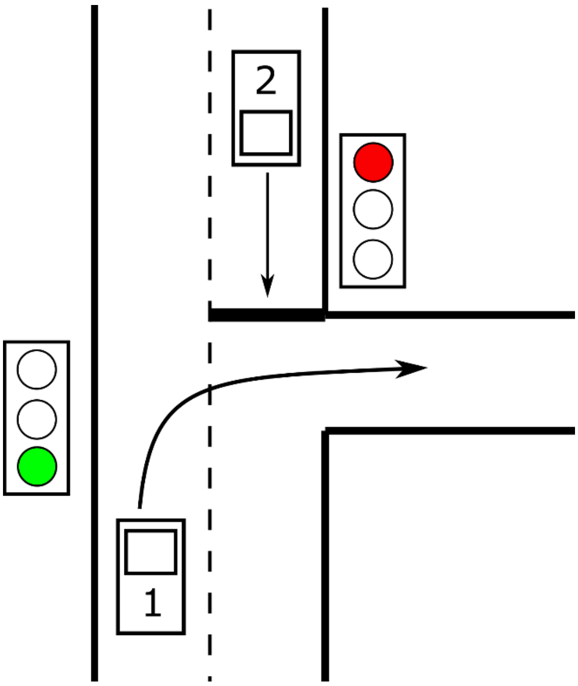


Figure 1.3 Example of traffic light control

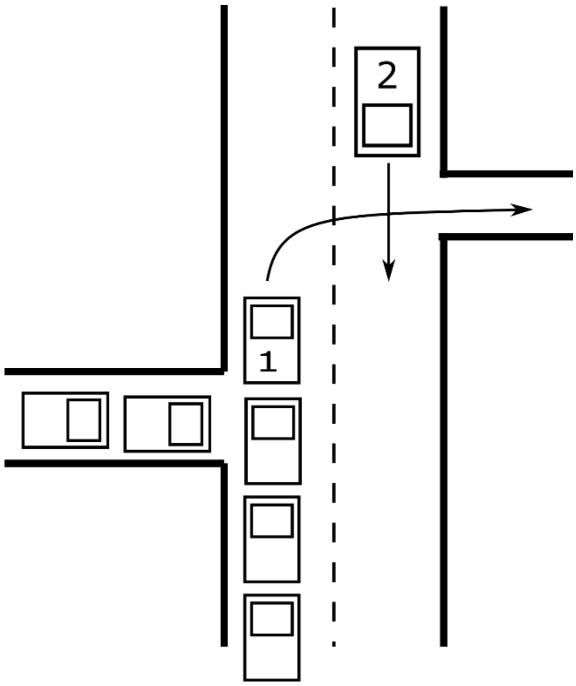


Figure 1.4 Blocking of intersections due to queues

In addition to reducing the risk of an accident, traffic signalization can increase the smoothness of the traffic flow, and prevent big jams from occurring in parts of the traffic network. Consider again driver 1 that wants to turn to the right. According to traffic rules, he or she must yield to the vehicles in the driving direction 2. If there is a steady flow of cars from this direction driver 1 might need to wait a long time. If, at the same time, there is a steady flow of vehicles that want to drive to the same direction as driver 1, a queue of vehicles will form. If this queue becomes too long it can block other intersections, causing again growing queues. See Figure 1.4 for an example of such situation. Such phenomenon can be problematic in urban areas, where adjacent intersections are in close proximity. If the queues of vehicles grow long enough on multiple routes, the situation can enter a state called a *gridlock*, meaning that all possible traffic movements are blocked by a queue that, in turn, is blocked by another queue that was formed by blocking.

The goal of traffic control is to choose the control parameters of the traffic lights such that some performance indicator, for example, travel times, traffic throughput, and total delay, of the traffic, is optimized. These parameters include, for example, cycle time (referring to the time taken to cycle through showing green light to all directions and starting a new control cycle), split timing (referring to the amount of green time given to any particular direction during a cycle), and offset (referring to the time difference between junctions in initiation of certain phases of the cycle). The traditional traffic control approach is to create a fixed cycle that the traffic light follows. Junctions controlled by traffic lights are assumed to be isolated, i.e. not affected by the traffic lights at other junctions, and the parameters of the fixed cycle are computed off-line based on historical traffic data. The modern approach is to control the full network of traffic lights, with the aim of optimizing a network-wide performance indicator, such as vehicle throughput or average waiting time in the system. Also, recent interest is in adapting the control of the network according to the present and predicted traffic conditions.

1.3 Problem Motivation and Background

Traffic in India is notoriously bad. Although vehicle ownership rate is considerably lower than the same rate in western countries, the traffic conditions are considerably worse. Indian traffic suffers from severe congestion. For example, in Kolkata, a major Indian metropolis, during peak hours the average traffic flow speed in central parts of the city can be as low as 10 km/h. This congestion causes massive delays. In addition to the delays and economic pressure on the commerce and industry, pollution and accident related problems are also magnified. Combined with the fact that the cars are often old and of bad quality, standing in a traffic jam and driving inefficiently slowly due to congestion produces high amounts of various air pollutants. The Indian metropoles are among the most polluted cities in the world (Salomi, 2015). The bad state of the traffic also causes a high number of traffic-related accidents. In 2009, 126.900 people died on the Indian roads, translating to 14 deaths per hour, and 0.13 deaths per 1000 inhabitants, a figure more than twice as high as in western countries such as Sweden or the United Kingdom. (Singh, 2012)

The problem of Indian traffic situation is made worse by a number of factors. Firstly, the population of India is growing rapidly. The annual growth rate is approximately 1.4 %. Moreover, the population of urbanized areas is growing even faster, at an annual rate of 3 %. This means that the population densities and areas of the already congested areas are going to grow in the near future. The number of urban dwellers is expected to reach 500 million, and 35 % of the total population by 2021. Secondly, the vehicle ownership rate (motorized vehicles per 1000 inhabitants) grows concurrently with the population growth. In the large cities, the ownership rate grew from 132 to 286 vehicles per 1000 inhabitants, in a decade between 1999 and 2009. Consequently, this combination translates to explosive growth of the total number of vehicles. Moreover, most of these vehicles are concentrated in the already congested metropolises, with 35 % of the total vehicles compared to only 11 % of the total population. (Singh, 2012)

The increase of both urban population and the vehicle ownership rate is going to make the already existing traffic problem only worse. This is already indicated by the fact that traffic-related fatalities are increasing faster than the population of India. The problem is also fostered by the governmental policies that promote private vehicle ownership over public transport. This policy is meant to support domestic industry and causes the modal split of traffic to be skewed towards private vehicles such as motorbikes, motorized three wheelers, and cars.

1.3.1 City of Patna

City of Patna, the capital and largest city of the state of Bihar in Eastern-India, shares many of the characteristics of other big Indian cities. The annual population growth is approximately 3 % or 32.5 % in ten years between 2001 and 2011. According to the latest census in 2011, the population of the city was 1.684.297, and the metropolitan area of Patna had a total population of 2.046.652.

In addition to the rapid population growth, Patna also exhibits explosive growth in the number of motorized vehicles. The vehicle ownership rate grew from 192 to 302 vehicles per 1000 inhabitants during the time between 1999 and 2009. Combined with the fast population growth means an increase of 114 % in the total number of vehicles. In 2012 there were 675.952 registered motorized vehicles in Patna. During this time period, the traffic infrastructure was not improved accordingly. (ARS T&TT, 2014)

1.4 Problem Identification

ARS is currently involved in a project initiated by the Bihar Urban Infrastructure Development Corporation (BUIDCO). The aim of the project is to relieve the bad traffic situation in the city of Patna. ARS has proposed to install a network of 88 traffic lights, the related management system, CCTV cameras, and a Central Control Room. The locations of these traffic light controlled intersections are shown in Figure 1.5. Currently, 66 traffic lights are installed and operating in fixed mode.

Managing the traffic lights beyond local fixed control is very difficult. It is not explicitly clear, how the control decisions in one junction affect the arrivals, and thus the optimal decisions, in another junction. Thus, it is also not clear, how the control decisions at individual traffic lights affect the total travel time in the network. The problem is made more difficult by the topology of the network. There is no clear single direction that vehicles follow through the network, in which case some simple controls could apply, but there are numerous junctions where arterial streets meet. Also, the network of the traffic lights is situated in the center of a large city, thus, the demand for transportation generated within the network is significant. Another reason for the difficulty of controlling a network of traffic lights is the inherent stochasticity of the traffic. The most common example of this is the intra-day temporal variation in the traffic flow, i.e. the occurrence of rush hours. This variation is usually connected to people commuting to and from work and thus follows a typical morning-afternoon cycle. However, even within this cycle, there are shorter period variations. Firstly, as discussed above, traffic control decisions at other junctions define the stochastic vehicle arrival process at downstream junctions. In addition, within a traffic flow, the behavior of vehicles is stochastic. This is due to the individual decisions by the driver, which lead to variation in for example acceleration, speed, rule discipline, and gap-acceptance.

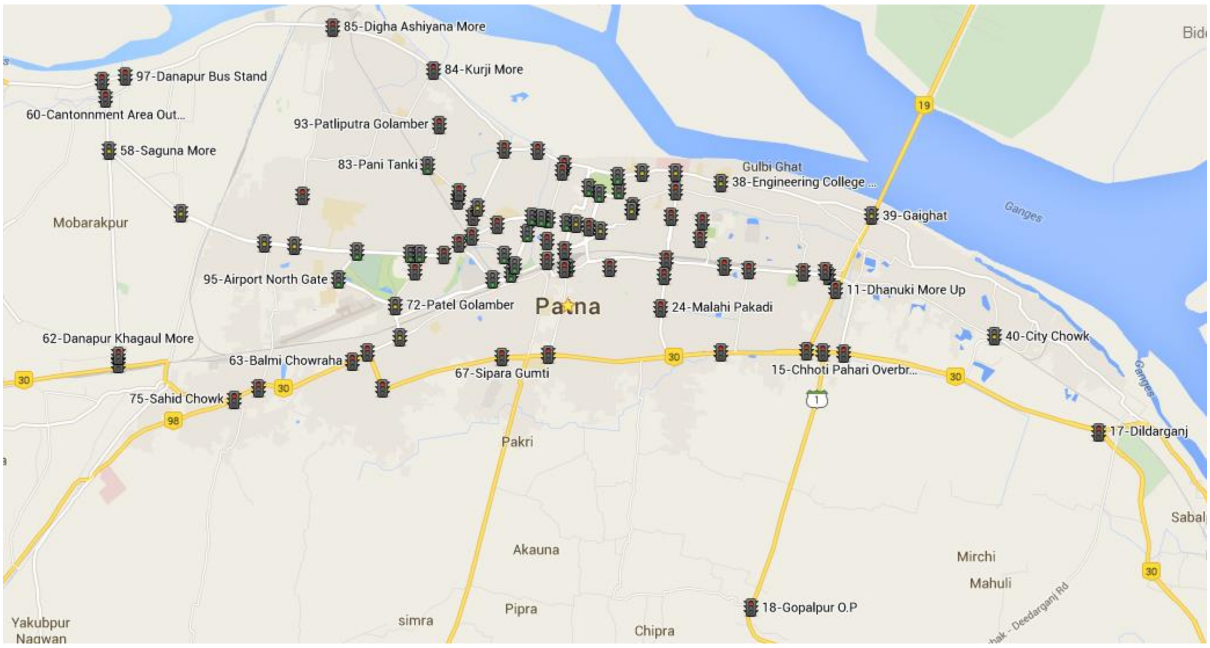


Figure 1.5 Network of traffic lights in the city of Patna

Making the control decisions are always related to some *objective function*. In the case of ARS, the goal is to minimize *the average travel time* through the network. We define travel time as the sum of link delay and intersection delay a driver faces when using the network. Intersection delay refers to the time spent queueing in an intersection, and link delay describes the time taken to traverse a road that connects two adjacent intersections. Local fixed control is unable to optimize this performance indicator on the network level.

The intersections are equipped with pattern recognition based camera detectors. They can be used to make real-time measurements of the actual traffic conditions and provide data for the decision making. However, local fixed control cannot utilize this information in the optimization of the traffic flows. Consequently, the core problem of the research becomes

> **Making the traffic control decisions is difficult without a proper network tool that considers the network as a whole.**

1.5 Research Goal

ARS currently lacks a dynamic way of controlling the traffic signals in its traffic light network. There are some existing and well-known solutions to adaptively control networks of traffic lights. However, all these solutions are closed, proprietary systems. Thus, ARS wants to develop an in-house solution to the traffic control problem. We want to solve this problem by designing an (adaptive) control algorithm that can set the control parameters of traffic lights in the network based on the current traffic situation. Consequently, the goal of this research is:

> **Design a(n) (adaptive) network management algorithm that aims at minimizing the average travel time in the network (and additionally preventing gridlocks) that is applicable to the city of Patna.**

1.6 Research Scope and Limitations

One of the simplifying assumptions we make to reduce the size of the problem instance is to assume that the phases of the intersections are given and fixed. This is reasonable assumption as these phases are most of the time standard among all traffic control implementations. Also, they are determined using traffic engineer's expertise and reflect the long-term junction capacity and safety requirements.

In addition to this, we are not going to research on the adequacy of the traffic infrastructure. We consider the network of traffic lights given and design an algorithm to control it. We also make use of the existing sensors and their functionality. The intersections are fitted with detection cameras on the stop lines. There are no traditional induction loops installed in Patna. We design our control as to use only the existing sensor inputs. In the end of this thesis we briefly discuss and analyze the adequacy of the current sensor infrastructure.

The algorithm should be able to control the traffic lights in real-time, or close to real-time. This means that the computation time to find the control decisions must be "short". The length of this time depends on the hierarchical level of the decision. For example, network-wide tactical decisions that are made few times an hour can afford to have computation times of around 15 minutes. On the other hand, local control decisions that are made possibly every second must have accordingly quick computations.

Due to the time constraints of this project and the complexity of the installation process, we cannot test the algorithm in the actual traffic network. Because of these limitations, we evaluate the performance of the system through simulations. Moreover, due to possibly excessive simulation times, these simulations can be run only on small scale networks. Also, we can only simulate the control algorithm against some basic control rules because the existing commercial solutions are closed third party systems and thus we cannot simulate their relative performance and functionality.

1.7 Plan of Approach

In order to solve the research problem, we follow the well-known Managerial Problem Solving Method. This method is designed to solve action/design problems, meaning that something needs to be changed in order to reach certain goal. Our research is clearly tackling an action problem, as we need to design a control method, in order to be able to control the network of traffic lights in Patna. This problem solving method also lays the foundation for the structure of this report.

1) Problem Identification
2) Planning the problem-solving process
3) Analyzing the problem
4) Generating alternative solutions
5) Choosing a solution
6) Implementing the solution
7) Evaluating the solution

This introduction chapter completes the first two steps. We have established the problem background, context, and identified the core problem that needs to be solved. We also determine the project goal and the relevant scope. The second step is completed by this discussion of the problem-solving approach. This involves recognizing the necessary steps and information that is needed to solve the problem we identified. This step also contains generating the research questions and the subsequent structure of the thesis.

After this, we complete steps 3 and 4. First, we analyze the problem by considering the traffic situation in Patna. We also present the network configuration and the existing traffic controls. We shortly consider the current control mechanisms and establish its performance. In addition, we collect system requirements that our control method has to fulfil from ARS. After this, we conduct a literature review to map possible network control approaches that could be used in the design of the controller. We search for existing methods and consider their usefulness in the traffic control problem of Patna.

After we have established the current situation, the requirements and constraints of the controller, and we have mapped possible existing approaches, we can choose the methods we want to implement in the controller. We propose an adaptive control algorithm that can be used to control the network of the traffic lights. We also implement a prototype of this algorithm as a module to a traffic simulation software. We end this research by considering the performance of the proposed control mechanisms. We conduct simulation studies to measure some performance indicators, such as waiting times and system throughput and compare them to the performance values of simple non-adaptive controls, such as fixed traffic lights.

1.8 Required Information and Research Questions

Context Analysis

The first information that we need concerns the traffic situation in Patna. Because the proposed algorithm must consider the specific situation in Patna, we analyze the traffic and the existing traffic infrastructure. These serve as the design requirements, limitations, and constraints to the traffic control method. Consequently, in chapter 2 of the thesis we want to answer the following research question:

1) **What is the current traffic situation in Patna?**
    a. **What is the performance of the current system?**
    b. **How does traffic in Patna differ from the western traffic?**
    c. **What is the network that is managed?**
    d. **What traffic data is available?**
    e. **What system requirements are set by ARS?**

Literature Review

In Chapter 3, we map existing solution approaches that can be used to control a network of traffic lights. This involves first defining the traffic control problem and the network control problem. After this, we conduct a structured literature review in order to search for existing solution approaches in our problem context. Given the requirements and the limitations we identified in chapter 2, we also want to analyze the usefulness of the existing approaches in solving our traffic control problem.

2) **What solution approaches exist in literature to control a network of traffic lights?**
    a. What is traffic network control?
    b. What approaches can be used to control a network of traffic lights?
    c. What are the pros and cons of these approaches?
    d. Which control approach fits best the traffic situation in Patna?

Algorithm Design

In chapter 4, we develop the network control algorithm to control traffic lights of Patna. We base this tool on the methods we identified in the literature review of chapter 3. This part constitutes the key contribution of this research. We take the identified solutions from the literature and apply and modify them to the specific control problem of Patna. Consequently, in this part of the thesis we answer the following research question:

3) **How can we control the network of traffic lights in Patna?**

Performance Evaluation

The last part of this research evaluates the performance of the proposed approach. Due to the short duration of this project, we are not able to implement the network controller in Patna, and thus, we resort to simulation to provide insight on the performance and functioning of the algorithm. Moreover, due to the large size of the full network, we are limited to simulating smaller subnetworks. Because the control algorithm uses a variety of input values, such as parameters chosen by the user, or the sensor inputs, we are interested in the sensitivity of the algorithm to these inputs. This leads to following research question

4) **How does the proposed approach perform?**
   a. **How can we simulate the functionality of the proposed algorithm?**
   b. **How sensitive is the algorithm to its input parameters?**
   c. **How does the algorithm perform against simple fixed control mechanisms under some trial cases?**

We end the thesis by concluding the results and findings of the research. We consider possible directions for future research and provide recommendations on the actual implementation of the control algorithm.

2 Urban traffic network in Patna

In this part of the thesis, we consider the current traffic situation in the city of Patna. We argue also the usefulness of implementing an adaptive traffic control system. First, we discuss the general characteristics of the traffic and the network. We also refer to a traffic survey to evaluate the performance of the current system. After this, we discuss the topology of the urban road network and the traffic light network. This analysis supports us in evaluating possible control methods. Next, we analyze traffic video recordings to analyze the behavior of various vehicle classes in the traffic flow. In addition, we compare the traffic characteristics to those of developed countries. This is important as most existing solutions are developed for the western traffic and possibly assume a well behaving and structured traffic flow. We consider the traffic detectors used to measure the traffic in the city of Patna. The data provided by these instruments forms the main input to our control system and, thus, possibly limit the kind of approach that can be used. We end the chapter by collecting system requirements set by ARS. Later in this thesis, we use these requirements to evaluate the usefulness of possible solution approaches. Choosing an approach that fits the requirements and preferences of the problem owner is more likely to receive organizational support and be implemented.

2.1 General description of the network

The city of Patna is suffering, like many Indian metropolitan areas are, of problems related to severe traffic congestion. Due to urbanization, general population growth, and rise in living standards and income, the number of motorized vehicles in Patna has risen significantly. In ten years between censuses of 2001 and 2011, the number of motorized vehicles increased by 114%. Because the traffic infrastructure has not been improved at a similar rate, the traffic congestion has become a major problem. The situation has various adverse effects, such as increased travel times, increased accident rate, and pollution.

The traffic infrastructure of Patna has been shaped by being in a central location and by being the capital of the state of Bihar. Patna is located in the crossing of 3 major rivers, most notably the Ganges. These waterways provide good possibilities for freight transport. This, in combination with being the local commercial and industrial center, increases the traffic demand on the urban road network. As a state capital, Patna has good road connections to other cities. This means the surrounding region generates additional traffic load in the city of Patna. Moreover, a few national and regional highways pass through the urban area of Patna.

The urban road network of the city of Patna exhibits characteristics that worsen the current traffic situation. Firstly, the roads in the city area are relatively narrow. The distribution of road widths shown in Figure 2.1 indicates that the majority of the roads in the city of Patna are narrower than 7.5m. This is usually the width of a two-way road having one lane per direction and a narrow shoulder. Because roadside parking is very common in Patna, the roads are very narrow for the traffic load, which contributes to the occurrence of traffic congestion. Secondly, the Ganges river has a major impact on the network. Most arterial routes run parallel to the river, and there are no circular routes around the city. In addition, there are no arterial routes in the north-south directions. Thirdly, the limited number of river crossings creates significant local traffic demands around the bridges. This phenomenon is visible in Figure 2.2, in which the arterial routes are superimposed on the map of the city of Patna.

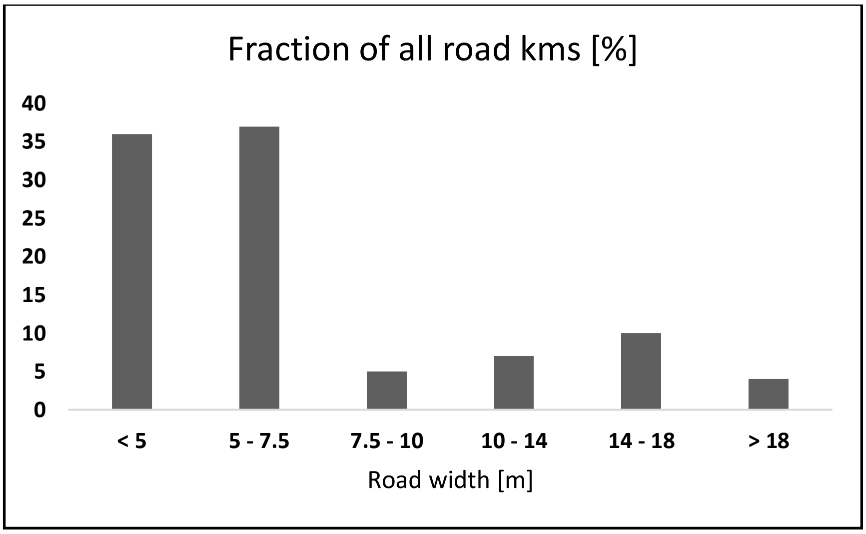


Figure 2.1 Distribution of road width in the city of Patna

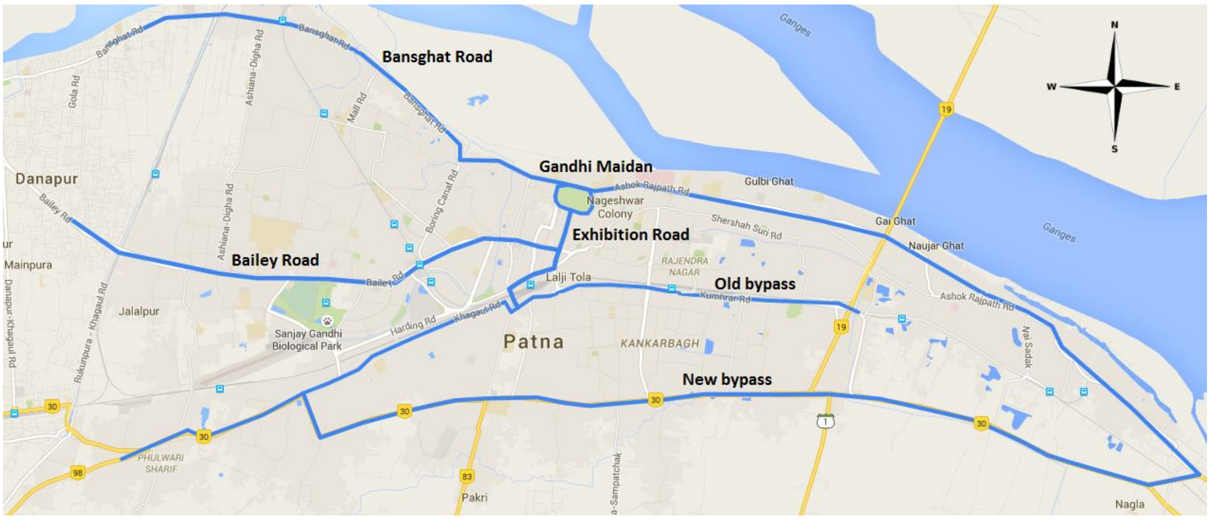


Figure 2.2 Arterial routes in the city of Patna emphasized in blue.

The main reason for traffic congestion is the intra-day temporal variation of the traffic demand. The problem is common to all urban areas in the world, in the long term the capacity of the road network can serve the daily traffic demand, but the majority of the actual traffic demand is lumped around dedicated peak-hours. Usually, and in Patna too, the peak-hours of the traffic occur in the morning, between 9-11 AM, when people commute to their work, and in the afternoon, between 6-7 PM, when these people return back to home from work. One way to represent the traffic demand variation is to report the so-called peak hour factor, which reports the fraction of the daily traffic demand that is realized during the busiest hour of traffic. If the traffic was spread equally over 24 hours, each hour would receive 4.2% of the daily traffic load. The actual peak hour factors in among a sample of 25 intersections in Patna vary between 8.2 % to 12 %. The average peak hour factor is 10 % (ARS T&TT 2014).

2.2 The performance of current system

As part of the initiation of the current project, ARS conducted a traffic survey to map the current traffic situation in the city of Patna. 150 intersections were visually observed for their level of congestion. A smaller subset of intersections, 25 junctions in total, was chosen for more accurate traffic counting.

The data of this survey, although limited, helps us establish the current state of the traffic network performance and the need of coordinated urban traffic control.

The initial set of 150 intersections was visually observed for their level of congestion by traffic engineers. They found that 45 % of these intersections were severely congested and another 38 % highly congested. Although it is unclear what qualifies as severely or highly congested in this classification, the result indicates that the majority of the intersections in Patna are operating at or near full capacity. The research also found that intersections with a traffic police guiding the traffic perform better than completely unsignalized intersections. This indicates that signalization is beneficial in improving the traffic flow, and we can expect to have similar results with traffic lights.

As a result of high levels of congestion, the intersections are suffering from the simultaneous reduction of vehicle throughput and increased waiting times. In general, this result is to be expected from an urban traffic system. Greenshields (1935) was first to established that on a highway the relation between the average speed of vehicles and the flow follow a parabolic expression. Later such relations have been derived between the flow and density of the traffic and between speed and the density of the traffic. Usually, these relations are reported in graphical form, called the *fundamental diagram,* shown in Figure 2.3**.** In recent literature, papers such as Geroliminis and Daganzo (2008) and Geroliminis and Sun (2010) have demonstrated that on macroscopic level urban traffic exhibits similar characteristics on the network level. Here, we present a very simplistic representation of the fundamental diagram as drawn by Woensel and Vandaele (2007). Although modern literature has established that the relation is not a simple parabola, like in our figure, and the curve has various discontinuity points, we use the representation in Figure 2.3 to argue the origins of the deterioration of the traffic performance in Patna.

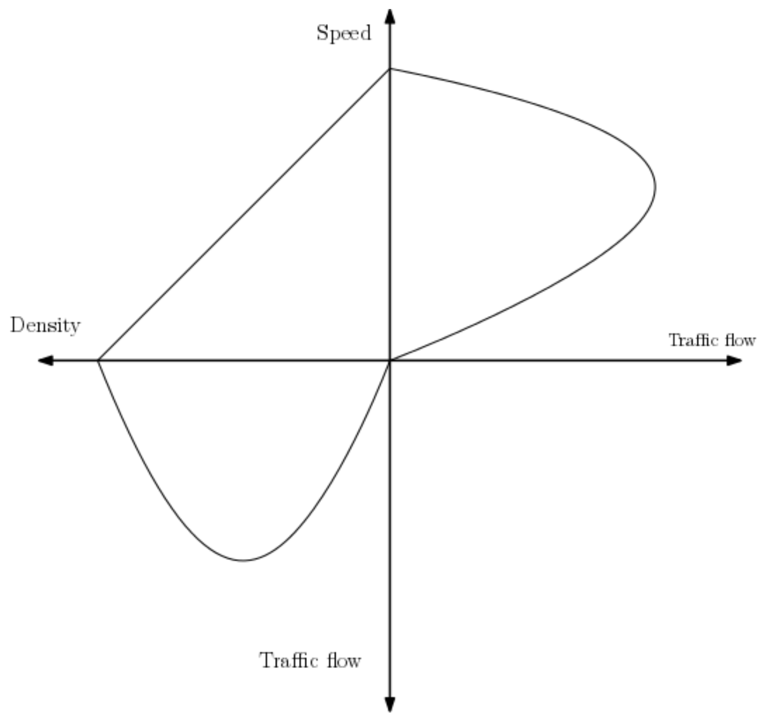


Figure 2.3 Fundamental diagram of traffic (Adapted from Woensel and Vandaele (2007))

We base our conclusion of the fundamental diagram on the fact that it has been shown to be applicable in explaining the dynamics of urban traffic on area or sub-network level. Now, the diagram has the following interpretation: as density, the number of vehicles on a unit length of the road increases the average speed decreases. This is intuitive because drivers have more vehicles that they need to be aware of and adjust their speeds accordingly. At the same time, the flow, the number of vehicles passing a cross section of the road in unit time, increases. The flow, however, does not increase linearly with the density. After the flow reaches a maximum value, any further increase in traffic density reduces the flow. This results in the parabola like relation between the two factors. In Patna, during the peak hours of traffic, the density is clearly well above the critical point, and any increase in the traffic density leads to simultaneous reduction of the average speed and the traffic flow.

The phenomena predicted by the fundamental diagram are clearly present in the urban area of Patna. During peak-hours, the 25 surveyed intersections had an average waiting time of 3.6 minutes per vehicle. In addition, the average vehicle throughput of these intersections deteriorates during rush hours. Off-peak, the intersections have an average vehicle throughput of 38 vehicles per minute but during the busiest hours of the day, only 17 vehicles are able to pass an intersection per minute. This deterioration essentially halves the service rate. In addition to the throughput rate, also the average speed through the intersections is halved during the peak-hours, from 15 km/h to 8.5 km/h (ARS T&TT 2014).

2.3 Network of signalized intersections

When completed, the network of traffic lights that we have to control consists of 88 intersections. The locations of these intersections are shown in Figure 2.4. Currently, civil engineering work is underway to install the traffic signals, the signal controllers, and the detector cameras. The locations of the signalized intersections were chosen by traffic engineering experts and local traffic police from an initial set of 150 possible locations. At the moment, 66 intersections are completed and are operating in either fixed or vehicle actuated control mode.

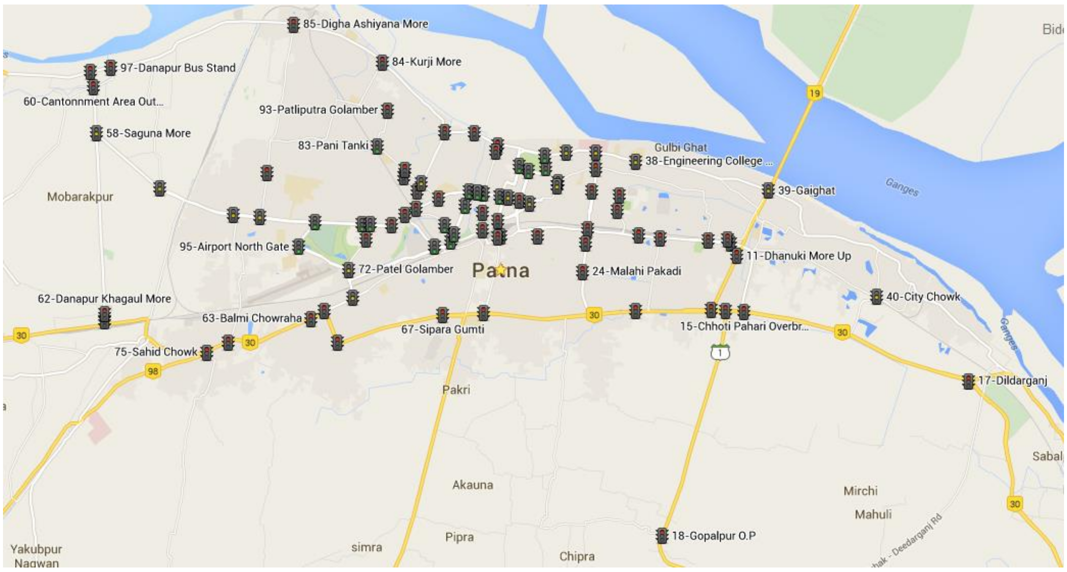


Figure 2.4 The network of signalized intersections

The 88 junctions consist of various types of junctions. 34 of them are typical 4-way, or 4-arm, intersections where 4 roads come together and share the intersection capacity. Such intersections are the stereotypical intersections used in formulating scientific approaches to solving the urban traffic control problem. There are a number of ways the traffic flows can be arranged in phases in such an intersection, an example of one possible way to arrange the conflicting traffic flows into four phases is given in Figure 2.5. Note, the pedestrians are given right of way with the cost of having partially conflicting travel movement with the left turning traffic. In general, this is a common practice that helps reduce the necessary number of phases, which, in turn, reduces the lost time when changing from phase to phase. The lost time consists of *safety time* between phases, during which no traffic is allowed to enter the intersection. This extra time is used to clear the intersection of previously moving traffic flow for the safe progression of the vehicles in the next active phase.

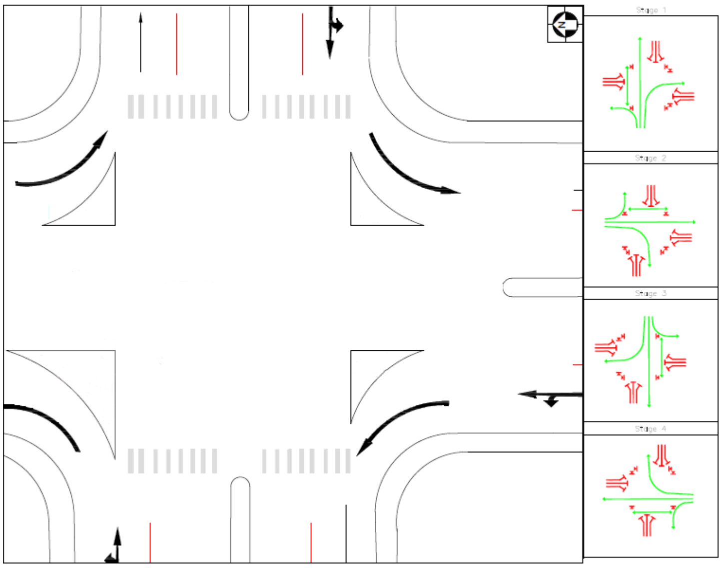


Figure 2.5 Four-way intersection

Other types of intersections in the network of Patna are the 3-arm intersections, either so-called T-junctions or Y-junctions and signalized roundabouts. There are 40 3-way intersections and they form the majority of the traffic light controlled intersections in Patna. One example of T-junctions is given in Figure 2.6. Remaining 14 intersections are signalized roundabouts. Traffic signals are used to relieve the traffic flow by restricting the entrance to a roundabout. Designing the traffic flows in a roundabout is rather straightforward. Traffic movements can be easily split into 4 phases by allowing each of the directions to enter the roundabout at a time. Simultaneously, we can allow left turns from the perpendicular traffic flow on the other side of the roundabout. Because each vehicle movement intersects all pedestrian crossings, for safety a fifth phase can be dedicated to all pedestrian movements. An example of such intersection and the related phase drawings are given in Figure 2.7.

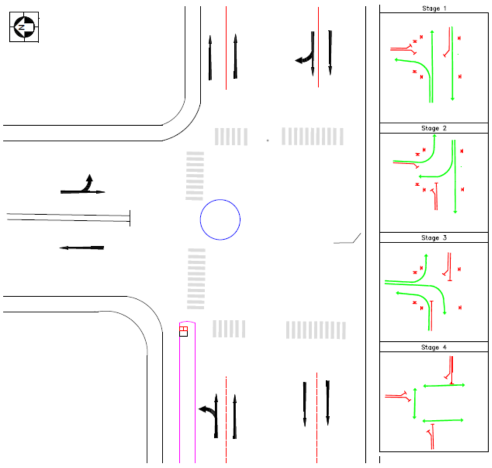


Figure 2.6 A T-intersection

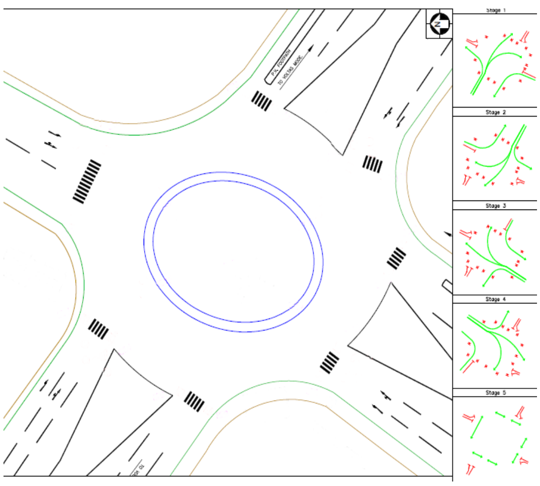


Figure 2.7 A signalized roundabout

The general topology of the traffic network in the city of Patna consists of 4 major arterial routes. All of them run in the east-west direction. The southernmost roads are the New and Old Bypass roads. The city center is split in half by the Bailey Road, and in the north, the Bansghat Road runs along the river. Bansghat Road and Bailey Road both end at Gandhi Maidan Road, a circular route around a park. The New Bypass and Bangshat Road merge outside the city center in western part of Patna. Due to its heavy traffic load, the circular Gandhi Maidan Road is considered to be an arterial route. Another important arterial route, Exhibition Road, connects Old Bypass and Gandhi Maidan in the north-south direction passing Patna Junction railway station. These routes are superimposed in blue color on the map of Patna in Figure 2.2.

Outside of the city center of Patna, the traffic lights on the arterial routes form linear chains of controlled intersections. Because the perpendicular traffic flows are smaller, fewer intersections need to be signalized and the distances between the controlled intersections are long. This has a few immediate consequences. Firstly, the traffic generated along the road between subsequent intersections can be substantial. This makes predicting the arriving traffic flow difficult because we have no way of measuring this emerging traffic demand. Because the arterial routes pass through residential areas, the demand is also probably highly time dependent. During the morning peak hours, the demand is directed towards the city center and industrial areas. During the afternoon peak, in turn, there are considerably fewer cars joining the road between two controlled intersections. Secondly, the platoon leaving the upstream junction does not provide a good estimate for the arrival flow at the downstream intersection. This is caused by the relatively long distance and the varying dynamic characteristics of the vehicles leading to the dispersion of the platoon. In addition, like the emergent traffic demand, a number of vehicles in a platoon might leave the arterial route between the adjacent traffic lights.

As these arterial routes come closer to Gandhi Maidan Road, i.e. closer to the city center, the network of the traffic lights becomes more grid-like. Other traffic light controlled routes merge with the main arterial routes. Because of its circularity, Gandhi Maidan Road itself forms a grid-like situation. In addition to this, the distances between the controlled intersections become shorter and there are fewer uncontrolled intersections along the links between adjacent traffic lights. Because of this, the network effects have higher impacts on the traffic lights. We can assume that a platoon leaving an upstream intersection will arrive at a downstream intersection together. Because there is not enough distance for the platoon to disperse properly, the arrival process at a downstream intersection is determined by the departure process at an upstream traffic light. Moreover, there are fewer vehicles that can either join or leave the road between the intersections. All of these factors make the traffic flows more predictable as the vehicles travel in better-defined platoons.

Some of the intersections are isolated from the other network. They are not located on the arterial routes and are not part of the denser grid of intersections in the city center of Patna. Because these intersections are not directly connected to the other intersections and there are multiple uncontrolled intersections between them, we assume that the platoons leaving controlled intersections have completely dispersed prior to arriving at these intersections. Moreover, we assume that the control decisions made at these intersections do not affect the arrival process at the other controlled intersections in any quantifiable way.

2.4 Traffic heterogeneity

The traffic characteristics of Indian traffic differ from the ones of Western “more organized” traffic. It is important to consider these differences as most of the existing traffic control approaches are developed for the western traffic. In order to choose a method that can be used in the city of Patna, we analyze the possible assumptions of these models and consider the extent to which the assumptions hold for the Indian traffic. We use traffic video recordings showing real traffic situations in Patna to analyze the structure and the behavior of the traffic flows. These videos were recorded at 4 intersections that are currently operating in a fixed cycle mode. The recordings that were made between 26-02-2016 and 27-02-2016 show 8 different inroads.

The first, and most obvious difference, at least to the situation in most countries in the world, is that the traffic is left-handed. The vehicles follow “keep left” rule, and are expected to drive as far left as is safely possible. The side of the road used by the traffic determines the conflicting traffic moves. Whereas in the Netherlands turning right is trivial, if not taking into account parallel bicycle and pedestrian flows, in India turning right conflicts with the traffic flow coming from the opposite direction. Turning left, in turn, is usually trivial.

The Indian traffic differs from the Western traffic in its vehicular composition. Whereas the Western traffic mostly consists of private 4-wheeled cars, the Indian traffic flow comprises a wide variety of vehicle types. Although differing in their behavior and characteristics, these vehicle types, including, for example, motorbikes, bicycles, 3-wheeled rickshaws and bike rickshaws, tractors, and trucks, share the same road. Such traffic, typical to urban areas across developing countries, is often called heterogeneous. The traffic typical to western countries is called homogeneous traffic (Khan & Maini, 1999; Arasan & Koshy, 2005).

The composition of all registered vehicles in the city of Patna and in the Netherlands is presented in Table 2.1. Although both values represent the ownership of the vehicles and are thus comparable, the values concerning Patna concern vehicles registered on the busy metropolitan area, whereas the Dutch values consider the proportion of vehicles throughout the country. As a result, the actual proportion of motorbikes in the Dutch urban traffic is probably considerably lower and, at the same time, the proportion of buses and other means of public transport somewhat higher. We use the vehicular composition in the Netherlands to represent a typical western vehicle distribution. Note also that this table does not present the proportion of bicycles in the traffic flow. Although the Netherlands probably exhibits slightly higher proportion of bicycles than Patna, in the Netherlands bicycles do not usually share the same right-of-way, and thus do not make the traffic heterogeneous.

We conclude that motorized 2-wheelers constitute the largest group of road users in the city of Patna. The proportion of motorbikes is nearly 8 times as high as in the Netherlands. We also see that although passenger cars form the second largest group of vehicles, their proportion of the traffic is considerably lower. This conclusion on the traffic composition is also clearly visible in Figure 2.8 and Figure 2.9, which are snapshots of the video recordings of the actual traffic conditions recorded in Patna. Interestingly, the registration date suggest that auto rickshaws constitute only a small minority of all vehicles, but the video recordings suggest that at times they are the great majority of the vehicles on the road.

Table 2.1 Distribution of vehicles in some vehicle classes in Patna and in the Netherlands (ARS T&TT, 2014; Centraal Bureau voor de Statistiek, 2016)

Vehicle class	Patna	The Netherlands
Passenger cars	19.8 %	73.8 %
Motorbikes	66.7 %	6.1 %
Buses and minibuses	0.8 %	0.1 %
Vans, trucks, and lorries	5.4 %	8.6 %
Auto rickshaws	4.5 %	N/A[1]

[1] Not Available

In addition to the high number of 2 and 3-wheeled vehicles, the traffic recordings indicate a substantial amount of pedestrian movement, street-side parking, relative narrowness of the roads, and traffic demand generated by businesses located right next to the roads. In the traffic recordings, pedestrians appear to cross the road without any consideration to the vehicular traffic flow and where it seems most convenient to them. Even when signalized pedestrian crossings exist nearby, pedestrians might choose not to use it. The above phenomena, typical to developing countries' urban heterogeneous traffic (Khan & Maini, 1999), are clearly visible in Figure 2.8 and Figure 2.9.

Figure 2.8 Example of traffic conditions in the city of Patna

The traffic composition of developed countries has a few key consequences. Firstly, the vehicles that share the same right-of-way differ considerably in their physical characteristics. For example, a passenger car weighs the equal of multiple motorbikes. In addition to the disruption of traffic flow, this is also a safety issue, and drivers of motorbikes and bicycles are most vulnerable to the traffic related fatalities and accidents (Singh, 2012). Secondly, the vehicle types have different dynamic characteristics. The possible top speed of vehicles varies between 5 km/h of pedestrians to up to 100 km/h of motorbikes and passenger cars. More relevant to the problem of urban traffic management is the acceleration, as it determines the time it takes for a vehicle to reach the traffic flow speed after the right-of-way has been established by the traffic light. Motorbikes and cars exhibit the fastest acceleration, with tractors and trucks being the slowest. We can assume that the pedestrians reach their free traffic flow speed almost instantaneously.

Figure 2.9 Heterogeneous traffic conditions

Another key characteristic of heterogeneous traffic is the relative lack of lane discipline. This means that even when lane markings do exist the driver often ignore them. Whereas in the homogeneous traffic vehicles move only forward in the middle of their lane, in the heterogeneous traffic there is also a considerable amount of lateral movement. This is caused by the high proportion of 2-wheeled vehicles that have more space to move laterally. When the traffic flow is stopped, bikes and motorbikes tend to try to fill any existing gaps in the queues so as to be as close to the stop line as possible. They also change lanes often or drive between lanes. A contributing factor is also the relative novelty of traffic control infrastructure and the related lane markings. The traffic lights that ARS is currently installing are the first such system in Patna. In addition, the general attitude towards traffic rules might differ from the western one. Traffic heterogeneity and lack of lane discipline are schematically represented in Figure 2.10 below.

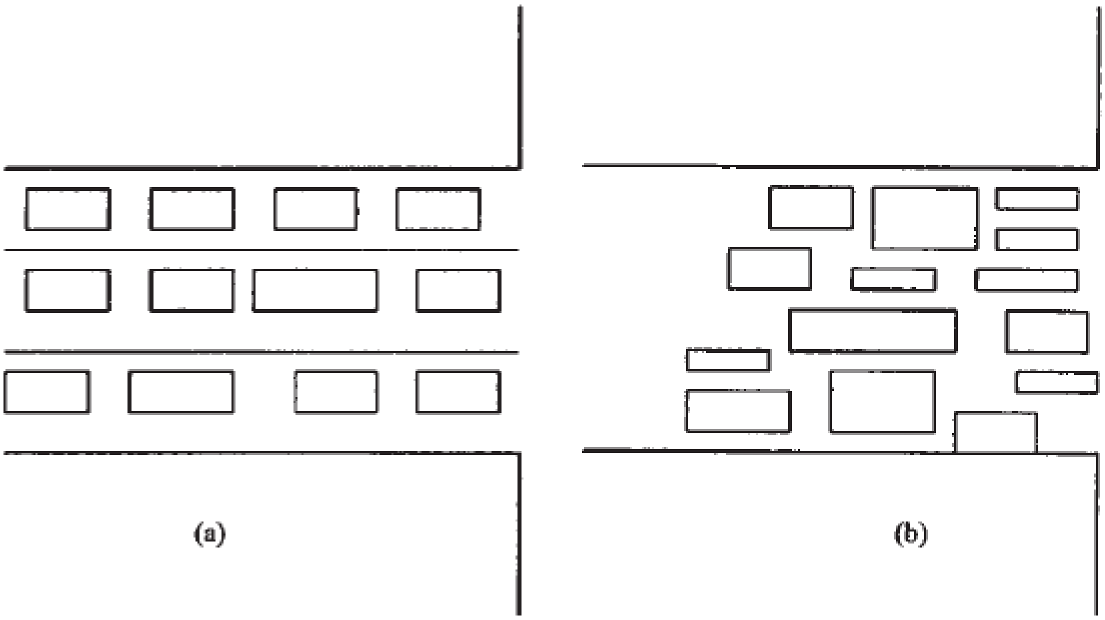


Figure 2.10 Schematic representation of homogeneous (a) and heterogeneous (b) traffic flows. Khan & Maini. (1999)

The analysis of the traffic recordings mostly supports the above result. Motorbikes seem to exhibit the least amount of lane discipline. They try to “micro-optimize” their route and position in the queue. This means that they switch lanes erratically to overtake slower vehicles or drive around pedestrians or cyclists. When the traffic flow approaches signalized intersection showing a red light, the motorbikes usually try to reach the front of the queue and stop as close to the stop line as possible. While doing this the motorbikes essentially fill any gaps in the queue, and as a result, the entire road capacity is efficiently used near the stop line.

Passenger cars and other 4-wheeld vehicles behave more predictably. Usually, they follow each other in a queue similarly to the vehicles in homogeneous traffic. Near the intersections, the cars also follow the lane markings considerably well, and there rarely are more than two cars next to each other. Motorized 3-wheeled rickshaws combine the behaviors of motorbikes and cars. They try to find the shortest and fastest way possible, but due to their larger size, they are sometimes unable to squeeze into gaps between cars in the queues.

The bicycles seem to cause the highest amount of disruption in the traffic flow. The proportion of bicycles and bicycle rickshaws in the traffic is high, but they are considerably slower than the motorized vehicles. The bicycles are frequently overtaken by other vehicles partially causing the erratic lack of lane discipline. Especially the bicycle rickshaws disrupt the traffic as they are more difficult to overtake due to their large size.

One key notion on the traffic is the relatively low speed of vehicles. Even under uncongested free flow situations, the vehicles seem to drive rather slowly. This is probably caused by the relatively poor road conditions, the narrowness of the road combined with road-side parking, pedestrian movements, bad quality of the vehicles and relative proximity of the adjacent intersections. This observation is also supported by the arterial traffic speed survey data, which suggests that the average speed on the arterial roads is only 16.8 km/h during peak hours and 21.7 km/h off-peak (ARS T&TT, 2014). For comparison, in the congested urban area of London in the UK, the average speed of vehicles during the weekday morning peak hours is 24.8 km/h (Department of Transport, 2016). Again, different vehicle groups exhibit varying speeds, with slower bicycles and rickshaws being repeatedly overtaken by other vehicles.

2.5 Traffic detectors and available data

The lack of lane discipline causes some problems to the assumptions of traditional traffic models. Firstly, the concept of a queue becomes ambiguous. Traditionally, each vehicle in the queue, apart from the leader, is following a vehicle directly in front of them. According to the driver's personal preferences, he or she chooses a distance, called the headway, between them. When the traffic is heterogeneous, the concept of a leader is not clear, as a vehicle can have multiple preceding vehicles on the same lane. Thus, also the headway is not clearly defined. Secondly, measuring the density of the traffic, the number of vehicles on a given length of the road, does not represent the traffic flow very well. The density of motorbikes that drive close to each other can be considerably higher than the density of passenger cars, however, they still share the same road capacity, and are essentially part of the same traffic flow.

Consequently, the heterogeneity of the traffic poses difficulties to the measurement of the traffic. Usually, urban traffic control systems use inductive loops to measure the speed and density of the current traffic. Also, the number of vehicles passing the cross section of the road defined by the loop is usually counted. This knowledge of the current state of the traffic is paramount in making well-informed traffic control decisions. The detector loops are usually placed upstream of the junction, thus allowing predicting the arrival pattern to the traffic lights. By placing a loop on each lane the system can also estimate the turn fractions for each exit road, i.e., the proportion of arriving vehicles using the respective exit road. Sometimes loops are also placed immediately after the intersection on the exit roads. This configuration allows accurately estimating the outflow from the junction, and due to traffic flow conservation, traffic does neither disappear or appear between detector loops, predict the arrival pattern at the downstream junction and estimate the turn fractions in the intersection.

Such measurements are made effectively useless by the traffic heterogeneity. Inductive loops do not respond properly on motorbikes or bikes, a majority of the vehicles in the Indian urban traffic. The lateral movement of the vehicles further complicates the measurement. Because vehicles do not necessarily follow the lanes, it is possible that a vehicle can drive over two loops simultaneously, invoking a response in either both or neither of them. It is also possible that a vehicle, especially a narrower motorbike or rickshaw, passes between loops without touching them at all. Similarly, because neat queues do not necessarily exist, multiple vehicles can occupy a detector simultaneously. All the above reasons lead the system to create significantly wrong or biased traffic counts. Besides, the lateral movement of vehicles makes estimating the turn fraction at upstream detector impossible.

Vehicles passing the detector on one lane, say one that would correspond to a right turn, can easily end up on another lane before reaching the stop line of the junction.

Because the difficulties of using an inductive loop, the signalized junctions in the city of Patna are equipped with camera detectors. They use visual imaging and pattern recognition techniques to determine the presence of a vehicle. In Patna, the cameras are used as “virtual loops”, whereby an area of the field-of-view of the camera is defined as “loop” and detector continuously determines the presence of vehicles in this area. An example of these virtual loops in one of the junctions in the city of Patna is given in Figure 2.11. In this case, the intersection consists of three roads, with each having a dedicated detector camera. Cameras A and C define only one loop (1 and 5) assuming that all detected traffic corresponds to one traffic movement. Direction B defines two distinct movements, one loop (4) is dedicated to detecting right-turning traffic, with the 2 other virtual loops (3) detecting the traffic flow that proceeds straight.

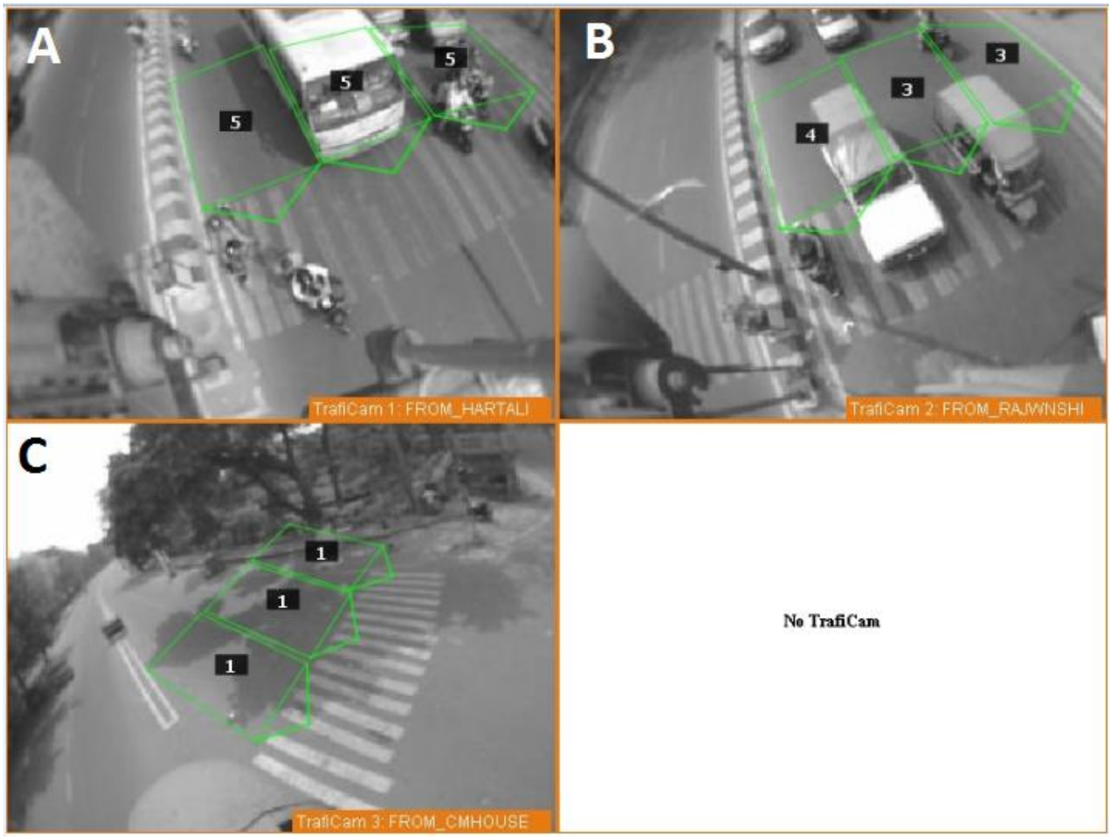


Figure 2.11 Traffic detector camera with virtual loops

In the current configuration of the camera detectors, only the occupancy of the virtual loop can be measured. This means that the detector yields a continuous output signal of either YES or NO (electronically 1 or 0) based on the presence of vehicles in the determined detection area. This asynchronous signal can be sampled by our traffic controller, say every 0.5 seconds. Usually, traffic detectors are also used to count the vehicles that pass the detectors. Due to the traffic heterogeneity discussed above this is not possible in the Indian traffic.

The current configuration makes estimating the turn fractions very difficult if not impossible. In certain cases, detector output can be assumed to correspond to only one traffic movement. For example, if there is a lane solely dedicated to right turning traffic, at some point the traffic lights establish the right of the way only to this movement, and, the lane has a detector loop, we can assume that all of the detected traffic are vehicles making a right turn. However, in most cases, such as in the intersections shown in Figure 2.6 and Figure 2.7 both straight going and left turning traffic share the same right of way. In these cases, the loops are occupied by an unknown and indistinguishable mix of two distinct traffic movements. As it is impossible to determine the turning fractions from the measurements, it is not possible to deduce the amount of traffic on the links leading out of the intersection as a function of the detector output. Consequently, it is difficult for the downstream intersections to accurately predict the vehicle arrivals.

2.6 System requirements

When Analyzing the requirements and limitations that our control method has to fulfill, it is important to consider the preferences and requirements set by the problem owner. These system requirements stem from knowledge on the traffic conditions and the traffic light infrastructure in Patna. Moreover, because this project is an initial step in a development project, it is important to align our solution approach with ARS portfolio and longer term plans. The system requirements presented in this section were mapped using freely structured interviews with the project team in India and the project owner in the Netherlands.

Our system should provide continuous, real-time, and stable control decisions. Because the system is intended for the operational traffic light control, the computation of the control decisions has to fast enough. The requirement of real-time decision making still leaves multiple possible approaches available. The system can, for example, make very fast decision every time unit, say every second, or make slightly slower decisions every traffic light cycle. In our context, stability, in turn, means that our system should not *oscillate, explode,* or *get stuck*. Oscillation is a common problem in a feedback system. A control decision leads to a reaction in the system, and the feedback nature of the system demands a stronger but opposite control decision to return the system to the desired state. If the control input had a bigger effect than necessary, the system can start oscillating around the desired state. Explosion has a trivial interpretation in the context of traffic management. A given control policy might lead to explosive growth of queue lengths in the system. For example, changing traffic light state frequently during traffic peak hours might lead to an unbounded growth of the number of vehicles in the system. Consequently, our system should try to prevent *gridlocks*, situations where existing vehicle queue mutually prevent each other’s traffic movements. If such situation occurs, the system should be able to cope with the situation, and aid in resolving the situation.

As part of the realm of real-time control, our system should be able to adapt to the current traffic situation. A fixed system that is configured using historical traffic information ages quickly as the prevailing traffic conditions change. The adaptability concerns multiple levels of dynamics in the traffic. The traffic flows change gradually over a long time. These changes are a result of seasonality or long-term evolution of the spatiotemporal traffic demand. The most obvious variability in the traffic conditions is the existence of peak-hours of traffic. The composition, routing, and intensity of the traffic differ considerably throughout the day, and the system should be able to cope it. The shortest term variations in the traffic are the random incidents that cause disruptions in the traffic. Examples of such incidents are accidents, buses stopping to pick up customers, or discharge of a large number of vehicles or pedestrians from events. Although these incidents are of short duration, they create shocks in the traffic that need to be managed by our system.

A key feature our system should provide is scalability and robustness. With scalability, we mean that intersections can be added to or removed from the control network without major implications for the operation of the system. This is important as the current set of traffic lights is an initial development in the city of Patna. The size of the city and the complexity of the urban traffic network justify adding additional traffic lights in the future. A scalable system can add these new intersections without major reconfiguration or reprogramming. In addition, the computational complexity of a scalable system allows the performance to remain on a reasonable level.

The ability to remove intersections from the system is related to the robustness of the system. Because the communication and control infrastructure can be unreliable, due to, for example, unreliable electric supply, or unreliable internet connection, the system should be able to cope with the momentary deterioration of the required functionalities. The loss of one or more intersections should not prevent the system from running normally at other still functional intersections.

2.7 Conclusions of the case study

We conclude that the general traffic network and the flow characteristics make controlling the traffic in the city of Patna a difficult task. Firstly, the network itself varies greatly. Some parts involve sparsely spaced arterial traffic lights. The platoons leaving an upstream intersection might be completely dispersed before arriving the downstream intersection. Secondly, the vehicles that compose the traffic flows can behave erratically and do not always conform the traditional traffic control assumptions. Because of this, and suboptimal location of the traffic sensors, the measurement of the actual traffic flow is difficult and our system can acquire only limited information on the actual traffic conditions

The implementation of an urban traffic control system will probably help alleviate the adverse effects of the current traffic congestion problem. The video recordings indicate that, in general, the vehicles do respect the authority of the traffic lights quite well, by forming queues and not jumping the red light. We can thus assume that our control input would have the desired effect on the traffic flow. In addition to this, an earlier research already found that the intersections with human controllers guiding the traffic exhibit lower levels of traffic congestion than the uncontrolled intersections. This is also suggested by the subjective assessment by the local traffic police, claiming that the intersections already operating under fixed traffic light control exhibit lower levels of congestion (Nair, 2016). Moreover, the intersections in the city center area are very close to each other, suggesting that coordinated traffic control can be beneficial.

Our urban control system should aim at minimizing the average delay caused by the traffic lights. Because the free flow speed is low, and probably bounded by the local traffic culture and the limitations of the infrastructure, we assume that the maximum reduction in traffic times is achieved by reducing the queueing time of vehicles at the traffic lights. This in contrary to increasing the average speed by equalizing the degree of saturation of the roads. This is further argued by the fact that currently the waiting times in the intersection are very long.

The reduction in average waiting times can be achieved by maximizing the utilization of the intersection capacity, i.e., reduce the amount of green time that is spent serving low or non-existent traffic flows. A further reduction in the waiting times can be obtained by improving the coordination between the traffic lights. In the best case, key traffic flows, such as those on the busy arterial routes, can pass successive traffic lights without stopping at them.

The control decisions have to be made adaptively and in real-time. This means that our system has to be able to capture and utilize both long and short-term variations in the traffic. Because our system makes operational traffic control decisions, the computations have to be relatively simple and fast.

The system should be robust against communication and hardware failures, and it should be scalable in the number of traffic lights and the topology of the network.

Our system is limited in the information that can be gathered on the traffic conditions. The number of individual vehicles cannot be counted. In addition to this, turn fractions cannot be estimated in the real time by the detectors. By sampling the occupancy of the loops frequently we can estimate the gap between vehicles. This suggests that we could use an approach that serves a traffic movement until the whole platoon is served, or vehicles are arriving too sparsely. This approach is usually called vehicle actuation.

Lastly, our control approach needs to be able to handle the awkward topology of the network, and the varying geometries of the signalized intersections. The intersections on the arterial routes away from the city center are only loosely connected forming linear chains of intersections and having long distances between subsequent traffic lights. In the city center area, however, the signalized intersections are closer to each other forming a grid of intersections that are highly dependent on the decisions made at other intersections. In addition to this, the varying geometry of the intersections has to be handled appropriately. Unlike a traditional assumption of a grid of four-way intersections each having two service phases, many of the intersections are signalized roundabouts or three-way junctions. Besides, pedestrians are many times given a dedicated pedestrian phase that has to be given green time separately from the vehicular traffic. The structure of the phases and the total number of phases can vary from intersection to intersection.

3 Literature Study

In this chapter, we discuss the existing literature on the traffic control. By doing this we answer the second research question: *What solution approaches exist in the literature?* The goal of this chapter is to find methods that we can use to create our traffic controller. We start by defining the urban traffic control problem in Section 3.1. We investigate a classification of control methods and describe the most basic urban traffic control problem, the control problem of one individual intersection. After this, in Section 3.2 and 3.3, we conduct a structured literature review. We start by considering a taxonomy of solution approaches. This helps us analyze the differences in the solution approaches and draw conclusions on the current state of the literature. Next, we present a small synthesis on the existing commercial solutions. Although we are interested in the state-of-the-art scientific solutions, existing commercial solutions are results of long development and have gone through trial-and-error phases. This helps us define the general requirements of a control algorithm. We end by reviewing the existing literature and scanning for the recent developments in the field of traffic light control. We compare various approaches and argue their usefulness in our problem context.

3.1 Urban Traffic Control Problem

Papageorgiou et al. (2003) define urban traffic control with the loop structure shown in Figure 3.1. The control system takes *disturbances* as input. These include, for example, the demand and its variability and possible incidents on the road. These can be measured or detected by *sensors*, which provide data to the computers that control the system. The data is transformed by a *control strategy* into *control inputs*, which determine the way the *control devices* should control the traffic infrastructure. The control strategy computes the control input based on goals defined by the human user. The extent to which these goals are achieved is measured by the output of the control loop. In the figure *total time spent* is used as the output, but various other indicators, such as total delay, average travel time, or vehicle throughput of the system, can be used to measure the performance of the control strategy. In this thesis, we call the system that implements the measurements and control strategy *urban traffic control system* or UTCS. Often these systems are also called *adaptive traffic management systems* ATMS. They are a subgroup of a wider concept of *intelligent transportation systems* (ITS).

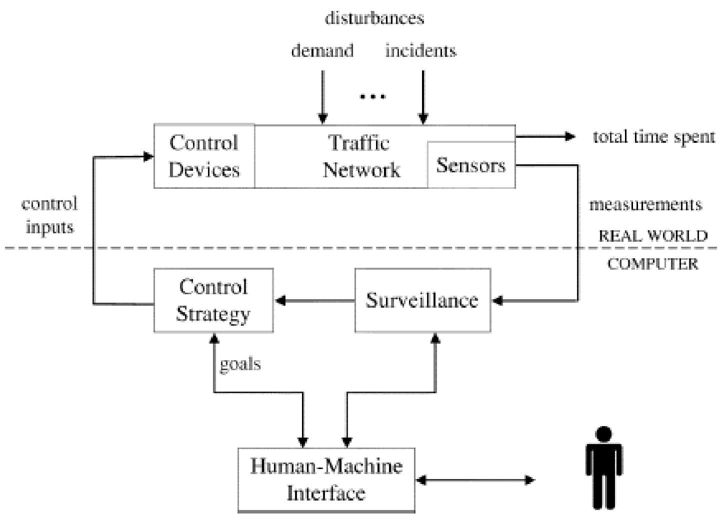


Figure 3.1 Traffic control loop (Papageorgiou et al. 2003)

In this thesis, we are interested in the *control strategy* part of the traffic control loop. We want to come up with an algorithm that makes traffic control decisions based on the sensor measurement data and the optimization goals determined by a human operator. Although making optimal traffic control decisions is difficult, the problem can be represented in a very simple form:

$$argmax_{\omega \in \Omega} f(\omega, \boldsymbol{\sigma}) \quad (1)$$

Function f represents the performance value of the traffic system. It takes two inputs, traffic signal control decisions ω and the actual traffic $\boldsymbol{\sigma}$. For the purpose of the optimization and decision making, the traffic conditions are exogenous but observable. This means that we can use information acquired through sensory measurements (estimate $\boldsymbol{\sigma}$ with $\widehat{\boldsymbol{\sigma}}$) to decide the optimal control policy ω. This variable describes the way the traffic lights are controlled. The set Ω represents every feasible control input. This set is defined by the network topology and the traffic rules and laws. The function $f: \omega, \boldsymbol{\sigma} \rightarrow \mathbb{R}$, is an algorithm that translates the traffic condition and the control decisions into a single performance value. Measuring the performance of the system depends on the point of view of the decision maker, and the function can be chosen to represent for example the number of stops by cars, the total time lost, the total delay, or the number of vehicles that can pass through the system in a given time. This function does not need to be simple, and often the decision maker cannot directly evaluate the effect of control decisions on the traffic performance.

A number of ways have been presented to solve the above problem. Usually, the methods try to use a simplified objective function *f* or try to reduce the size of the search space Ω. Often both options are used simultaneously to find near-optimal control decisions heuristically. In this chapter, we first investigate the categories of urban traffic control and define the control problem of an individual intersection. This serves as defining the decision variables and the search space Ω. After this, we investigate various methods that can be used to solve the above problem.

3.1.1 Categories of traffic control strategies

We consider here a classification of control strategies. This helps us to understand the fundamental differences in the ways various control approaches model the urban traffic control problem. Traditionally, the traffic control strategies are categorized based on two factors, namely their *temporal* and *spatial* dimensions. Traffic controller can either work based on a *fixed* control schedule using historical data, or on *real-time* optimization. The spatial dimension divides methods between *isolated*, methods that consider one junction only and that relax the problem by not taking the network effects into account, and *coordinated* methods that consider a network of traffic lights where intersections affect each other (Bell, 1992). One way to illustrate the categories of traffic control strategies is the 2x2 matrix shown Figure 3.2.

Fixed, or fixed-time, control schemes emerged earliest and represent the first attempt to control the traffic flows. Historical traffic data is collected and then used to infer statistical patterns and characteristics that can be used to design a fixed control schedule. A clock is used to cyclically implement the plan. The earliest attempt to optimize and model this strategy is the seminal paper of Webster (1958). In this paper, Webster used empirical methods to establish a well-known and often-used formula for the optimal cycle time of an isolated junction. Essentially, Webster used the degree of saturation of the roads at the intersection to find a cycle time that guarantees sufficient capacity and minimizes the average lost time in the intersection. The same degrees are then used to design the *split times*, i.e. to allocate the cycle time to each of the phases. Some papers refer to these control strategies as *off-line* control, emphasizing the fact that the control plans are precomputed and the controller simply follows them using a clock.

The temporal opposite of fixed control are the *real-time* and *on-line* control strategies. These control schemes aim to continually adapt the decisions to actual traffic conditions. This usually involves using various sensors, or detectors to measure real traffic flows and provide data for the control decision making. Consequently, many researchers name these strategies *adaptive.* The key idea of using real-time information is to overcome the problem of *ageing*, or the increasing discrepancy between the current traffic conditions and the traffic conditions for which the fixed-time plan was computed, that the fixed plans face. Van Katwijk (2008) further divides real-time policies into *actuated* and *adaptive* control policies. Actuated policies refer to commonly used control policies that use detector information to make simple extend-or-terminate decisions concerning the currently active green phase on one intersection. Adaptive policies refer to more comprehensive real-time adjustments of the control parameters.

In the spatial domain, the simplest form of control policy is the *isolated control*. These policies consider only one intersection and assume that it has no effect on the nearby intersections and that it is not affected by the adjacent intersections. Because only local information is considered, these policies are sometimes called *local control* (van Katwijk (2008)). Local control, in combination with the fixed time plans, were historically the first attempts to automate traffic control, with works such as Webster (1958). Even today, local control is relevant in the form of popular vehicle actuated systems and the emerging interest in various decentralized control schemes.

The spatial extension of local control leads to *coordinated* or *area* control. These policies acknowledge the fact that traffic signals at adjacent intersections affect each other. They also consider network-wide performance values and try to establish a connection between control decision at an intersection and the related global performance. The earliest, but still relevant, example of coordinated control is the well-known *green wave*. This technique refers to allowing certain directions in the traffic flow to progress through the network without stopping. The green wave is achieved by offsetting the initiation of green light between intersections in the green wave direction with the average driving time needed to reach the next junction. With such control, a vehicle departing from an intersection reaches the downstream intersection just as that intersection initiates green to this vehicle, and the vehicle can progress uninterrupted. Because of the obvious challenges caused by real-time computation and communication, these approaches have emerged most recently and are currently the standard state of the art solution approach to traffic control.

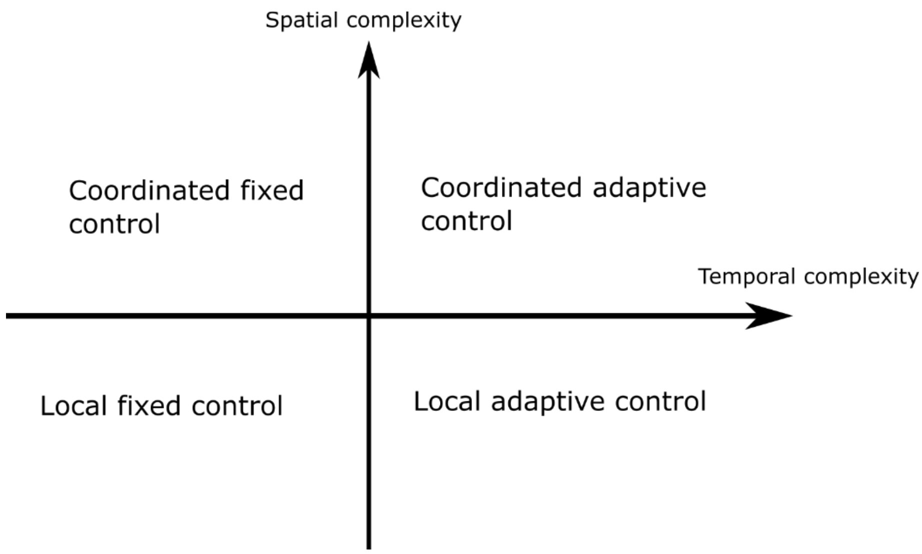


Figure 3.2 Four categories of urban traffic control systems

3.1.2 Fixed-time isolated traffic light control

Although in this thesis we are interested in the real-time control of a network of intersections, the control of an isolated intersection provides us with a good introduction to urban traffic control. Moreover, the control of urban traffic network boils down to controlling multiple intersections simultaneously. Considering the control of an isolated intersection introduces us also to the structure of the decision variables, i.e. the control parameters of the traffic light. While some network control approaches relax the assumption of the type of control presented here, most of the exiting solutions simply adjust the parameters of individual intersections according to predefined network coordination algorithm.

As defined in Figure 3.1, the purpose of the control strategy is to compute the control inputs that are to be given to the control devices. The most commonly used traffic control devices are the traffic lights installed at an intersection of roads. The basic function of traffic lights is to provide temporal separation of *conflicting traffic flows*. Because of increased risk of collision, conflicting flows are flows that cannot simultaneously share the same space. In addition to this safety aspect, traffic lights can improve the flow of the traffic by sequentially allocating the full intersection capacity to traffic flows. Because traffic lights remove the risk of conflicting with other traffic, the traffic flow that has been given the right of way can drive faster and vacate the intersection quicker. This can increase the degree of capacity utilization, and consequently improve other performance indicators such as average waiting times or intersection average throughput.

The problem of controlling the traffic lights of an individual intersection, which we call the *traffic light control problem* concerns making the best possible control decisions concerning the traffic lights. Traffic lights are installed in an intersection, where two or more traffic flows share the same space. In Figure 3.3, we give an example of a typical intersection, the so-called 4-arm intersection, considered by the traffic control literature. Vehicles enter the intersection via *in-roads* (roads A, B, C and D) and leave the intersection via *exit roads* (roads E, F, G, and H). Each of the roads consists of one or more lanes, and each lane corresponds to a set of allowed traffic movements (indicated with the arrows and the numbers in Figure 3.3).

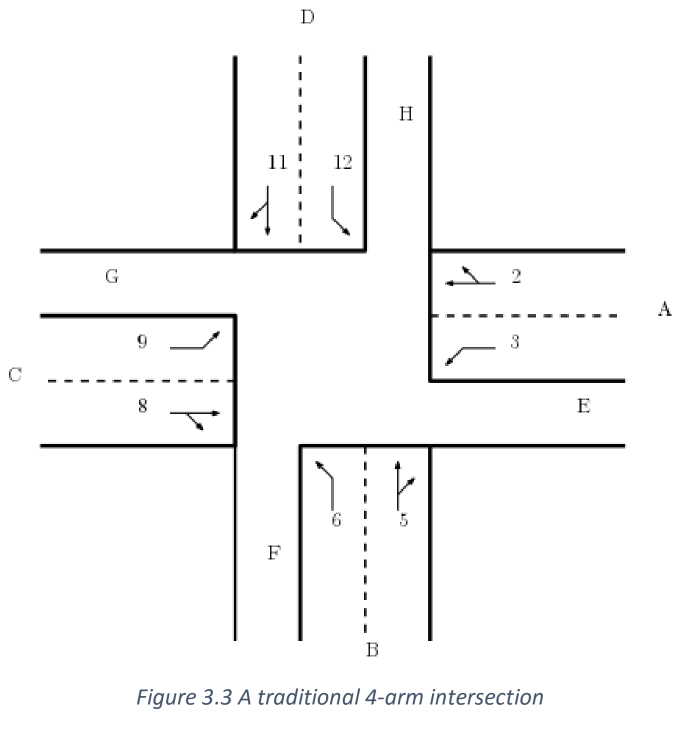


Figure 3.3 A traditional 4-arm intersection

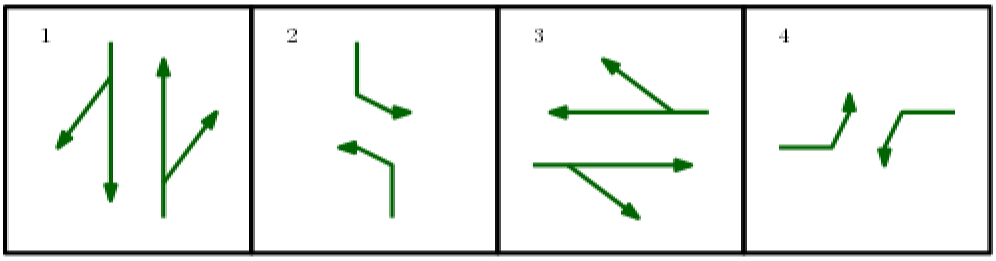


Figure 3.4 Possible phase plan for a 4-arm intersection

The basic unit of traffic light control is the *phase,* which represents the group of traffic movements that are allowed to move simultaneously. This group can consist of non-conflicting or *partially conflicting* traffic movements. Partially conflicting moves share a part of the intersection space but have well-defined right-of-way rules. A common example of partially conflicting move is, in the right-sided traffic, the conflict of vehicles turning right and pedestrians crossing the parallel crosswalk. In this case the vehicles must yield and slow down for the turn, and the movements are generally placed in the same phase. In our example, the traffic movements in the intersection are divided into four phases, as shown in Figure 3.4.

As mentioned in Chapter 3.1, the most traditional and simplest control approach is to organize the phases of a traffic light into repetitive cycles. A *phase switching sequence (PSS)* that visits each of the phases sequentially and a *cycle time,* referring to the time taken to show green light to each of the phases in the PSS and the *yellow phases* between the green lights, are chosen. Formulas have been presented as to how to choose the fixed cycle time. Webster (1958), for example, presents an empirical formula on a cycle time that guarantees sufficient capacity and equalizes the degree of saturation on the roads leading to the intersection. The PSS can be chosen as to minimize the time lost showing red and yellow traffic lights. Within the cycle, each phase is allocated a part of the total cycle time, called *split time.* This determines the duration of the green light that is shown to the respective traffic movement. Consequently, the cycle time is the sum of split times of all green, yellow, and red phases. A clock is used to determine the duration of each green light. After the split time of the currently active traffic movement has been reached, the system automatically activates the next green phase, and the related safety yellow and red time between them. Yellow time is used for safety to indicate the drivers that the active phase is about to change. The red phase, during which no phase is active, is used to give vehicles still crossing the intersection time to clear the conflicting area before the next phase is activated. After each of the phases in the PSS have been activated, the clock resets to beginning and the cycle starts again.

An example of fixed cycle control decision for the intersection of Figure 3.3 using the phase structure of Figure 3.4 is given in Figure 3.5 below. The phases are organized in a fixed cycle that has the duration of 142 seconds. First Phase 1 is served for 40 seconds, after which yellow light is shown for 5 seconds to signal the drivers that the active phase is about to change. To provide safe transfer from Phase 1 to Phase 2, traffic lights show red light to all directions for 3 seconds. Now, Phase 2 is given green light for 30 seconds. A similar green-yellow-red sequence is conducted for Phases 3 and 4. After the all-red period after Phase 4, 142 seconds has elapsed since the beginning and the traffic cycle is initiated again. A fixed cycle controller will cycle through this predetermined cycle until either switched of or given a new cycle.

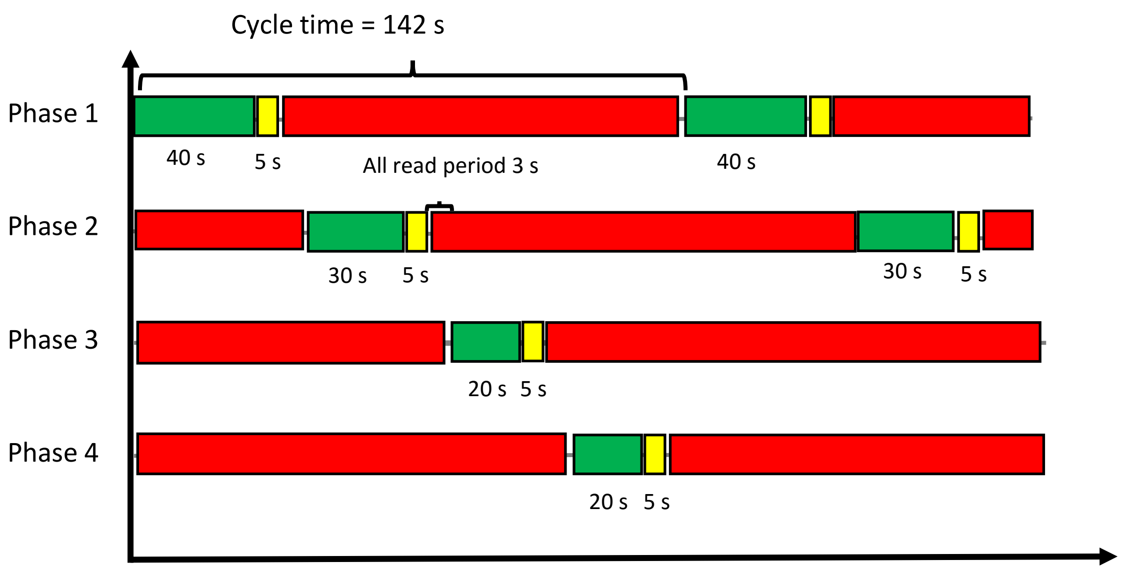


Figure 3.5 Example of a fixed cyclic plan

The urban traffic control problem concerns making control decisions in a network of traffic lights. Because the distances between traffic lights are usually short, the arrivals of vehicles are defined by the control decisions of the adjacent traffic lights. In such situations, coordinating the traffic lights can improve the performance of the system. The traditional way of coordinating the traffic lights adds one additional decision variables, the *offset.* Firstly, all traffic lights in a given region are set to operate with a common cycle time. This cycle time should reflect the traffic condition at the most congested intersection. If the differences between the degree of saturation between the intersections are large, some of the intersections can operate with an integer multiple of the common cycle time. Secondly, intersections are set to offset the beginning of their control cycles. If this time difference is chosen to be the expected travel time between the intersections, in the optimal case, a platoon of vehicles leaving an upstream intersection reaches the downstream intersection just as the right of way is established for this traffic flow. Vehicles can now proceed without stopping at the intersection.

3.1.3 Difficulties in controlling the urban traffic

Although solving the urban traffic control problem is important, it turns out, controlling traffic lights in an "optimal" way is very difficult. According to Papageorgiou et al. (2003), urban traffic control problem is *stochastic, non-linear*, and *non-convex*. These characteristics rule out many efficient control and optimization methods. Lämmer and Helbing (2008), in turn, point out that urban traffic networks is a large, complex feedback system. Next, we briefly consider these impediments, because the modeling choices and assumptions used in the literature reflect these factors. Moreover, the control approach we want to propose is also susceptible to these limitations

One of the key difficulties in urban traffic control is that modeling the flow of the vehicles is difficult. There are a number of different ways to model *uninterrupted traffic flows*, which occur on highways and arterial roads, where vehicles can flow without obstructions. One example of such models is the famous LWR model that describes traffic as kinematic waves. The formulation is based on fluid dynamics and was developed independently by Lighthill and Whitman (1955) and Richards (1956). These models, however, are unable to describe the *interrupted traffic flow* in the urban area, where intersections are located closely and traffic rules and traffic lights may regulate the flow of the traffic.

One common way to model the traffic dynamics in urban network is to divide time into small time steps t and describe the number of vehicles waiting at an intersection at each point in time as a queueing process. During a phase that serves an in-road the respective queue is discharged using the full intersection capacity. This flow rate is usually called the *saturation flow.* When a road is not served, i.e. shown red light, the queue builds up according to the service process of the adjacent intersection and external arrivals. Vehicles that were not served during the service phase form a queue and wait for the next service phase. Such a formulation, used by, for example, Varaiya (2013) and Le et al. (2015), is presented in Equation 2:

$$x_k(t+1) = x_k(t) - C_k(t+1)S_k(t) \wedge x_k(t) + \sum_l a_{k,l}(t) + d_k(t+1) \tag{2}$$

where $x_k(t)$ represent the number of vehicles at intersection *k* at time *t*, $C_k(t)$ is the random number of vehicles that can leave the intersection during one period, $S_k(t) \in \{0,1\}$ is the control decision variable representing whether movements from queue *k* are allowed, $a_{k,l}(t)$ represent arrivals from queue *l* to queue *k* during time slot *t*, and $d_k(t+1)$ are the external arrivals to the queue. The operator ∧ represents the minimum operator, i.e. $(a) \wedge (b) = \min\{a, b\}$.

The above formula serves here to point out the inherent stochasticity of the urban traffic control problem. Clearly, the number of vehicles on the road *k* must be a random variable, because many of the summands describing it are random variables too. The number of vehicles that is served during a service phase is stochastic, as it depends on the drivers' behavior and the types of vehicles. The number of external arrivals to the system is random and the number of vehicles arriving from the adjacent intersections depends on a random queueing formula as above.

The formulation reveals also the non-linearity of the urban traffic control problem. Firstly, the evolution of the queue length is described by a minimum operator. This is common to all queueing systems but leads to non-linear behavior of the queue length. Secondly, the queue length in a given intersection at a given time depends on the queue length and the service process of the adjacent intersections one time step earlier. These values, in turn, depend on the queue lengths and control decisions of the intersection of our interest one time step earlier. Lämmer and Helbing (2008) point out that such a feedback loop leads to non-linear and complex network behavior. According to them, in long term, it is impossible to predict the way a given control decision affects the network performance and the way the state of the network evolves.

The urban traffic control problem is also computationally demanding. As Papageorgiou et al. (2003) point out, the problem can be represented with discrete state variables and discrete decision variables, and the problem becomes a combinatorial optimization problem. In addition to this, Papadimitriou and Tsitsiklis (1994) have shown that the closely related problem of finding optimal control policy to the multiclass closed queueing network is an NP hard problem. Consequently, the state, solution, and action spaces grow exponentially as the number of intersections in the control problem increases. Optimal control is, thus, only tractable for individual intersections (Papageorgiou et al. 2003). Any control approach that attempts to control a larger number of intersections simultaneously must rely on heuristic computations.

3.2 Taxonomy of van Katwijk

Before considering the state of the current literature we want to establish a framework, which we can use to analyze the differences between control approaches. This framework can be used to assess the relative merits and usefulness of the theoretical model that we identify in the literature. It also helps structuring the results of the literature review. One such good framework for categorizing the differences between control methods is the taxonomy proposed by van Katwijk (2008). According to him, the traffic control methods differ in six dimensions:

1. Architecture
2. Search algorithm
3. Decision variables
4. Prediction models
5. Planning horizon
6. Update frequency

The architecture of a traffic controller refers to the way various control modules, detectors, software, and hardware, are related to each other. Possible architectures are: *centralized, distributed,* or *hierarchical*. Centralized traffic controllers employ powerful central computer that gathers all network information into one place and makes global control decisions. Traffic light controllers at the intersections simply follow the centrally computed decisions. Distributed, or decentralized methods, in turn, are the complete opposite to the centralized systems. Local traffic light controllers are smart and capable of using locally available information to make decisions concerning the junction. The hierarchical systems try to combine the advantages of both systems by employing both regional and local control methods. The central computer could, for example, compute a framework decision, within which the local controllers can make local optimization decisions. The different design choices per dimension of the taxonomy are presented in Table 3.1. The table leaves out the dimension of update frequency, as the choice in this dimension is largely decided by the choices in the other dimensions. A model cannot be updated more frequently than it is computed, and the plan has to be updated every time the end of the planning horizon is reached.

Search algorithms refer to the way the control system find (near) optimal control decisions. Firstly, the algorithms can be divided in *complete* and *incomplete exploration*. The former evaluates each possible control decisions and the latter uses “rules of thumb” to concentrate only on “promising” control decisions. Secondly, algorithms are traditionally divided into *constructive* and *move-based* algorithms. Constructive, using methods such as *dynamic programming* or *branch-and-bound*, sequentially build optimal decisions from scratch. Move-base algorithms, in turn, start with a feasible, but possibly bad, solution and try to find ways, or “moves”, to improve it. In general, but not always, complete exploration is related to constructive algorithms and incomplete exploration to move-based algorithms.

Decision variables can be either *cyclic* or *a-cyclic,* reflecting a key modeling choice (van Katwijk, 2008). Cyclic solutions start with the imperative of designing regular plans that are repeated periodically. Most of the existing UTCSs operate cyclically, and try to compute the optimal *splits, cycle time*, and *offset* values. A-cyclic approaches relax the requirement that phases always follow each other in a predetermined sequence. The optimization goal is to find the best control sequence. This dimension of the taxonomy is related to the distinction between *phase-based* and *group-based* control mentioned by Osorio and Bierlaire (2008). The former policy assumes the phase order to be fixed, and the latter aims to construct the best phase sequence.

Predictions are necessary for adaptive control of the traffic lights. It is useful to know the number of vehicles arriving at the traffic lights. This can either be done measuring *imminent arrivals* using a detector such as an inductive loop. *Estimating the expected arrival flow* can be based on using a predictive traffic model or using exit detectors at upstream intersections. *Turn fractions*, or *turning rates,* indicating the distribution of vehicles from an incoming lane to the outgoing lanes of the intersection are an important input to urban traffic control systems. They can be estimated and modeled in a number of ways. These methods include entropy maximization, Bayesian estimation, and maximum likelihood estimation. Lastly, proactive systems must model the queues at the intersection. This can be done using either *vertical* or *point queues* that assume the links to have infinite capacity and the number of vehicles in the queue is simply a number, or using *horizontal* or *spatial* queues that also model the queue spillback and possible blocking behavior caused by the fact that link capacities are finite.

Planning horizon defines the *look-up* or *predictive* capabilities of a traffic control policy. Urban traffic control systems use measurements to establish the current situation and make predictions over the planning horizon to the future. Also the goodness of a control decision is determined over the planning horizon. Essentially, planning horizon defines “how far in the future the system looks when making decisions”. The length of the planning horizon can be either fixed or variable, and it is split into successive intervals of either fixed or variable length. Deciding the length of the planning horizon is a trade-off. In one hand, the longer the planning horizon is the better the policy can prepare for future contingencies and the better the decisions are. On the other hand, the solution space grows exponentially in the length of the planning horizon and accurate models can be intractable for even short planning horizons. In addition, if the planning horizon is very long the models can become biased as predictions are made based on other predictions.

Update frequency describes the frequency of decision making. In one hand, the update frequency is bounded by all previous considerations, on the other hand it determines the adaptive capability of a control method. For example, if a control method does optimization only every one hour, the model cannot capture minute-to-minute variations in the traffic. If the traffic model is very complicated and has a long planning horizon, or if very complicated optimization methods are used, the necessary computation time limits the update frequency. Consequently, accurate models have troubles in adapting to the changing traffic, as infrequently made decision cannot capture the short term variations in the traffic flow.

Table 3.1 A taxonomy of adaptive traffic control systems

<table>
<tr><th>Dimension</th><th colspan="3">Design choices</th></tr>
<tr><td>Architecture</td><td>Decentralized</td><td>Hierarchical</td><td>Centralized</td></tr>
<tr><td>Search algorithms</td><td>Constructive vs. Move-based algorithms</td><td>Complete vs. incomplete exploration</td><td></td></tr>
<tr><td>Decision variables</td><td>Cyclic</td><td>A-cyclic</td><td></td></tr>
<tr><td>Prediction models</td><td>Arrival models
Imminent arrivals
Expected arrivals</td><td>Queueing models
Vertical/point queues
Horizontal/spatial queues</td><td>Turn fractions
Entropy maximization
Bayesian estimation
Maximum likelihood</td></tr>
<tr><td>Planning horizon</td><td>Variable horizon fixed interval</td><td>Fixed horizon variable interval</td><td>Variable horizon variable interval</td></tr>
</table>

3.3 Existing commercial systems

There is a large gap between the state of the art commercial solutions and the recent theoretical methods. This is due to the fact that traffic network management tools operate on the “forefront of technological capabilities” (Papageorgiou et al., 2003; van Katwijk, 2008). Existing commercial solutions take in general centralized and heuristic approach in solving the network control problem. They can be divided into *generations* based on their level of sophistication and capabilities (Hamilton et al. 2012; Gartner 1985)**.** They are complete systems involving various modules and performing a number of tasks. The exact decision procedures, however, are often not publicly available, because the systems are commercial and proprietary. The recent theoretical approaches, on the other hand, take more limited approach concentrating on individual parts of the process. Due to these limitations, these systems have not been applied in practical installations very often, and only give theoretical or simulation based results on the performance. Because of this discrepancy, we start by giving a short review of existing commercial systems, followed by a more comprehensive analysis of existing theoretical results. The commercial systems that have went through decades of iterative improvements and have enjoyed relative commercial success give us indication on characteristics that a good control algorithm should exhibit.

The most successful commercial UTC system with a US market share of 33 % and more than 200 international installations is SCOOT (NCHRP Synthesis, 2010; Hamilton et al. 2012). The tool is a real-time extension of the popular TRANSYT fixed-time network optimizer developed by the same authors Like the name *Split, Cycle, Offset Optimization Technique* suggests, the algorithm makes cyclic decisions based on upstream measurements of the actual traffic conditions. The detectors that are placed upstream from the intersections are used to create a “cyclic flow profile” that describes the variation of traffic during a signal cycle. Based on these flow profiles the length of queues at intersections are predicted. The performance of the current control decisions is measured with a combination of performance indicators, including queue lengths, number of stops, and, degree of saturation. Splits, cycles and offsets are incrementally slightly increased or decreased based on the measurements of the traffic flow. The network is split in fixed subnetworks that share the same cycle time. Adaptation to the long term changes in the traffic flow is achieved through the sequential application of small incremental changes (Hunt et al. 1981; Robertson and Bretherton, 1991).

Another popular network management tool is the *Sydney Coordinated Adaptive Traffic System* or SCATS. Like SCOOT it was devised in the 1970s. This tool belongs to an earlier generation of network controllers as it is *semi-adaptive*. This means that the system maintains a library of signal plans that are computed off-line, and chooses plans that best fit the current traffic situation measured by upstream detectors. Minor changes are made to the signal timings to better fit to the current traffic conditions. In order to reduce the complexity of the control problem, SCATS splits the network into subnetworks that can be managed independently. This clustering into subnetworks is dynamic, and based on the actual traffic conditions. When the traffic conditions change, new subnetworks can be created or intersections in existing clusters can be switched to other subnetworks.

A common approach to solve the urban traffic control problems is to use dynamic programming. These techniques emerged in the 1980s. Although dynamic programming, a mathematical technique used to solve multistage decision problems that relies on dividing the initial problem into a number of smaller sub-problems, theoretically fits the setting of traffic light control problem the challenge of the method is that the computational requirements grow exponentially with the problem size. PRODYN (Henry 1983) tries to restrict the solution space by dividing the network control problem into a set of subnetwork control problems. These subnetworks are optimized with dynamic programming, and the time resolutions is set to 5 seconds. The global network problem is then solved using a two level iterative calculation procedure. OPAC (Gartner 1983), on the other hand, restricts the solution space by dividing the solution procedure into stages (with length between 50 and 100 seconds) and requiring at least one but no more than three phase changes during each stage. A rolling horizon method is used to reduce the dependence of predicted traffic flows.

RHODES, a more recent urban traffic control system from the 1990s, tries to incorporate the improvements in computer and communication technologies in an urban traffic control system. The researchers propose a hierarchical optimization framework to solve the problem of controlling traffic lights. Each layer in the hierarchy, three in total, considers more sophisticated and accurate data than the layer before, and makes decisions based on the layer specific data, layer specific decision model and the decisions made by the preceding layer in hierarchy. The coarsest decisions are made by the top layer, which considers network capacities and estimated network loads and solves a Network Load Problem. The network flow control layer uses this information, in combination with predicted platoon flows, to compute target green times for the local controllers. The third and last layer, computes phase change epochs based on the target timings and predictions of individual vehicle arrivals (Mirchandani & Head, 2001).

CoSiCoSt, abbreviation for *Composite Signal Control Strategy,* is an urban traffic control system developed by CDAC Thiruvananthapuram (Center for Development of Advanced Computing) in India. It is developed to be able to control the inherently heterogeneous traffic in India. The system divides the network into sub-networks or corridors that share a common cycle time. Along corridors offsets are set to provide green waves. The split times are decided in a two-level process. A central controller proposes stage timings to the intersection controllers based on the demand trend analysis. After this, the local intersection controllers have limited autonomy to modify the split timings based on actual measured traffic conditions. Loop detectors are used to measure presence of vehicles to make green extension decisions. If queues exist after maximum service phase, the system flags the phase congested and provides additional green time in the next phase. If simple green extensions are not enough to disperse all queues, the cycle time is increased (CDAC, 2014).

3.4 Review of current scientific literature

Next, we consider the recent scientific approaches in the field of urban traffic control. Because most of the commercial solutions rely on 1970s and 1980s technology, it is important to map the more recent theoretical developments. These methods are likely to appear in the future urban traffic management systems. As the newer methods reflect the developments in theoretical research, computational tools and computer communications capabilities, they serve a better starting point for our control method. We do this by conducting a systematic literature review.

Most imminent conclusions of the state of the current literature is that a wide variety of methods have been proposed. This reflects both the importance of solving the urban traffic problem and the applicability of various recent computation tools. Because of this, the current research is also scattered in various publications. In addition, most of the published work is found in various conference proceedings. This is a sign of the infancy of the field, and the relative novelty of many of the solution approaches. No standard forum, apart from the journal *Transportation Research,* has emerged, and the various solution approaches are often reported in the journals or conferences of their respective fields.

3.4.1 Decentralized policies

The most obvious development in the recent literature is the increase in the popularity of fully local, or *decentralized*, solution approaches. This development is in accordance with literature reviews of Bell (1992) and Papageorgiou et al. (2003) calling for research in such direction. Most of the existing commercial solutions, such as SCOOT or SCATS, take either fully central or hierarchical approach. This means that the system maintains knowledge of the whole system state, and bases its optimization in finding the best global solution. Many of the current papers challenge the feasibility of such approach.

For example, Lämmer and Helbing (2008) argue that the variations in the control decisions affect the traffic conditions, which again affects the optimal control decisions. Because of this feedback loop, the systems are essentially very large complex non-linear systems that evolve in a way that is impossible to predict in long run. Another problem with global information and optimization is the "curse of dimensionality". As the number of intersections in the system grows linearly, the size of the search space, state space, and action space grows exponentially. In addition, the centralized solutions assume that all information is centrally available causing pressure to the functionality and reliability of the communication network. As a consequence, various recent papers propose using local control policies that rely only on locally available data to make efficient control decision. In these methods, the dimension of the problem grows linearly, and the system relies less on the reliability of the communications network between intersections.

Local self-regulation

One of the first attempts to control an intersection in a decentralized and self-organized manner was proposed by Gershenson (2007). In his approach each traffic light is controlled by an intelligent self-organizing software, called an agent. The intersections do not communicate with each other but use only local rules to make control decisions. The basic control method is counting the vehicles. When the number of vehicles on an in-road exceeds a set threshold, the road is served by giving it green light. This basic model is extended by adding a minimum green time that reduces the number of quick phase changes that could lead to overall capacity deterioration. Finally, Gershenson also proposes using detectors to estimate gaps in the traffic flow to detect the end of a platoon. In this control version, a road is served after it has accumulated enough vehicles and green light is shown until the whole platoon is served or a predefined maximum number of vehicles have passed the intersection.

A very similar approach of using strong local optimization methods that gives rise to implicit control and beneficial network behavior, is the self-stabilizing control approach of Lämmer and Helbing (2008). The authors use the self-regulation phenomenon of uncontrolled pedestrian flows as inspiration and develop an anticipatory queueing formulation that is then used to assign each traffic movement a priority, or "pressure". As vehicles arrive to the queue at the intersection the pressure related to that traffic flow increases. When pressure grows high enough, switching to this phase becomes more profitable than the penalty incurred by the yellow phase, and green time is given to the traffic flow. In a later work Lämmer and Helbing (2010) expand their control model with a central *inspector* module that is used to alleviate the myopic nature of the control and improve the network wide coordination between the traffic lights. Most of the time the inspector allows the self-regulation to take place but interrupts when global control decisions need to be made.

Burguillo-Rial et al. (2009) propose a simpler version of Gershenson's (2007) self-organizing control. They argue that counting the exact number of vehicles approaching an intersection is difficult and expensive. The authors use simple stop line detectors to count the vehicles that are served during an active green phase. This measurement is then used to implement self-organizing control. The green time allocated to a phase in the next cycle is directly proportional to the number of vehicles served during the last service cycle. The traffic control is reactive, but very simple to implement.

The key advantage of local self-regulation is the adaptiveness and simplicity of the control. Decision making process simply compares the lengths of the queues at the intersection. When any of the queues grows long enough, it is discharged by assigning it the right of way. This enables also higher degree of adaptation to the short-term local variability in the traffic. Because the traffic lights do not need to follow a parametric cyclic control scheme, the controller can better respond to actually occurring traffic conditions. In addition, the local decision making combined with the limited information that is needed places fewer requirements on the control and communication architecture. Each traffic light can be given an autonomous controller that has access to the local detector information. In Gershenson's (2007) approach no communication is needed, as there is no information that is shared between the intersections. Lämmer and Helbing's (20008) solution, in turn, requires only very simple communication of the decisions that have been made at the adjacent intersections.

The key problem with using local self-regulative traffic light control is the lack of coordination between the intersections. The methods do not consider network information or consider network-wide performance. Self-regulation can, for example, send vehicles to areas that are already severely congested, although the optimal decision might be to let the area clear the existing traffic first. Moreover, the basic form of self-regulation is not able to create green-waves, which are known to improve the progression of traffic. Since the decisions are based on the queue length at the in-roads of the intersection, arriving platoons will only be taken into account as they join the queues. Lämmer and Helbing (2008) are able to create progression through using anticipatory queue lengths in the decision making. This, however, changes the key imperative of simple self-regulation, and requires more complicated predictive models and involves exchange of information between the intersections. Without additional constraints, self-regulation can lead to blocking and starvation, by denying the right of way from traffic flows with low intensity.

Backpressure routing

Probably the most widely used local control method proposed by many recent papers is the so called *backpressure algorithm* (see for example, (Varaiya, 2013) or (Le, et al., 2015)). The algorithm was initially developed for optimizing the stochastic routing in a wireless packet switching network by Tassiulas and Ephremides (1992). The algorithm is also called *max pressure algorithm* by (Varaiya, 2013) and *maximum differential backlog* by (Yeh & Berry, 2007). The various names of the algorithm reflect the operating principle of the algorithm. In a similar way to the approach of Lämmer and Helbing, the queues of vehicles at the intersection are seen to incur a pressure to the intersection. As the name used by Yeh and Berry (2007) suggest, the pressure related to a traffic movement is the difference between the queue length at the intersection and the weighted queue lengths at the downstream intersections. The turn fractions are used as weights and essentially the algorithm computes the difference in the queue length at the intersection and the expected queue length at the downstream intersections. The algorithm then picks the traffic movement with the highest backlog (pressure) for service. As the vehicles are served from the active queue and other vehicles join the queues on the other queues, serving another traffic movement becomes profitable and the algorithm changes to serving that movement.

Since the backpressure has been considered by many researchers, there are various extensions to the basic algorithm. Essentially, these extensions try to address the problem of inherent myopia of the algorithm. In its basic form, the algorithm uses only local information to make local decisions without any considerations to the global state of the traffic network. Gregoire et al. (2015), for example, propose supplementing backpressure with consideration on the finite link capacity. Traditional backpressure algorithms do not consider the spatial queue lengths on the adjacent links and the link capacities. If a traffic flow has locally high pressure, as a flow that corresponds to the most capacity critical traffic flow probably does, the algorithm can decide to use adjacent link that is already operating at full capacity and cannot receive more vehicles. Another such extension is the *cyclic phase backpressure* algorithm proposed by Le et al. (2015). This approach makes cyclical green time allocation decisions based on the pressure differences. Consequently, this approach guarantees each phase a minimum service within each cycle.

As in the case of local self-regulation, the advantage of backpressure control is its simplicity. Each controller controls a traffic light and can make decisions independently and using local information. Backpressure improves slightly the myopia of self-regulation by considering the queue lengths also at the downstream intersections. This enables more complete picture of the local traffic conditions. Moreover, because backpressure traffic control and backpressure routing have received extensive research interest, the mathematical foundations of the approach and its characteristics are well understood.

Backpressure control does come with its own problems. The standard form of the algorithm assumes infinite link capacities. Like with local self-regulation, this can lead to suboptimal traffic control decisions of sending traffic on links that are already congested. Because the control method is based on queue lengths, the basic form of backpressure algorithm can easily lead to unfair control decisions, leaving low intensity traffic consistently without green light. These problems, however, have been addressed in Gregoire et al. (2015) and Le et al. (2015).

Scheduling based traffic control

Xie et al. (2012) propose modeling the single intersection control problem as a *single machine scheduling problem*. Their approach aggregates arriving vehicles in clusters, with arrival time and duration, referring to the time taken by this platoon to traverse the intersection, assigned to them. They propose considering these clusters as jobs in the scheduling problem, which allows the application of vast operations research literature on the traffic control problem. Clustering and scheduling approach allows dynamic programming like near-optimal control, but an efficient way of restricting the solution space. Control of the network is decentralized as each intersection constructs their own schedule. In order to estimate the arrival of clusters, nearby intersections exchange their schedules and detector information. In Smith et al. (2013) the authors apply the method in a real-world implementation, with promising results.

Common to all the decentralized systems is that the network control is *implicit* and *emergent*, i.e. none of the parts of the system can individually decide the complete future of the system. Global system behavior emerges from the individual and possibly myopic decision making. This causes challenges on the coordination of adjacent traffic lights. No formal part of the algorithm coordinates the decisions made at the individual intersections. Coordination occurs in the underlying structure of the decision making algorithm. For example, the self-organizing traffic lights of Lämmer and Helbing (2008) can create green waves by extending the prediction horizon. Because anticipated pressures are used, rather than measured vehicle counts, the traffic lights can anticipate arrivals of arterial platoons and green waves can emerge.

Although the decisions that are made using only local knowledge are probably not optimal for the whole network, they can have other good characteristics. The backpressure algorithm is known to be stable in the original packet switching context. In this case *stability* guarantees that, if the capacity of the network is sufficient in relation to the demand, the policy will not lead to infinite queue lengths. This property is formally proven for the backpressure algorithm by Le et al. (2015) and Varaiya (2013). In the case of self-organization, Lämmer and Helbing (2008; 2010) suggest only heuristic arguments for the stability of their approach.

Common to all of these approaches is that they apply the decentralized architecture on the taxonomy of van Katwijk (2008). The decision variables used are both cyclic and a-cyclic. The self-regulation, the schedule driven control, and the pure form of backpressure all try to find the best possible control sequence by relaxing the requirement of cyclic control. As Xie et al. (2012) points out, a-cyclic control can achieve greater degree of adaptability and facilitate better optimization, because the solutions are not tied by the parametric cycle requirements. The cyclic extension of backpressure by Le et al. (2015) constructs cyclic plans. In addition, the self-regulatory methods use additional, cycle-like, constraints, such as maximum green or red times, to guarantee that each road gets served eventually.

The concept of planning horizon and update frequency varies between the approaches. Whereas in the schedule driven control the length of the planning horizon has very concrete meaning and it determines the complexity of the scheduling problem, backpressure and self-regulation controls consider only current actual queue lengths. An exception to this is the approach of Lämmer and Helbing, whereby an anticipatory queueing model is used to determine the future arrivals to the intersection.

The key advantage of decentralized control schemes is their scalability. Because decisions are made only on the local level, the size of the search space grows linearly in the number of intersections, as opposed to the exponential growth in the state space on the network level. In addition, due to the use of only local information, the decisions can be made quickly and the possible communication bottlenecks can be solved. The reduced amount of communication helps the system to be more robust against communication errors. In addition, the installation of such system in the real-world network is easier, as fewer communication links need to be built.

The major disadvantage of a decentralized control architecture is that the interaction effects in the traffic light network are ignored. Because the traffic lights are relatively close to each other, the arrivals of vehicles at an intersection and the subsequent optimal control decision clearly depend on the control decisions made at the adjacent intersections. Usually this information is not available or it is not used in the decentralized architectures. Moreover, decisions that would provide globally optimal behavior are out of the reach of the “short-sighted” local optimization.

3.4.2 Computational Intelligence

Many of the recent approaches in urban traffic control come from the fields of computer science, especially artificial intelligence (AI) and computational intelligence (CI). These papers recognize the complexity of the traffic control problem and argue that computational intelligence can try approaching the problem in a similar way a smart human controller would. Balaji and Srinivasan (2010) argue that because the traffic detector data is fundamentally distorted by random noise, conventional mathematical tools fail, but computational intelligence methods provide a way to analyze the data and control the traffic. Some of these approaches include *genetic algorithms* (see for example Turky et al. (2009); Chin et al. (2011)), *artificial neural networks (ANN)* (see for example Srinivasan et al. (2006); Dai et al. (2011)), *ant colony (ACO) and particle swarm optimization (PSO)* (see for example García-Nieto et al. (2012) and *fuzzy logic and fuzzy control* (see for example Mu et al. (2010); Talab et al. (2013))

Common to all of these solution approaches is that they model the urban traffic network as a black box. The mathematical structure of the control problem is analyzed only superficially, and a simplified objective function is used. The solution imperative is the application of stochastic search, excessive computing power, and computer learning algorithms. Usually, the methods involve defining a simple objective function, say waiting times at the intersections, as a (simple) function of the current network decisions and states. In addition, the structure of the decision variables has to be well-defined, usually it also has to be defined as to conform the limitations of the chosen solution method. The near-optimal solution value is then found using data-driven and computation intensive search. Another pivotal and mutual characteristic of the artificial intelligence methods is that is that they are inspired by natural organisms and the adaptive search occurring in natural processes. The key argument is that natural organisms are able to react intelligently to changes and unforeseen events. Various mathematical models have been devised to replicate this ability.

Evolutionary algorithms

Genetic algorithms are random search algorithms that mimic the reproduction and evolution occurring in natural organisms. A candidate solution called chromosome, represents an individual of a population. The structure of the solutions, for example, a vector of numbers, describes the “genes” that have been inherited from the previous generations. The objective function is used to measure the “fitness” of the solution. Like in nature, the best performing solutions have a higher probability of finding a reproductive partner, i.e., another solution to the problem. In reproduction the two solutions exchange their structural information to, in combination with some degree of alteration of the information (mutation), create an individual of the next generation. The key assumption is that this new solution probably possesses good characteristics from both parents, and in general the next generation is supposed to contain better candidate solutions than its parents.

The key design choice when using genetic algorithms is to choose the way the decision variables are represented. In addition to this, also the way the *fitness*, i.e. the objective value, related to the solution candidate is computed has to be chosen. After this, the way genetic algorithms are designed takes care of the optimization process. Turky et al. (2009) consider a simplistic situation of a single 4-way intersection with vehicle and pedestrian movements. The chromosome represented in binary digit encodes the green and red times in the intersection. The fitness value is computed by running a simple cellular automaton simulation measuring the number of vehicles and pedestrians waiting in the queues. In another approach by Chin et al. (2011), two adjacent intersections are considered, and the chromosome contains values for offset, cycle time, phase sequence, and split times for both of the intersections. The fitness of the candidate solution is computed by running a microsimulation model to measure the distance the vehicles have travelled and the total waiting time encountered by the vehicles.

Genetic algorithms belong to a larger group of *evolutionary algorithms*. The optimization process starts with very bad solution candidates but through variation operations, the candidates are slowly evolved into good solutions. Another group of evolutionary algorithms that have been used to solve the urban traffic control problem are the swarm algorithms ACO and PSO. These algorithms try to model the way large groups of relatively simple animals, such as ants or bees, exhibit higher-order complex behavior and apply this to the optimization problem. Like genetic algorithms, they belong to the set of evolutionary algorithms. The common analogy of the optimization search is the way the relatively simple animal colonies such as ants, find food in their environment. The ants depart from the nest to seemingly random directions and walking a straight line. If an ant encounters food it picks it up and carries back to the home nest. At the same time, the ant leaves behind a trail of chemicals that can be sensed by other ants. When encountering a trail of this *pheromone* an ant follows it with a probability that is proportional to the strength of the trail. Consequently, if an area with food is found somewhere, slowly the trail is marked by more and more ants eventually leading to all ants following the trail to the food source.

García-Nieto et al. (2012) apply particle swarm optimization to the problem of finding phase timings for the traffic lights in an urban region. They reduce the urban traffic control problem into finding green times of the predefined phases of the intersections in an urban region. The paper proposes using a PSO algorithm to conduct an evolutionary search to find near-optimum control parameters that optimize a compound objective value consisting of vehicular throughput, waiting times, and the number of vehicles that have to stop. The fitness of each candidate control decision is evaluated by running the SUMO microsimulation tool, with the current solution candidate as the control input.

The evolutionary search algorithms have a number of advantages. The optimization algorithm is simple to implement, the relatively unrestricted definition of the decision variable allows modeling the complex control parameters of traffic lights, and the evaluation of the fitness value of a candidate solution is accurate. Using microsimulation to evaluate the goodness of a solution makes modeling the traffic unnecessary, and provides the best possible estimate of how actual traffic would react to the control decisions. This, however, is at the same time the biggest impediment on the application of evolutionary algorithms. Running a simulation model takes a significant amount of time, and the simulation has to be run for each candidate solution at each simulation step. Consequently, the simulation model has to be executed possibly thousands of times. A very similar approach, called *simulation-based optimization*, by Osorio and Bierlaire (2015) found that the solution time for a reasonably sized urban area is in the range of several minutes. As a result, evolutionary search does not seem to fit the real of real-time control of traffic lights.

Artificial neural networks and fuzzy logic

Artificial neural networks try to construct a mathematical model of the human brain. Although not as fast as a computer in routine computation tasks, a human brain can easily handle complex decision problems involving vast amounts of data. Artificial neural network is a large hierarchical set of non-linear threshold functions, called *neurons* that mimic the way natural brain cells work. Each neuron receives input, computes the response based on an *activation function*, and transmits, or “fires”, its response. By connecting neurons with each other, the output of one neuron becomes the input of another, a complex network consisting of *layers* of neurons can be constructed. The input data is fed to the network as input to the first layer of neurons. The transformed data is read as decisions from the output of the last layer of neurons. Each of the neurons has an individual set of parameters that can be adapted to provide desired output, i.e. the activation of the threshold function with a certain input. By adapting the parameters of each of the neurons separately, a network can be *trained* to provide desired output with a given input. A large number of interacting neurons can map very complex relations between the input and output data. In general, it has been shown that given enough neurons in the network, an artificial neural network can map any arbitrary objective function.

The artificial neural network approach of Dai et al. (2011) uses a traditional three-layer network (input, “hidden, and output layers) to make decisions concerning the termination or extension of currently active green phase. The input layer takes current phase times, current phase state (red or green) and number of vehicles waiting in each phase as input values. These values are then mapped through the hidden layer to simple extend/terminate decision on the output layer. The network is trained on-line using reinforcement learning. During a control loop, the neural network makes control decisions. The goodness of these decision is measured as the derivative of the number of vehicles in the intersection. Based on the goodness of the last decision, the network is either rewarded or penalized for associating the decision output to the input data.

Fuzzy control, in turn, relates to the use of mathematical field of fuzzy logic. It models input signals continuously between 0 and 1, as opposed to the digital logic that assumes value can be only 0 or 1. The control decisions are chosen by simple predefined decision rules that map the fuzzy system states to the fuzzy control decisions. Fuzzy control has been widely used in traditional machine control, and thus the extension to the control of a network of traffic lights seems reasonable.

The approach of Talab et al. (2013) considers a traffic actuated setting, where an isolated 4-way intersection is divided into two phases. Both phases have their own fuzzy controller, which are used to make phase extension and termination decisions. The controllers take queue lengths and vehicle arrival rates at both phases as input. A simple decision logic describes the phase extension time as function of the current fuzzy state of the system.

Mu et al. (2010) try to take the network aspect and the network information into account. Again, the fuzzy controllers are distributed and have the control authority over individual intersections. Each controller can, however, collect information from the adjacent intersections to supplement the local information. The controller computes necessary green time based on the measured queue length. Information on the queue length at the adjacent intersection is used to compute extensions, either positive or negative, on the green time. The input values are "fuzzified" by aggregating them into groups described by linguistic values. For example, arrival rates can achieve values from the set:

$$arrivals = \{very\ few, few, medium, many, too\ many\}.$$

Similar fuzzification is done to the values relating to queue lengths, and green time extensions. Next, the authors define decision rules relating the fuzzy system state to the decision. The paper complements the basic control method by implementing a *special case controller* that makes control decisions if the traffic conditions differ greatly from the normal situation.

Fuzzy control makes controlling traffic very simple. The complexity of the decision making is significantly reduced by aggregating the state space to fuzzy groups. This makes complicated models for the traffic flow unnecessary. Fuzzy logic captures also the fundamental nature of the sensory measurements. Measurements are always stochastic estimates of the true values, and this is especially true in the case of Indian traffic. In addition, the need of mathematical optimization tools is eliminated. The control decisions are defined in a simple fixed mapping between the states and the control decisions.

Obviously, the simplicity is the greatest weakness of the fuzzy control scheme. The system loses a lot information by describing the sensory measurements using fuzzy aggregate values. Moreover, the aggregation is arbitrary and has to be done by the designer of the system. Because the fuzzy variables and the decision logic are fixed, the controller cannot adapt to neither short term nor long term variations in the traffic dynamics.

The papers that propose the use of computational intelligence attribute its usefulness to multiple factors. Firstly, the urban traffic systems are usually very large complex non-linear systems. Consequently, it is very difficult to find optimal solutions to the control decision problems. CI provides a simple way to come up with near-optimal solutions. Secondly, CI is data driven, and does not need precise mathematical models of the traffic. Lastly, because CI solutions model adaptive systems by their nature, they seem like an obvious choice for an adaptive traffic control system. A key advantage of the CI techniques is that due to their flexibility the methods can easily be combined. One example of this is the so-called multi-agent system of Balaji and Srinivasan (2010), whereby neural networks are used in a decentralized control hierarchy to make individual control decisions in each of the intersections.

Although the CI methods offer promising benefits in traffic control, the very operating principle of these systems becomes their biggest limitation and problem in controlling a large network of traffic lights. Artificial neural networks, for example, use historical data to learn a way to infer control decisions from the traffic data. This means that they are inherently incapable of handling dynamically changing traffic situations. Changing traffic demand would require retraining the network to respond correctly to the new data. The problem with genetic algorithms is their computational efficiency. The convergence to a good solution can be slow, and they can realistically only solve very small problem instances leaving real-time control of any relevant network out of their reach. The particle swarm operation methods, PSO and ACO, suffer from the same intractability issue as genetic algorithms. Even the fuzzy control scheme has so far been tested only with small network instances (Ahmad et al. 2014).

3.4.3 Multi-Agent Systems

Many recent paper propose the use of the so-called multi-agent systems (MAS). These are systems that consist of numerous individual decision makers. An agent, usually a software, can make independent decisions considering the entity that it controls. It can perceive the environment, make decisions based on the state of the system, and communicate its decision to the environment. Key characteristics of an independent agent are: *autonomy, social ability, reactivity,* and *proactiveness* (Woolridge and Jennings, 1995). An agent can decide itself how to relate data to decisions. A multi-agent system is the interaction of many interacting individual agents trying to solve a common problem. (Roozemond, 1999).

Like many of the computational intelligence methods mentioned above, also the multi-agent systems are inspired by naturally adaptive processes. The key idea behind the approach is to enable “life-like” adaptability and self-organization (Protham et al., 2009). A natural example of multi-agent systems is an ant colony. Although individual ants are very simple, when interacting with each other a collection of many ants can exhibit complex behavior, such as nest building, food searching, and migration. Multi-agent systems, trying to mimic this adaptability, have been found useful in highly complex and unstructured control problems, where no adequate models for the system exist (Barcelos de Olivera and Camponogara, 2010). The approach has been applied in various complex planning and control problems, such as the barge planning problem in the port of Rotterdam (see for example, Douma 2008).

Multi-agent systems can be characterized as computational or artificial intelligence methods, and fall under the categories of *distributed AI* or *organic computing*. We consider multi-agent systems here separately, however, because they do not constitute a solution approach, per se. Multi-agent systems are simply an architecture or a framework for distributed control approach. The solution imperative is similar to the distributed approaches, namely that the problem with the optimization dimensionality and information locality is solved by making only local decisions. The difference, and the key characteristic of MAS, is that the local controllers communicate and interact with each other. This communication is the key in the rise, or *emergence*, of higher-order complexity and control, without any central control (Roozemond, 1999). The communication between agents allows the autonomous decision makers to take into account the behavior of other agents, and, thus, have more complete understanding of the current state of the problem.

The functionality of a multi-agent system is defined by the combination of agent functionality and the communication framework between the agents (Douma, 2008). The agent functionality describes the way in individual agent perceives the information at hand, and how it makes decisions based on it. The communication scheme defines the way the agents communicate with each other. The agents can, for example, share and receive information prior to the decision making, share their decisions that have already been made, or negotiate with each other concerning the current decision being made. Care must be taken not to mix up the multi-agent systems with the evolutionary search methods such as ACO and PSO. Although we used a similar analogy to explain their design and operation principles, the methods are completely different. ACO and PSO are move-based algorithms that use directed randomness to explore the search space. Multi-agent systems, in turn, are not a solution method but rather an architectural choice. The fundamental principle of MAS is the combination of autonomous decentralized decision making and the agent-to-agent communication.

Applying MAS to the urban traffic control problem is straightforward. Each unit at the lowest level of control, either an intersection or an in-roads of an intersection, is assigned an autonomous controller. These controllers are given a task to optimize the local traffic flow according to an objective function. In addition, a framework for the agent communication is defined. Intersection agents can, for example, share the traffic measurement data from their local detectors, or negotiate on establishing coordination between the intersections. Some papers add additional hierarchical levels by creating agents that are responsible for regional control or facilitating the agent to agent communication.

Because multi-agent systems themselves are not solutions to the control problem, the functionality of the agents still needs to rely on methods discussed above. In the current traffic management MAS literature, the agents have been given various control methods. For example, Smith et al. (2013) applies the schedule driven intersection control proposed by Xie et al. (2012) to make the agent decisions. The agents communicate the vehicular outflows that result from the local schedule to the adjacent intersections to allow prediction of traffic flows in the near future. Benhamza and Seridi (2015), in turn, use very simple ranking of traffic movements to choose next active phase and its duration.

The majority of the MAS approaches use the computational intelligence methods to implement the agent functionality. Balaji and Srinivasan (2010), for example, use artificial neural networks at each intersection to learn the traffic dynamics and make control decisions. Another common approach is the machine learning algorithm *reinforcement learning* used by Balaji et al. (2010), Prothman et al. (2009) and El-Tantawy et al. (2013). Barcelos de Oliver and Camponogara (2010) combine reinforcement learning based multi-agent system with model predictive control. Common to all of these approaches is that they try to use statistical methods to "learn" to make good control decisions. When the method comes up with a good control decision, a reward is issued and the optimization parameters are adjusted to produce similar behavior in the future.

We considered multi-agent systems here because they appear often in the recent traffic control literature and they have recently been applied in similar complex planning problems in other fields. Because MAS alone is not an urban traffic control system, we cannot place it on the framework of van Katwijk (2008). MAS is rather a new design choice in the *architecture* dimension of the framework. On one hand, multi-agent systems exhibit decentralized traits by implementing autonomous local control. On the other hand, the key emphasis on communication and coordination brings MAS closer to centralized and hierarchical systems. Multi-agent systems, however, cannot be considered to be hierarchical systems as the emphasis of the communication is on lateral rather than vertical communication, and the coordination arises implicitly from mutual goals and preference of the local controllers.

3.5 Results of the literature review

The recent literature seems to have taken the approach of challenging the feasibility of the traditional centralized network control. Some papers, most notably Lämmer and Helbing (2008;2010) and Gershenson (2007), argue that due to the computational complexities, traffic light control is out of the reach of any optimization method, as they usually require a considerable amount of computation time. Consequently, they propose using adaptation instead. This means that the traffic lights try to capture the short term variation in the traffic flow at the intersection level, and use only very rudimentary decision-making logic. Instead of complicated mathematical optimization models, simple decision policies are devised.

Another direction in the recent academic traffic management literature is the application of various computational intelligence methods in the urban traffic control problem. This reflects the recent improvement in the computer and computer communication capabilities. These methods usually fail in that the modeling and system assumptions used are unrealistic. Because the underlying traffic models are very simplistic, the models lack in their accuracy and their adaptation to short-term variation in the traffic. In addition, computational intelligence methods are computationally demanding and are often intractable for real-time applications.

Multi-agent systems, or distributed AI, try to combine the simplicity of decentralized or local control methods, the additional information processing provided by the centralized approaches, and the adaptivity of natural organisms. The key difference between multi-agent systems and pure hierarchical traffic control approaches, such as RHODES, is that the key emphasis of the multi-agent adaptation is in the collaborative decision-making and communication. Due to the improvements in computers and computer communication networks, there have been recently many applications of multi-agent systems to the urban traffic control problem.

One key conclusion from the literature review is that any urban traffic control method has to consider the trade-off between utilization and coordination. Coordinating traffic lights is easily achieved by using fixed time control with properly determined offsets. These methods, however, fail in the green time utilization as the cycle- and green times are computed for average traffic loads. Systems that try to maximize the utilization at the intersections, for example, vehicle actuated systems, fail to maintain coordination between the intersections, because the start and end of a service phase can vary from cycle to cycle.

Most of the existing systems and the recent methods in the literature consider the queue lengths at the intersection as the system state. Decisions are made to minimize the future queue lengths or queue lengths are used in real-time to allocate green time to phases. Various ways are used to deduce this information. The existing commercial systems gather this information from upstream detectors. A traffic model is fitted in the sensory output to provide predictions on the vehicle arrival times at the intersections. The scientific models, in turn, do not usually consider the way to collect this information but assume that it is known in advance and available to the decision maker.

The wide variety of the recent traffic control approaches is clearly visible in Table 3.2. We use the taxonomy proposed by van Katwijk (2008) to reflect the differences in design choices of the methods. Each paper that was identified in the above discussion is analyzed for its *architecture*, *decision variable, prediction model, planning horizon*, and *update frequency*. The symbol (??) means that the design choice is not clearly documented, cannot be implied from the report, or is not relevant for the type of control method.

We see that most of the research we identified uses decentralized architecture. The choice between *cyclic* and *a-cyclic* depends on the architecture choice. We have identified decentralized methods that use either of the design choices, but none of the centralized approaches consider the *a-cyclic approach*. This design choice combination is probably infeasible, as a centralized controller would have to continuously monitor the complete state of each of the intersections to determine a state change that constitutes a reason for changing the current control input. The update frequency is often, but not always, linked to the choice of the decision variable. Cyclic plans are usually updated after the current service cycle is completed, whereas a-cyclic control methods update either after a fixed planning horizon or "continuously" reacting to small cues in the change of the state of the traffic.

Table 3.2 Comparison of papers using the taxonomy

Paper	Control method	Architecture	Decision variable and decision	Prediction models	Planning horizon	Update frequency
Gershenson (2007)	Self-regulation	Decentralized	A-cyclic (next phase to be served)	Expected arrivals, upstream detector	Variable length, variable interval	Continuous, short time interval T
Lämmer & Helbing (2008; 2010)	Self-regulation	Decentralized	A-cyclic (next phase to be served)	Expected arrivals, upstream detector + traffic model	Variable length, variable interval	Continuous, short time interval T
Burguillo-Rial et al. (2009	History-based self-regulation	Decentralized	Cyclic (green splits in the following cycle)	Imminent arrivals, stop line detector	Fixed length (cycle time)	Once per cycle
Varaiya (2013)	Backpressure	Decentralized	A-cyclic (next phase to be served)	Vertical queues, expected arrivals, upstream detector	Variable horizon, variable interval	Continuous, short time interval T
Le et al. (2015)	Cyclic backpressure	Decentralized	Cyclic (green splits in the following cycle)	Vertical queues, expected arrivals, upstream detector	Fixed horizon (cycle time)	Once per cycle
Gregoire et al. (2015)	Capacity aware backpressure	Decentralized	A-cyclic (next phase to be served	Spatial queues, expected arrivals, upstream detector	Variable horizon, variable interval	Continuous, short time interval T
Xie et al. (2012)	Scheduling based intersection control	Decentralized	A-cyclic (next phase to be served)	Vertical queues, expected arrivals, upstream detectors + traffic model	Fixed planning horizon, fixed planning interval	Rolling period T
Turky et al. (2009)	Genetic algorithm	Decentralized	Cyclic (fixed cycle parameters)	Spatial queues,	Fixed planning horizon, fixed planning interval	Fixed plan
Chin et al. (2011)	Genetic algorithm	Centralized	Cyclic (fixed cycle parameters)	??	??	Fixed plan
Dai et al. (2011)	Artificial neural network	Decentralized	A-cyclic (phase extension/termination)	Imminent arrivals	??	Continuous, short time interval T
García-Nieto et al (2012)	Particle-swarm optimization	Centralized	Cyclic (fixed cycle parameters)	??	??	Fixed Plan
Mu et al. (2010)	Fuzzy control	Decentralized	Cyclic (extension time of current phase)	Imminent arrivals	Fixed planning horizon, variable interval (extension period)	Once per service phase
Talab et al. (2013)	Fuzzy control	Decentralized	Cyclic (extension time of current phase)	Expected arrivals (queue lengths)	Fixed planning horizon, variable interval (extension period)	Once per service phase

3.6 Controlling traffic in Patna

In addition to Analyzing the functionality and input data of each of the scientific methods, it is important to consider how each of the identified methods fit the specified system requirements collected in Chapter 2.6. This is important, as choosing a method that fits the needs of the problem owner is more likely to be implemented and useful in solving the actual problem. Moreover, the communication and sensor network can set hard restrictions on the operation and functionality of the system. As we have concluded earlier, our traffic control method has to be able to provide real-time solutions, adapt to the dynamically changing traffic, has to be stable and it has to be scalable in the number of intersections in the system and be robust against possible hardware and communication network failures. In addition, the system has to be able to provide good decisions that reduce the average delay. The system has to be able to deal with the limited input data provided by the virtual loop sensors used in Patna and be able to deal with the heterogeneity of the traffic. For more comprehensive discussion concerning the system requirements we refer to Chapter 2.6.

In order to support choosing the method that best fits our purpose, we score in Table 3.3 each of the recent scientific methods using the system requirements identified in chapter 2. In the scoring, we use a method similar to the well-known Likert scale. The scale we use runs from (--), indicating that the method does not fulfill the requirement well, through (-), (?), and (+) to (++), which indicates that the respective method fits well the requirement. The midpoint of the scale, represented by (?), indicates that current literature does not provide evidence to make conclusive scoring on the requirement. This analysis helps us to choose the most applicable solution method to our control problem. We summarize also the input data that the methods use as the limited information available from the sensors in Patna could restrict the usefulness of a method.

Table 3.3 Comparison of various control methods

Method	Real-time solution	Adaptivity	Scalability and robustness	Stability	Optimality	Input data
Self-organization	++	++	++	?	Heuristic local delay minimization	Number of vehicles in the in-roads
Backpressure;	++	++	++	+	Local throughput maximizing control policy	Number of vehicles in the in-roads; estimated turn fractions
Schedule driven control	+	++	++	?	Local delay minimizing control policy	Arrival distribution of vehicles at the intersection
Evolutionary algorithm	-	-	-	+	Globally optimal network controls	Microsimulation model of the traffic and network
Artificial neural networks	+	-	+	?	Local throughput maximizing policy	Number of vehicles in the in-roads
Fuzzy controllers	++	-	+	?	Local delay minimization	Number of vehicles in the in-roads

According to the comparison, it is obvious that the decentralized control methods, self-organization and backpressure routing, score the highest. The decision rules are very simple and can be solved frequently to provide real-time control. As only local information or information shared with only the adjacent intersections is used, the system is scalable and can operate even if some of the communication links or intersections are not operating as desired. These methods by their nature emphasize adaptation to the current traffic conditions over optimization and are able to react even to swift changes in the traffic flow. Like SCOOT does, these methods adapt to long-term variations in the traffic by continuously adapting to the prevailing short-term traffic conditions. Because the short term traffic conditions are a function of the long-term traffic conditions, the system implicitly adapts to the more gradual changes in the traffic.

In addition to decentralized policies, also the fuzzy control method scores well on our requirements. Because the sensor measurements, i.e. the input data, are always uncertain and essentially fuzzy, and especially so in the Indian traffic, modeling the system state as fuzzy variable is a reasonable choice. Moreover, because the fuzzy controller is essentially a simple mapping between system state and control decisions, the methods score well on the real-time requirement. After the decision rules have been defined off-line, the computational complexity of traversing a list of possible control choices is linear. Scalability depends on the definition of the controller and its input values. When each intersection is controlled using only local knowledge the network can be easily modified by adding or removing intersections from it. If, however, the controller uses information gathered from multiple adjacent intersections, adding or removing one intersection requires a reconfiguration of the decision rules. Adapting to the varying traffic, obviously, is limited as the system state can only be described using a predefined set of fuzzy states. Moreover, the decision logic is defined off-line by the system designer leading to possibly biased rules.

Many of the computational intelligence methods score low because of the awkward modeling assumptions they make and the long computational time that is needed. Overly simplistic models of the queueing process or the traffic, like in fuzzy control, leads problems in adapting to the changing traffic. Because the system does not have a proper understanding of the current state of the traffic, it also cannot make decisions that utilize short term variations in the traffic. Many computational intelligence techniques assume *a priori* knowledge of the traffic, rendering real-time control and adaptation out of the reach of these methods. On the other end of the modeling accuracy spectrum, methods that rely on simulations to evaluate a performance of the system as a function of the control decisions can utilize very accurate description of the traffic conditions and can achieve optimal control decisions even on the network level. The obvious downside of such approach is the long computation time needed to solve these models.

3.7 Conclusion

When it comes to finding a scientific method for controlling the traffic in the city of Patna, the results of our literature search are twofold. In one hand, the unstructured and heterogeneous traffic, per se, does not yield any of the control approaches invalid or unusable, and a large number of different approaches have been proposed. The methods used in the literature do not have strict assumptions on the dynamics of the traffic flow. And as we concluded in the previous chapter, on a coarse level the Indian traffic behaves similarly to the western homogeneous traffic. On the other hand, many of the control methods proposed in the literature assume the availability of accurate and ample data. In this case, the traffic heterogeneity causes problems indirectly as the sensor network in the city of Patna is unable to provide us with complex information.

In our control problem, the prediction horizon is very short, because the detectors are placed on the stop lines of the intersections. Consequently, our solution approach cannot use information on the imminent arrivals, as assumed by most state of the art solutions. Due to the detector layout and functionality, we cannot estimate vehicle counts or turn fractions in real-time. By sampling the detectors regularly, however, we can estimate gap sizes in the traffic flow. Consequently, our control method could use this information to estimate when the queue that was waiting on the in-road has been served. Queue lengths, in turn, can serve as cues on the traffic intensity and queue lengths. Such vehicle actuated control logic is commonly used throughout the world.

As suggested by the high scores in Table 3.3, the decentralized solution architecture of urban traffic control fits our control problem well. Firstly, the network in question consists of 88 intersections. A centralized algorithm to control a network of this size would be too slow to solve for real-time traffic control purposes. Secondly, such systems scale better in the number of intersections and are more robust against communication network problems that can be a burden in India. In addition, because the input information used by our system is fundamentally limited, a simplistic decision-making policy that inherently adapts to the current traffic conditions is a good choice. Moreover, the literature has shown that even local control alone can perform well and that with proper coordination mechanisms even such behavior as a green wave can be established.

Among the decentralized policies the backpressure routing provides the best control choice. Although both self-organization and backpressure are myopic and use local information, backpressure takes a larger amount of network information into account. The computation of the pressure considers the difference in the queue length at the intersection and its adjacent intersections. Thus, the algorithm has a better understanding of possible overloading of nearby intersections. This can help reduce the possibility of gridlocks occurring.

Applying backpressure control in the city of Patna does not come without its challenges. We must provide each road with a minimum service level. Consequently, we follow the cyclic-phase backpressure approach proposed by Le et al. (2015). A second problem comes from the measurement of queue lengths, the input data used by all of the papers. Because this information is not available, we use an approach similar to the one proposed Burguillo-Rial et al. (2009), whereby the control decisions for the next service cycle are based on the measurements during the last active service cycle. We allow the queue of the vehicles move and measure estimate the number of vehicles as they pass the intersection. The estimate can then be used in decision making in the next decision making cycle. By doing this, we bypass the problem of having only stop line occupancy information. Obviously, the key assumption here is that the platoons served during subsequent service cycles do not differ fundamentally from each other. This is probably true in the highly saturated traffic in Patna, and such approach of making continuously very small adjustments to the control parameters has already been used by existing commercial solutions such as SCOOT.

4 Control algorithm development

In this chapter, we develop our traffic control method and answer the third research question: *How can we control the network of traffic lights in the city of Patna?* The chapter is based on the results of the two earlier chapters, problem analysis and literature review. We start, in Sections 4.1 and 4.2, by making relevant modeling assumptions and by proposing rigorous definitions on the elements of our traffic control problem. We continue by describing the control architecture used by our approach. After this, we describe the functionality of the controllers by adapting the backpressure algorithm that we chose in Chapter 3 to fit the available data and the network in the city of Patna. We end by proposing an adaptation of a measurement algorithm that our control method uses to estimate the queue lengths.

4.1 Model assumptions

Making relevant assumptions is necessary to make development of a control algorithm feasible. Urban traffic systems are very complex systems with various different dynamics. We propose assumptions that make modeling the situation simpler, but manage to retain the key characteristics of the actual traffic situation in Patna. Our model is based on the following assumptions

1. Drivers do not change their routing as a function of the control input
2. Vehicles reach their free flow speeds instantaneously after a start delay and drive the links on a fixed free flow speed. Queues discharge at saturation flow rate.
3. Turn fractions are fixed, or measured for interval times, and are same for all incoming traffic.
4. Cycle times are fixed, or fixed for current time of the day
5. Network control is achieved using a fixed network cycle time and offsets are not used

Assumption 1 is commonly used in the traffic control literature. This assumption is reasonable as the drivers do not have complete knowledge of the system state and the control decisions made at the other intersections, and thus probably choose to follow their normal routes. Assuming this implies that we can consider traffic measurements, such as turn fractions, to be fixed for the short period of time during which we make the control decision. The more general problem of traffic control incorporating the driver routing behavior is called Equilibrium Network Design Problem (ENDP). An interested reader can see Cascetta et al. (2006) for an example of treating such a problem.

Assumption 2 makes modeling the platoon progression times simpler. This is assumed by many papers such as Xie et al. (2013). The other option is to use a formula that relates the average vehicle speed to the degree of saturation of the link. Because the free flow speed in Patna is relatively low, and we are interested in the intersection delay optimization, the assumption is reasonable. Assuming the discharge rate to be the saturation flow rate is supported by the analysis of the video recordings. The vehicles seem to reach their free flow speeds quickly and use all available road capacity in the process.

The 3rd assumption circumvents the problem of estimating turn fraction with the current detector configuration. Because in Patna with the current sensors some of the traffic movements are indistinguishable from each other at the stop lines, we use fixed turn fractions that are measured off-line. To increase the accuracy of the control method, the fractions can be measured for various time intervals, say, morning peak-hour, afternoon peak-hour, and off-peak traffic. Although the turn fractions fluctuate in short term and change in long-term, in medium-term we can assume that the turn fractions are defined by slowly changing characteristics of the network. The assumption that the turn fractions in an intersection are same for all arriving traffic simplifies the necessary measurements.

The assumption of fixed cycle times simplifies the control decisions made by the control algorithm. By using a predetermined fixed cycle time, the adaptiveness of our control approach is realized in the dynamic and adaptive allocation of the available fixed cycle time. Deciding the length of the cycle time is essentially a *capacity problem* and the decision horizon is longer. Our control algorithm makes short term *capacity allocation* decisions on a cycle to cycle basis.

The last assumption is related to the fixed cycle time assumption described previously. In addition to the split adjustment decisions, state-of-the-art urban traffic control systems are able to compute optimal offset values between pairs of intersections, use multiple cycle times within a network of subnetwork, and dynamically adjust the cycle times at each intersection. These systems, however, are results of substantial development and refinement effort. The scope of this project does not enable our control algorithm to consider each aspect of the urban traffic control problem. Consequently, we choose to use the same cycle time at each intersection. This guarantees that after a warmup period we can expect the arrival process at an intersection to resemble the arrival process at that intersection during the previous cycle. Our algorithm is also not able to compute the offset values, and thus they are not used.

4.2 Network Model

In the current literature the network of traffic lights is usually modeled as a graph of nodes and links. The nodes represent the intersections of the urban traffic network, and the links model the roads connecting the adjacent intersections. We need to adapt the standard models, however, because of awkward assumptions made by many of the papers. Firstly, many papers assume that all lanes coming from certain direction constitute a single in-road of the intersection. Secondly, most papers assume simplistic phase structure. Often the intersections are modeled as standard four-way intersections with four service phases. Either each in-road is served separately, or each service phase serves one in-road and allows one or more non-conflicting traffic moves. In our case, however, the network consists of different types of intersections. In addition, the phases reflect these differences, and the in-roads consist of multiple lanes. The lanes on a given in-road can belong to different traffic movements are not necessarily served in the same service phase. Although modeling our network is more tedious than the traditional four-way grid, our system actually contains more information about the traffic flow, as in-roads are split in lanes and different phases.

As a basis of our network model we adopt the model of Le et al. (2015). Consider a network of intersections. Let this network consists of a set of junctions, indexed by $\mathcal{J}$. Each of these junctions $j \in \mathcal{J}$ consists of a number of in-roads $\mathcal{I}_j$. Each in-road $i \in \mathcal{I}_j$, in turn, consists of a set of lanes $\mathcal{K}_i$. For simplicity, let $\mathcal{K}_j$ be the set of incoming lanes at intersection *j.* Let the intersections be connected by a set of links, denoted by $\mathcal{L}$. We use notation $kk' \in \mathcal{L}$ to indicate that there is a road that links incoming lane *k* at intersection *j* with lane *k'* at intersection *j'.*

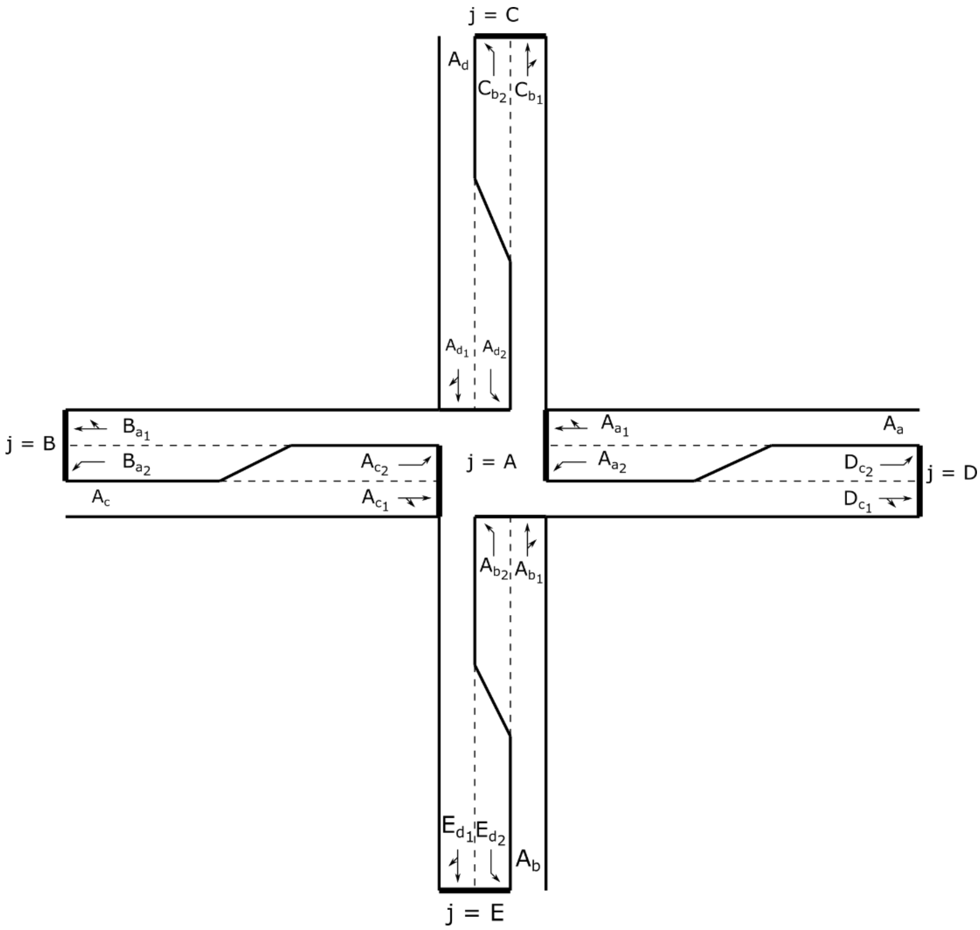


Figure 4.1 Network model

Figure 4.1 shows an example of the structure of a network consisting of 5 intersections A, B, C, D, and E. We are interested in junction A, i.e. j = A. The inroad set is the following

$$\mathcal{I}_A = \{A_a, A_b, A_c, A_d\}.$$

The set of lanes served by intersection A is

$$\mathcal{K}_A = \{A_{a_1}, A_{a_2}, A_{b_1}, A_{b_2}, A_{c_1}, A_{c_2}, A_{d_1}, A_{d_2}\}$$

The links set $\mathcal{L}$ defines the set of lanes that connect intersections to each other. For brevity, we present here the structure of only a subset of the complete set of links in the network of Figure 4.1. Consider the way intersections A and B are connected, i.e. j = A and j' = B. Vehicles departing from intersection A can reach intersection B using the following connections:

$$\mathcal{L}_{A\to B} = \left\{\{A_{a_1}B_{a_1}\}, \{A_{a_1}B_{a_2}\}, \{A_{b_2}B_{a_1}\}, \{A_{b_2}B_{a_2}\}, \{A_{d_1}B_{a_1}\}, \{A_{d_1}B_{a_2}\}\right\} \subset \mathcal{L}$$

We call the combination of lanes that are simultaneously established the right of way a *service phase*. We represent a service phase for junction *j* with a vector $\Omega = (\sigma_k : k \in \mathcal{K}_j)$, where σ_k is the service rate, i.e. the number of vehicles that depart the lane in unit time, of the vehicles served from lane *k* at intersection *j.* The service rate can be interpreted as follows: if a lane is assigned green light during service phase Ω, then $\sigma_k > 0$, otherwise $\sigma_k = 0$. From our earlier model assumption, we know that the service rate is equal to the saturation flow rate of the lane. The set of all service phases, or the phase plan, at intersection j is denoted by $\mathcal{S}_j$. We use the intersection A Figure 4.1 to illustrate the structure of the set $\mathcal{S}_j$. We use the phase plan that is presented in Figure 3.4.

$$\mathcal{S}_A = \{\Omega_1, \Omega_2, \Omega_3, \Omega_4\},$$

Where

$$\begin{aligned}
\Omega_1 &= \{0,0,\sigma_{A_{b_1}},0,0,0,\sigma_{A_{d_1}},0\} \\
\Omega_2 &= \{0,0,0,\sigma_{A_{b_2}},0,0,0,\sigma_{A_{d_2}}\} \\
\Omega_3 &= \{\sigma_{A_{a_1}},0,0,0,\sigma_{A_{c_1}},0,0,0\} \\
\Omega_4 &= \{0,\sigma_{A_{a_2}},0,0,0,\sigma_{A_{c_2}},0,0\}
\end{aligned}$$

4.3 Decentralized control architecture

We concluded in chapter 3 that the traffic control approach that fits the needs of ARS and the city of Patna the best is the cyclic-phase backpressure approach. This method belongs to the class of decentralized control policies, as each intersection makes its own control decisions autonomously. In addition, the information used in the decision making is collected from the intersection itself and the nearest adjacent intersections. We chose this method, because it provides flexibility and scalability to our control system. The decentralized decision making and information also provides higher level or robustness against possible hardware and communication failures; a possibly failing intersection does not affect the operation of the controllers at other intersections.

To implement the decentralized control algorithm, also the controller architecture has to be decentralized. This means that each intersection is given a controller that decides the state of the traffic lights in the intersection. In addition, this controller also manages the virtual loop detectors that are installed in the intersection. Because our control algorithm uses queue lengths at the adjacent intersections as input values, our controller has to be able to communicate with other intersection controllers in the system.

As our algorithm makes autonomous local decisions but still communicates with its neighbors, it is easy to see it as a multi-agent system. Each intersection is assigned an intersection agent with the goal of maximizing local throughput. The agents communicate with each other to share their knowledge of the current traffic conditions. This information is shared in the form of platoon length estimates that the detector agents at the intersection measure. We assume that this collaboration results in implicit network control. This is also suggested by the theoretical results that show the stability of the backpressure control.

The multi-agent architecture makes implementing the system simpler. We can create a universal controller that fulfills the requirements and simply install copies of it to each of the intersections. Splitting the functionality into smaller blocks also improves the robustness and the scalability of our system. The maintainability of the system, however, can become an issues, as updating the control system requires updating each of the intersections separately.

The approach of splitting system into smaller functional blocks, often called modularization, can be taken further on the lower levels of the system too. This means that we split the actual intersection controller agent into modules. These modules reflect the functionality requirements we discussed above. Such a modular design makes changing and improving the controller easier. The functionality of the controller is defined by the interaction of the individual modules, rather than the actual implementation of the modules, any modules can be changed without affecting the functionality of the other parts of the system. In order to fulfill the requirements above, we propose that the intersection controller agent has the following modules:

- Data collection module
- Communication module
- Decision logic module
- Traffic state estimation module
- Control actuation module

The process of making traffic control decisions starts at the data collection module. This part of the controller collects and maintains a database of both local and network information. The local estimates of the platoon lengths are fetched from the detector agents controlling the virtual loop detectors at the intersection. The network information contains the most recent detector measurements fetched from the adjacent intersection. The communication with other intersection agents is implemented in the communication module, which takes care of sending and receiving the detector measurements.

The decision logic module makes the traffic control decisions based on the input from the traffic state estimation module. The state estimation module computes the estimate of the current queue lengths on the in-roads of the intersection and on the in-roads of the downstream intersection based on historical detector measurements. The decision logic module is the key to adaptive traffic control as it implements the control algorithm that we propose later in this chapter.

The last part of the intersection agent architecture is the signal actuation module. This module translates the output of the decision logic module into control commands to the actual traffic signals. In our architecture, this module implements fixed traffic light control, based on the parameters decided by the decision logic module. This module also implements the software to hardware connection that allows our algorithm to interact with real physical world.

The structure of the traffic light agent is presented in Figure 4.2. The direction of the black arrows represents the direction of the flow of information between the various modules. We see that in addition to the traffic light controller agent, an intersection also contains detector agent. The purpose of this agent is to measure the estimated platoon lengths using the virtual loops that are installed in the intersections. It communicates this information to the data collection module in the traffic light agent. The environment outside of the intersection represents the state of the urban traffic system. This is the combination of the state of the other intersection control agents and the state of the actual traffic. The control actuation module affects the state of the environment by affecting the state of the traffic through the traffic signals it operates. The state of the environment can be partially observed by the intersection control agent. The agent can observe the state of the local traffic through the detector agents. In addition, the agent receives a measurement of the state of the local traffic at all its adjacent intersections. The communication module has bidirectional information flow. It transmits information to and receives information from the environment.

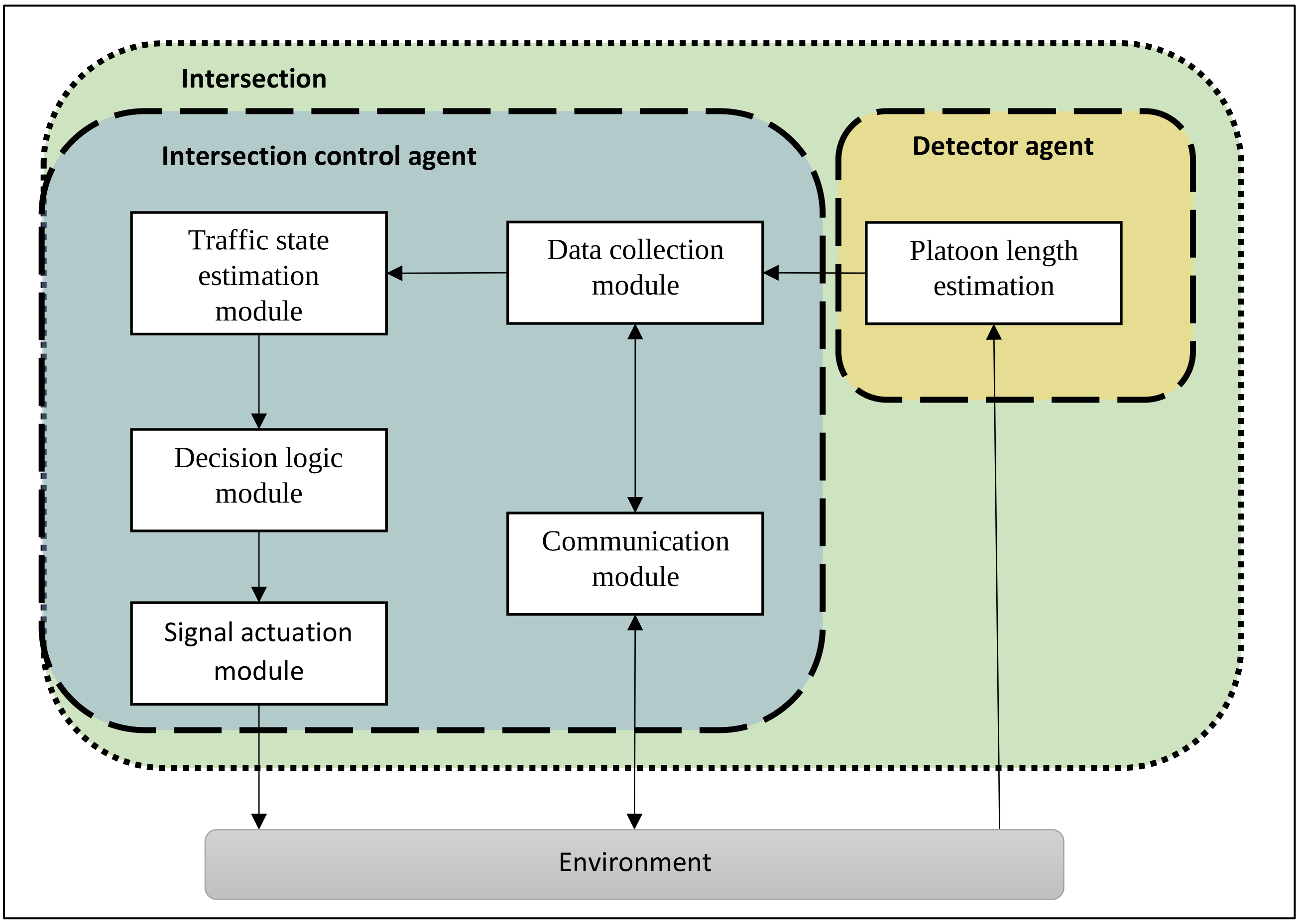


Figure 4.2 The structure of the intersection control and detector agents

4.4 Intersection controller

As we concluded in chapter 3, we base our decision making logic on the cyclic-phase backpressure method proposed by Le et al. (2015). In the network control architecture that we proposed earlier, this method is implemented in the decision logic module of the intersection control agent. The standard backpressure algorithm assumes that the controller is able to construct an unbiased estimate of the actual number of vehicles at each approach of the intersection at any given time. Because of the limitations imposed by the characteristics of the detectors and the characteristics of the traffic, as concluded in chapter 2, this information is not available to our control method. Consequently, we use Burguillo-Rial et al. (2009) as an inspiration to propose a method we call *history-based cyclic-phase backpressure algorithm.* This method uses the traffic measurements from earlier service phases to construct an estimate of the current traffic conditions at the approaches of the intersection. With this estimate of the queue length, we can use the backpressure algorithm to make the control decisions. In addition, we generalize the approach of Le et al. (2015) to fit more general intersection topology and service phase structure.

The backpressure algorithm models the number of vehicles waiting at the intersection as a pressure that is incurred to the intersection. We call this the upstream pressure. The downstream pressure to the intersection is incurred by the queue lengths at the downstream intersections. The backpressure algorithm computes the difference between upstream and downstream pressure for each service phase and serves the phase with the largest pressure gradient. In the cyclic-phase backpressure algorithm the magnitude of the pressure difference is used to allocate proportions of the available green time to the service phases.

The first step in backpressure algorithm is to compute the pressure difference (the weight) associated with each of the phases. Because our network model differs slightly from the one used by Le et al. (2015), we must adapt the formula used to determine the pressure of a phase. The pressure of a phase now depends on the length of queue at the lanes served by a phase, rather than the length of queue on the in-roads. The pressure associated with service phase Ω at intersection *j* is given by:

$$w_{\Omega}\left(\hat{Q}(t)\right) = \sum_{k\in\mathcal{K}_j} \sigma_k \{\hat{Q}_k(t) - \sum_{k':kk'\in\mathcal{L}} \hat{p}_{kk'}(t)\hat{Q}_{k'}(t)\} \tag{3}$$

where $\hat{Q}_k(t)$, is the estimate of the number of vehicles at lane *k* at time *t* and $\hat{p}_{kk'}(t)$ is the probability that a vehicle at intersection *j* and lane *k* will move to intersection *j'* and lane *k'*. After we have computed the pressure for each of the service phases σ at intersection *j* we allocate the available cycle time to the phases. The proportion of the cycle time at intersection *j* allocated to service phase σ is given by:

$$P_{\Omega}^{j} = \frac{e^{\eta w_{\Omega}(\hat{Q}(t))}}{\sum_{\pi\in S_j} e^{\eta w_{\pi}(\hat{Q}(t))}}, \tag{4}$$

where $\eta > 0$ is a parameter of our model. To interpret the effect of the parameter η, we consider the range of values it can achieve. First, assume that η approaches the lower limit of its value, i.e. $\eta \to 0$. Now, the distribution of cycle time behaves as follows:

$$\lim_{\eta\to 0} P_{\Omega}^{j} = \lim_{\eta\to 0} \frac{e^{\eta w_{\Omega}(\hat{Q}(t))}}{\sum_{\pi\in S_j} e^{\eta w_{\pi}(\hat{Q}(t))}} = \frac{e^{0* w_{\Omega}(\hat{Q}(t))}}{\sum_{\pi\in S_j} e^{0* w_{\pi}(\hat{Q}(t))}} = \frac{1}{|S_j|},$$

where $|S_j|$ is the cardinality of the set of service phases at intersection *j.* Clearly, when $\eta \to 0$ the distribution of green time among service phases approaches uniform distribution, and every service phase is given equal share of the total available cycle time. Next, we consider the situation $\eta \to \infty$. Let π' be the service phase for which the pressure is the highest, i.e. $\pi' = arg\max_{\pi\in S_j}(w_{\pi}(\hat{Q}(t)))$. This leads to following result:

$$\begin{aligned}\lim_{\eta\to\infty} P_{\pi'}^{j} &= \lim_{\eta\to\infty} \frac{e^{\eta w_{\pi'}(\hat{Q}(t))}}{\sum_{\pi\in S_j} e^{\eta w_{\pi}(\hat{Q}(t))}} \\ &= \lim_{\eta\to\infty} \frac{1}{e^{\eta(w_{\pi_1}(\hat{Q}(t) - w_{\pi'}(\hat{Q}(t)))} + \cdots + 1 + \cdots + e^{\eta(w_{\pi_{|S_j|}}(\hat{Q}(t) - w_{\pi'}(\hat{Q}(t)))}} = 1.\end{aligned}$$

Consequently, as $\eta \to \infty$, the proportion of cycle time allocated to the service phase with maximum pressure approaches 1. These results mean, that the parameter η defines the strength of the preference of higher pressures. The closer the value of η is to 0, the less the pressure difference dictates the proportions, and the distribution of cycle times approach uniform distribution. In the opposite situation, when values of η are sufficiently large, only the phase with largest pressure receives cycle time. As a side note, we notice that as $\eta \to \infty$ our policy approaches the original a-cyclic backpressure algorithm, which picks always the phase associated with the highest pressure.

Because the yellow and red phases in the traffic control cycle are used to improve the safety and flow of traffic, our control algorithm does not affect their durations. Consequently, some of the cycle time is lost and the available green time is less than the cycle length. Let C be the total cycle time, and G_Ω^j be the amount of green time allocated to phase Ω at intersection *j.* Because we want to allocate only the available green time, we have

$$G_\Omega^j = P_\Omega^j * (C - L),$$

where *L* is the "lost time" representing the sum of the duration of all yellow and red phases. Because

$$\sum_{\Omega \in S_j} P_\Omega^j = 1,$$

we only allocate $(C - L)$ amount of time to the green phases. The interpretation of the above result is that we allocate only the proportion of the cycle time that is not needed for the fixed "overhead" phases. In addition, we note that the green time allocation proposed in Equation 4 guarantees a useful property:

$$P_\Omega^j > 0, \ \ \forall\, j, \Omega.$$

4.4.1 Queue length estimation

The input that our traffic control algorithm needs is the accurate estimate of the queue lengths. However, this information is not directly available to us. In this section, we propose a way to estimate this by using only stop line detectors. Our estimate is based on the measured platoon lengths during earlier service phases. In addition, we propose using a forecasting method, such as, *exponential smoothing* or *moving average* to reduce the probability of oscillating control decisions and queue length estimates. Moreover, such a method also enables our system to capture longer term variation in the traffic flow.

The stop line detectors used in Patna can only detect traffic when vehicles cross the stop line. Consequently, any detection can take place only after the control decisions has already been made and a traffic flow has been established the right of way. To solve this problem, we propose using historical data from prior traffic light cycles to estimate the number of vehicles queueing during the current service phase. Although, the backpressure algorithm requires an unbiased estimator of the current queue length, for practical reasons we propose using an estimator based on historical measurements. Instead of counting the number of vehicles that are currently waiting at the intersection, our system estimates the number of vehicles that passed the intersection during the last green allocation.

One way to estimate the number of vehicles that were served during the last service cycle is to notice that the expected number of vehicles served on lane *k* at intersection *j* during a service phase σ is $\sigma_k G_\sigma^j$, where σ_k is the rate at which vehicles from lane *k* are served during green light and G_σ^j is the amount of green time allocated to the vehicles departing from lane *k.* We call σ_k rate the saturation flow rate referring to the assumption that the outflow from an intersection is a fully saturated flow. The formula for the expected queue length, however, is true only if the queue of the vehicles is not exhausted before the end of the green light. If no vehicles are using the remaining green time the intersection capacity is wasted and the number of vehicles served during the cycle is lower than our control policy expects. We can use the fact that either the full queue is discharged during green light or the fact that not all cars are served during the cycle as an indicator of the traffic conditions.

Detecting the end of a moving queue using stop line detectors is straightforward. During the movement of the queue the vehicles are following each other at the saturation flow rate, and the gaps between vehicles are short. Because we can measure the estimated saturation flow rate, we can also compute the estimated time gap between vehicles. After the initial queue has been served, the gaps between subsequent vehicles become longer. We use the detectors to measure the time gaps between vehicles and deduce the arrival rate of the vehicles. When the intensity of the arriving vehicles drops below the saturation flow rate, or equivalently, the gaps between subsequent vehicles cross a fixed threshold, we say that the initial queue has been served at the in-road. By noting the amount of time taken to serve the queue with $\tilde{T}_l$, we can now modify the above estimator to say that

$$\hat{Q}(t) = \sigma_l \tilde{T}_l \tag{5}$$

4.4.2 Gap searching algorithm

In order to estimate the queue length in equation 5, we propose a *gap searching algorithm* in the platoon length estimation module in the detector agent. After initiation, the algorithm continuously measures the occupancy of the detector loop. If the loop is unoccupied for a period of time, called *threshold gap*, the module concludes that the existing queue has been served. Finding this threshold gap in the platoon relates to the logic of vehicle actuated controllers. This logic extends conveniently to the case of heterogeneous traffic faced in the city of Patna. Even though the vehicles do not follow each other in neat queues, the intensity of the traffic is still defined by the time gap between the subsequent vehicles. The detectors that are used in Patna can measure the occupancy of the detector loop, i.e. the presence of the vehicle. By sampling this measurement, we can form an estimate of the continuous evolution of the gap between vehicles.

Algorithm 1 Gap-searching algorithm

1. $input = \begin{cases} \Delta^d : \ time\ resolution\ at\ detector\ d \\ h_{th}^d : threshold\ gap\ at\ detector\ d \\ g^\sigma : maximum\ green\ time\ of\ phase\ \sigma \\ d : detector \\ \rho(\mathrm{d})_t : the\ utilization\ of\ detector\ d\ at\ time\ t \end{cases}$
2. $clock = 0; gap = 0; \tilde{T} = 0$
3. $While\ clock \leq g^\sigma$
    a. $if\ (\rho(\mathrm{d})_{clock} = 0)$
        i. $gap = gap + \Delta^d$
    b. $else$
        i. $gap = 0$
    c. $if\left(gap \geq h_{th}^d\right)$
        i. $return\ \tilde{T}$
    d. $clock = clock + \Delta^d; \tilde{T} = \tilde{T} + \Delta^d$
4. $return\ \tilde{T}$

We propose an algorithm for measuring the length of the queue by adapting the algorithm proposed by Nuli and Mathew (2015). Consider dividing a time interval *I* into a finite number of time slots, with resolution Δ. This means that we sample the utilization of the virtual loop every Δ time units, say, every ¼ seconds. If the virtual loop is not utilized during the time of measurement, we say that a gap is detected. If a gap is measured successively often enough, such that the combined duration of subsequent gaps exceeds the threshold gap, we say that we have detected the end of the queue. This means that the continuous flow of vehicles, i.e. a platoon, has ended. This mechanism is presented in Algorithm 1.

4.4.3 Traffic state estimation

Using Equation 5 directly as the estimator of the queue length can be problematic. First problem has to do with the oscillation of the estimator. Without using any predictive model to predict the future queue length, we risk sequentially over and under estimating the actual value of the queue length. Second problem concerns adaptation to the changing traffic conditions. A predictive model can help capture long-term variations in the traffic by smoothing out random noise in the measurements. As a fix to these problems, we propose using a smoothing method to form more stable estimates of the queue length at time *t.*

Two of the most commonly used smoothing techniques are the exponential smoothing and the moving average. In order to construct an estimate, the exponential smoothing algorithm takes the most recent observation and weights it with a parameter α and adds the earlier estimate with weight $1-\alpha$. The parameter α defines the smoothing factor. The lower the value of the parameter, the more the estimator depends on the historical measurements. The effect of a historical measurement decreases exponentially in its age. Using exponential smoothing with parameter α, the current queue length estimate is given by:

$$\hat{Q}_l(t) = \alpha\sigma_l\tilde{T}_l + (1-\alpha)\hat{Q}_l(t-1), \tag{6}$$

where $\hat{Q}_l(t)$ is the estimate for the number of vehicles in the queue on lane *l*, $\tilde{T}_l$ is the current measurement of the queue service time, and $\hat{Q}_l(t-1)$ is the queue length estimate during last service cycle.

Another common smoothing method is the moving average. The parameter of the algorithm defines the “window” that is used to average the measurements. To compute the current estimator, we cycle through the measurements that belong to the window projected from most current measurement towards the historical measurement, and compute their average. When using moving average with window size *n*, the queue length estimate is given by:

$$\hat{Q}_l(t) = \frac{1}{n}\sum_{i=0}^{n}\sigma_l\tilde{T}_l(t-i) = \frac{\sigma_l}{n}\sum_{i=0}^{n}\tilde{T}_l(t-i) \tag{7}$$

4.5 Conclusion

In this chapter, we proposed a traffic control method that suits the requirements and restrictions faced in the city of Patna. Our approach generalizes a control method proposed in an earlier paper by Le et al. (2015). This method assigns the effective green time of a traffic light cycle to the phases based on difference in queue lengths at the intersection and at the downstream intersections. We apply the method by generalizing it to more a general network control situation. In addition, we overcome the limitations imposed by the detector infrastructure and the heterogeneity of the traffic by proposing an adaptation that allows us to construct an estimate of the current traffic situation. Instead of using real-time knowledge of exact queue lengths, an infeasible and probably unattainable measure, our algorithm uses traffic measurements from prior traffic light cycles to estimate the current queue length.

5 Performance measurements

In this chapter we consider the performance of the algorithm we proposed in chapter 4. Because we are not able to install the system in the actual traffic lights in Patna, we evaluate the performance using micro simulation models. In Section 5.1, we start by arguing the usefulness and applicability of such approach. After this in Sections 5.2 and 5.3, we propose two network that we use to evaluate the algorithm and run simulations on them. We optimize the control parameter values and then compare the performance of the algorithm to a simple fixed cycle controller. In Section 5.3, we run experiments with the simulation model to determine the optimal control parameters used in by the algorithm. After this, in Section 5.4, we evaluate the sensitivity of the results to the modeling choices we made. In Section 5.5, we speculate with the availability of perfect queue length information. We analyze the "value" of this information to consider the usefulness of the backpressure approach and to consider the usefulness of possible sensor infrastructure related investments. This chapter answers the fourth and last research question: *How does the proposed approach perform?*

We place special emphasis on the statistical strength of the conclusions of our simulation results. In the current traffic control literature, simulations are often referred to in an ad-hoc and simplistic manner. The comparison between the performances of the simulated systems often considers any statistical model. The relative performance is usually reported graphically or in a simplistic cross tabulation. Our approach to improve the statistical strength of the conclusions consists of multiple steps. Firstly, we run multiple replications of each simulation and demand setting. This means that we generate multiple demand cases using different random numbers. These demand cases exhibit the same long-term demand pattern but the arrival times of individual vehicles differ from each other. By considering the average performance over all these replications we eliminate possible bias caused by the specific arrival pattern of vehicles that could occur within one replication.

Another possible bias in the estimation of the average performance could be caused by the so-called "warmup" period. Because the system starts empty and it takes time for the system to reach a stable state, the performance measurement during the beginning of the simulation does not represent the actual value of the long-term average performance. We reduce the impact of this bias by estimating the length of this unstable period and discard the performance measurements made during the period.

The last action in improving the statistical strength of our conclusions is the use of formal statistical comparison methods. We run simulations using both our control algorithm and a simple fixed control approach. After this we use the well-known paired-t test to establish confidence intervals on the difference in the performance values. This will allow us to make stronger arguments than by simply comparing simple graphs.

5.1 Microsimulations

Microsimulations are computer software that simulate the dynamics of traffic in a traffic network. These models aim to provide realistic simulations of real-world traffic by simulating the behavior of each individual vehicle in the system. Time is split in discrete steps, say 1 second, and the behavior of each vehicle is simulated for each time slot. The *vehicle following model* describes the way each vehicle follows the vehicle in front of it. Thus, for each simulation step, each simulated vehicle perceives the distance to the vehicle in front and decides appropriate change in its speed. In addition to this, the vehicles follow a predetermined routing through the network. When necessary they change lanes according to a *lane changing model.* Microsimulations allow also the simulation of traffic lights by either forcing the simulated vehicles to stop or providing them the right of way. Consequently, we can implement our control algorithm in a microsimulation package and use it to simulate the performance of our control method. Because of the high degree of accuracy and realism provided by a state of the art microsimulation software, we can use these simulations as an indication of the performance that could be expected from the control method when implemented in a real-world traffic network.

All available microsimulation tools assume that the traffic is homogeneous. Only one vehicle occupies a stretch of a lane at a time, vehicles follow each other in neat queues, and the drivers respect the lane markings. Although in general the traffic in the city of Patna does not fulfill this requirement, we can still use microsimulations as a tool of estimating the performance of our control method. As we concluded in chapter 2, the underling traffic process fulfills the requirements set by microsimulation tools. After a queue of vehicles is moving, the drivers follow each other in queues. In addition, the way we defined the functionality of our detector loop and the estimator of the current queue length relaxes the assumption of constant headway between vehicles. We measure the time a detector loop is occupied, and relate this to the number of vehicles in the queue through the saturation flow rate, which can be measured to take into account the heterogeneous composition of the traffic.

Simulation of Urban Mobility, commonly and later in this thesis referred to as SUMO, is an open-source microsimulation tool developed by Institute of Transportation Systems in Germany. The car following model in it is comparable with the more sophisticated commercial simulators and it is widely used in literature. In addition to the basic simulation functionality, SUMO also provides an API called TraCI that can be used to interact with the simulation during the simulation run. We use this toolkit to implement our traffic control algorithm and conduct experiments to simulate its performance. This simulation architecture is presented in Figure 5.1. Our control algorithm uses the API to fetch information about the state of the virtual loops, computes the control decisions using the control algorithm we proposed in Chapter 4, and sends the control input to the simulation software through the communication pipeline provided by TraCI.

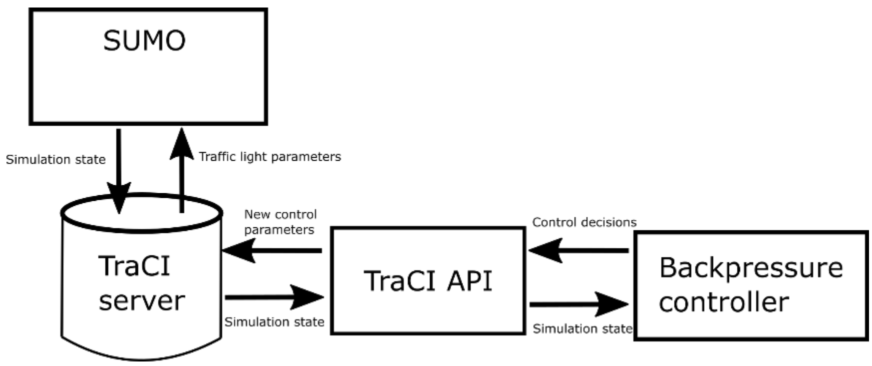


Figure 5.1 Simulation architecture

As we concluded earlier, the main optimization goal of our control algorithm is to reduce the average waiting time in the system. Obviously, we want to use the simulations to see how using our control algorithm affects the average waiting time in the system. SUMO offers some readily implemented KPIs, or key performance indicators. For our purpose, the best indicator is the *delay* per vehicle. This value is reported for each simulated vehicle and it represents the amount of time that the individual vehicle lost due to the traffic lights and other traffic while traversing its route through the network. Although we concluded in Chapter 2 that the majority of the delay in the system probably consists of waiting times at the intersections, this measurement allows us to estimate the effect of our control algorithm more comprehensively, taking into account also possible effects of regulating the degree of saturation of some segments of road.

5.2 Linear network

We use an arterial network shown in Figure 5.2 as the topology for our first performance test. Three intersections 2, 3, and 4 form an arterial route that is controlled by our control algorithm. The remaining intersections 1 and 5-11 are the intersections where vehicles enter the network. This structure of resembles the arterial route found on Bailey Road in the city center of Patna. Because these peripheral intersections do not have downstream intersection as defined by our control algorithm for all of their traffic movements, they are controlled by a simpler version of our algorithm. These intersections conduct queue length measurements in a similar way to the other intersections but use only local information to make the control decisions. Consequently, whereas intersections 2, 3, and 4 use Equation 3 to compute the pressure associated to a service phase, the peripheral intersections use pressure described in Equation 8.

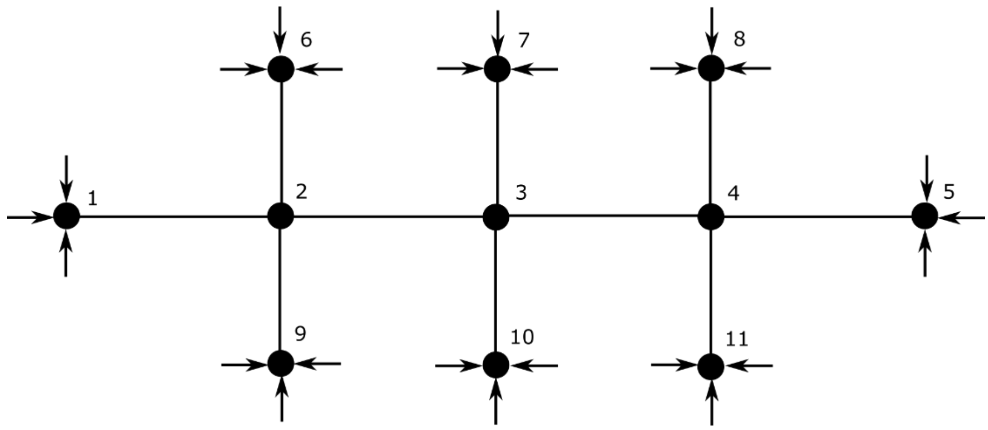


Figure 5.2 Network topology used in the simulations

$$w_{\Omega} = \sum\nolimits_{k \in \mathcal{K}_j} \sigma_k \hat{Q}_k(t) \tag{8}$$

For simplicity, the intersections in the network are all identical. The topology of the intersection is shown in Figure 5.3. We use the traditional 4-arm intersection as described in Figure 3.3. Each in-road consists of 2 lanes. The rightmost lane allows traffic to turn right or proceed straight. The left lane allows only left turning traffic. We arrange the traffic movements in four service phases as in Figure 3.4. In addition, 8 phases are needed to improve safety and the flow of the traffic. After each service phase, the currently active green light is turned into yellow for 3 seconds. This gives the drivers time to react to the change in the right of way. After the yellow phase, there is a short 2 second period showing all lanes red light. This gives the vehicles still in the intersection time to clear the intersection area.

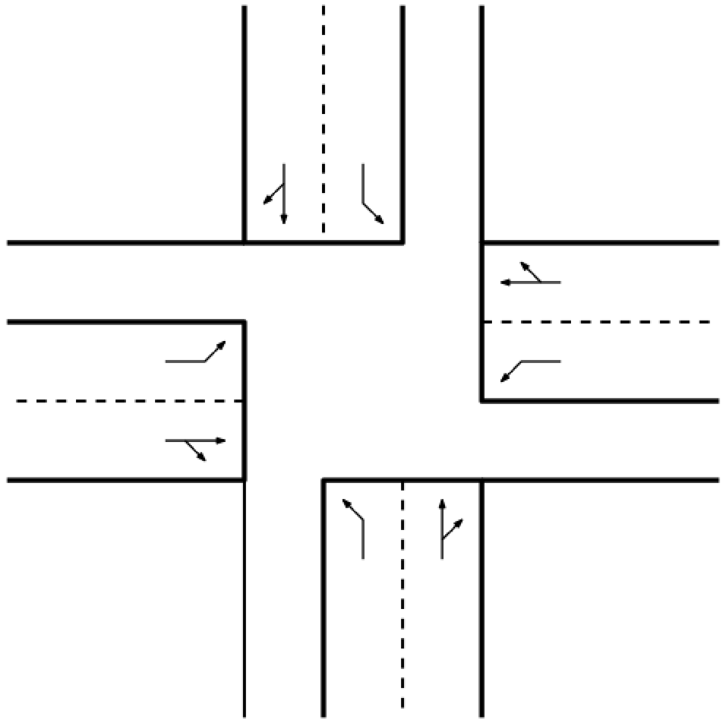


Figure 5.3 Simulation intersection topology

The traffic load, or the number of vehicles in the network, has a major impact in the dynamics of an urban road network. As we mentioned in Chapter 2, as the number of vehicles in the system reaches a critical value the flow of vehicle starts to decrease. This is also a well-known phenomenon from queueing systems. With low values of utilization, a slight increase in the arrival rate translates into a slight increase in the waiting time. But when the utilization approaches 1, i.e. the average load approaches the average service capacity, an increase in average arrival rate leads to an explosive increase in the average waiting time. Consequently, it seems obvious that the type of control that is necessary can depend on the utilization of the network. In addition, the utilization of the urban road network can affect the relative performance of traffic control methods. In order to consider this possible effect, we simulate the network using multiple traffic arrival rates.

In order to compare the performance of our control method, we use fixed cycle traffic light control (FCTL) as a benchmark. This is a common approach in traffic control literature, as fixed control is still common in practical applications and can often provide good control decisions. For the definition of fixed control, we refer to Chapter 3. We use the method described in Webster (1958) to compute the near-optimal fixed cycle length and the associated green split times. This method is described in detail in Appendix A. For simplicity, we use same fixed cycle times at each of the intersection of the network. This is a common method of coordinating the intersections, but we simplify the situation by not using offset values. The average delay encountered in the system when using this method is presented in Figure 5.4.

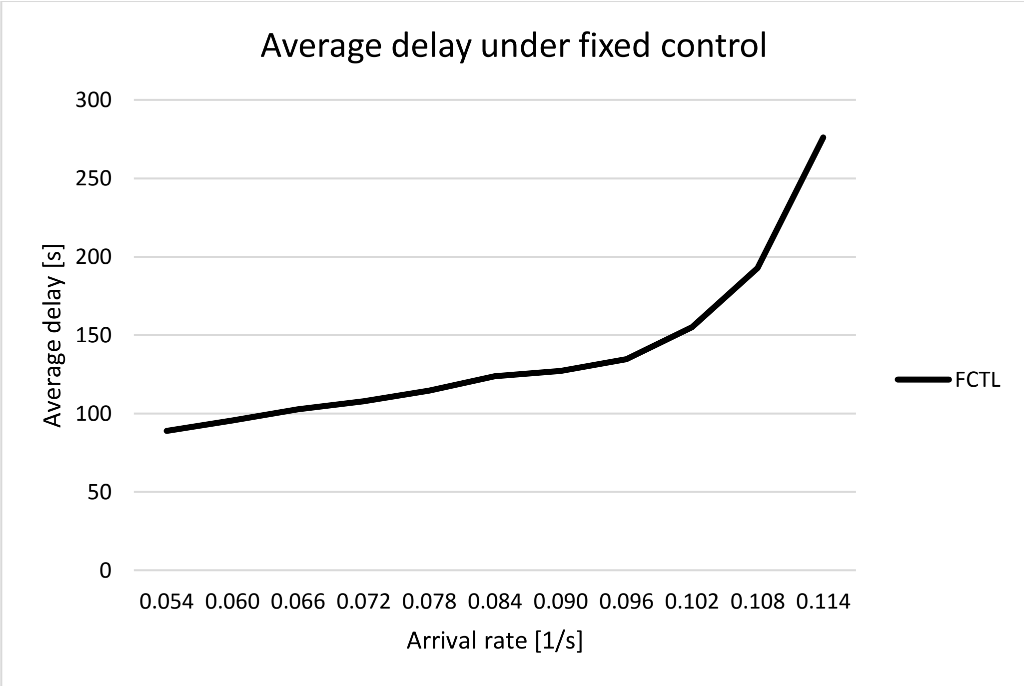


Figure 5.4 Average delay under fixed control

Although the graph representing the average delay does not look like perfectly convex, it is strictly non-decreasing and, given its shape, represents the phenomenon described above. We split the arrival rates into "regimes", with arrival rates 0.054 to 0.078 belonging to *light* traffic conditions, 0.084 to 0.096 *medium* traffic, and 0.102 to 0.114 to *heavy* traffic conditions. We demand case $\lambda = 0.09$, the mid-point of medium traffic regime, to be the baseline demand we use in our comparisons. This arrival rate means that the total load, or the total number of vehicles in the system is $24 * 0.09 = 2.16 \frac{veh}{sec} = 7776\ veh/h$. By using the turn fractions, we can compute the average load on each edge of the network. The distribution of the traffic is represented Figure 5.5.

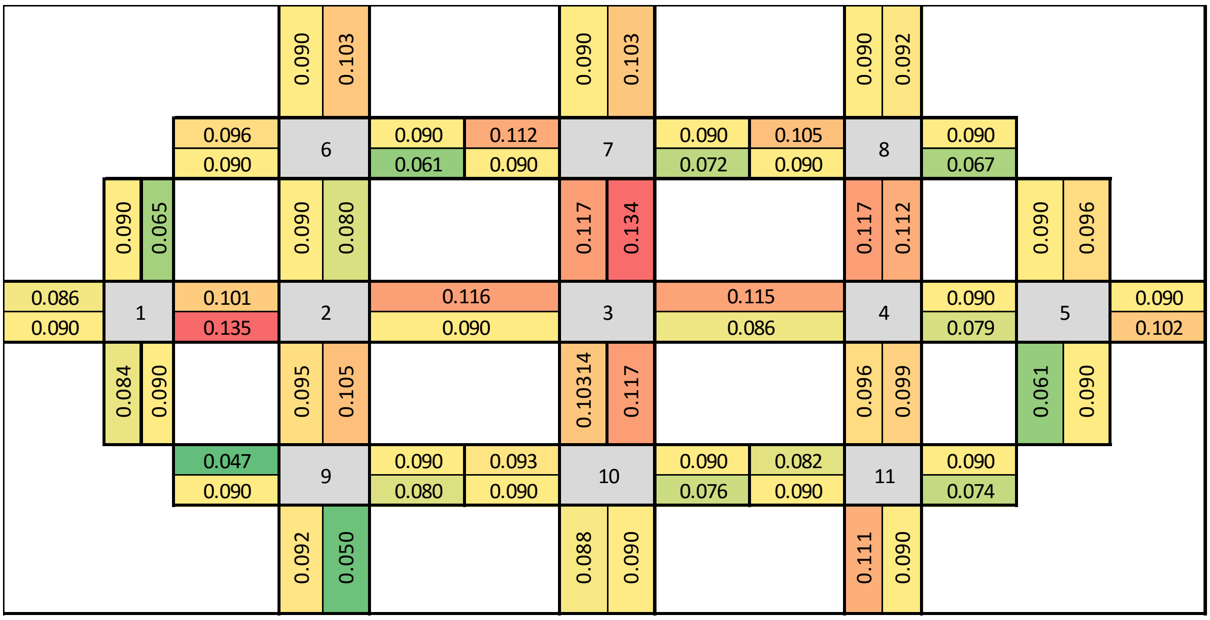


Figure 5.5 Distribution of traffic in the network

5.2.1 Parameter optimization

After defining the structure of the network, the turning fractions, and the traffic demand in the network, we can build a simulation model based on those values. This simulation model can be used to test the performance of our control algorithm. The behavior or the algorithm, however, depends on the choice of the input parameters. Thus, the first step in simulating the performance of the algorithm is to find near-optimal parameter values. In order to conduct the parameter optimization efficiently, it is important to consider the optimal number of simulation runs and the necessary simulation run lengths. Because microsimulations are computationally exhaustive and we have to run a large number of simulations with varying control parameters, finding the smallest amount of simulation runs that yields statistically valid results helps us to reduce the computational effort.

The first step in simulation output analysis is to consider the type of the simulation at hand. Our setting fulfills the definition of a *nonterminating* simulation as described in (Law, 2015). This mean that there is no natural event that would define the end of our simulation. The simulation model generates arriving vehicles continuously using the fixed arrival intensities described above. Consequently, we are interested in the *steady-state* performance of this system. At some point in time, the system reaches a state, in which the delays of vehicles start oscillating around certain fixed value. We call this the steady-state average delay. This is an important value as it gives the expected delay of an arbitrary vehicle arriving in the system and provides a good indication of the performance of a traffic control system. The goal of our research and, consequently, the goal of our control algorithm is to minimize this value.

When interested in steady-state performance in the case of a nonterminating simulation, precautions have to be taken to reduce the impact of the so-called warmup period (Law, 2015). In the beginning of our simulation, there are no vehicles in the road network and arriving vehicles can pass through the network without any delay caused by other vehicles. This, however, does not represent the average delay very well. Usually, vehicles encounter delays caused by queueing at intersections. Queues form because multiple vehicles share delay is caused by queueing at an intersection because of also other vehicles in the system are using the network. Consequently, we have to determine the length of this period and discard the data of vehicles belonging to this period to have more accurate estimate of the actual steady-state average delay.

The control parameter space of our control algorithm is infinitely large. Firstly, we have to choose a value for parameter η from a continuous range $\eta \in \mathbb{R}$. Secondly, the traffic state estimation method involves making a discrete choice between either moving average or exponential smoothing. Based on the choice we have to choose either the moving average window size $w \in \mathbb{N}$ or the exponential smoothing parameter $\alpha \in [0,1]$. The optimal control setting can be any combination of these parameter values. Obviously, our approach is to sample the parameter space and find near-optimal control parameters. To make good sampling feasible, we establish a coarse range of parameter values that contain the optimal combination of control parameters.

We start by experimenting with the magnitude of the parameter η. It defines the strength of the preference of phases with higher weights and is the main parameter of our control method. As shown in Chapter 4, when $\eta = 0$ each service phase receives an equal share of the available cycle time, and when $\eta = \infty$ the phase with highest associated pressure receives the entirety of the effective cycle time. It is obvious that neither of these extreme values is desirable, and there must exist a value $\eta^* \in [0, \infty[$ that is able to capture the dynamics of the vehicle arrivals and adapt to the traffic in a way that minimizes the average waiting time in the system. For this experiment we fix the traffic state estimation method, and thus that the magnitude of η alone explains any differences in the simulation output. As the smoothing method, we use exponential smoothing with parameter $\alpha = 0.2$.

In order to reduce the variance of our performance measurement, we run multiple replications of each simulation settings. We generate a list of arriving vehicles and their arrival times prior to the actual simulation runs. To have more realistic estimate of the long-term average performance of our control algorithm, we generate 13 distinct vehicle arrival patterns. Each use same arrival rates and turn probabilities, but assign different seed values to the pseudo random generator used to emulate the random Poisson arrival. Thus, the arrival times of vehicles represent same long-term demand pattern but are different instance of it. By simulating the control algorithm using these demands and computing the average performance we get better understanding of the average performance of the algorithm. Using only one or few instances of the actual demand runs the risk that some characteristic particular to the random arrivals we generated makes the average performance estimate biased.

With the vehicle demand generated, we run the simulation model. Initially, we set the simulation time "long enough", i.e. 3600 seconds of simulated traffic. Later on, we compute the shortest possible simulation time that is required to provide statistically consistent performance estimates. We start by determining the length of the warmup period using the Welch's method described in Law (2015). Essentially, moving average with varying window length is used to smooth any high frequency variation from the data in order to discover long-term variation in the average performance. This method is graphically presented in Figure 5.6

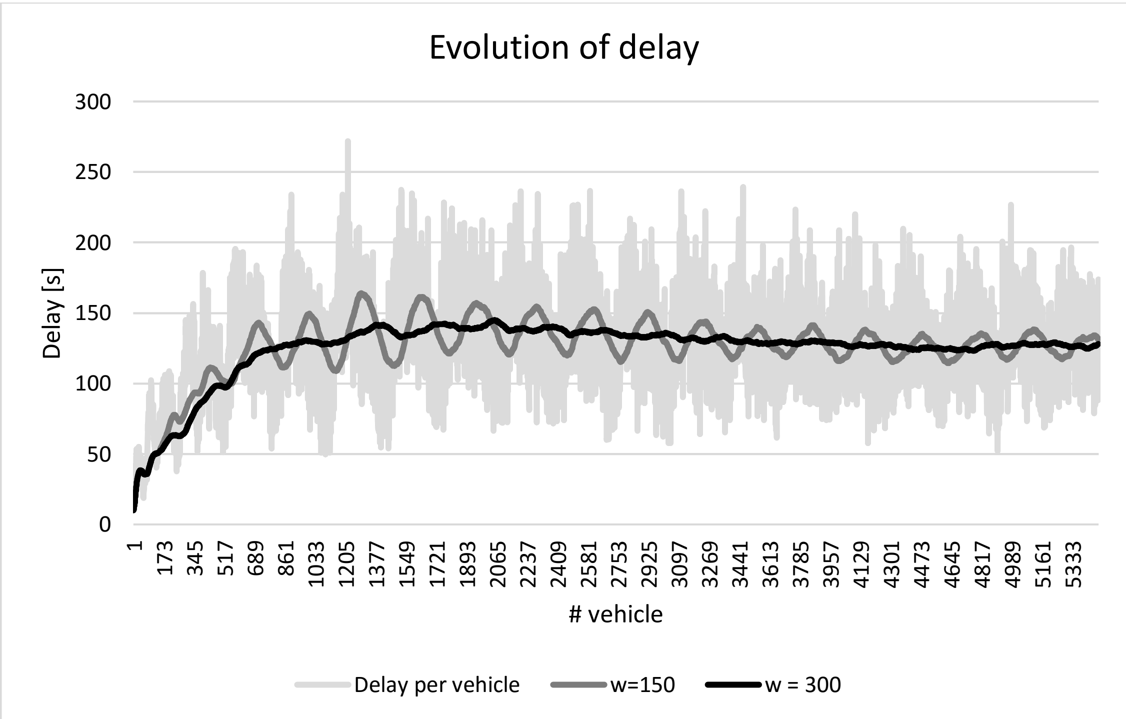


Figure 5.6 Warmup period detection

Initial observation of the data is that the delay that each vehicle encounters varies considerably. This is to be expected as vehicles have varying routes through the network. Some vehicles traverse only two road segments and visit only one intersection. At the same time, with positive probability, some vehicles visit 5 traffic light controlled intersections and traverse 6 road segments. This also signifies the reason that we are interested in the average delay in the network. The graphs showing the moving average values for window sizes 150 and 300 are clearly smoother and a long term change in the average delay is clearly visible. We also see that after the initial rise and subsequent decrease in the average delay value, the long-term average delay stabilizes after approximately 3400 vehicles. We say that these 3400 first vehicles are part of the warmup period and that the delay they encounter does not represent the actual performance of the system. Consequently, in all our analysis, the measurements concerning first 3400 vehicles are discarded.

Next, we begin searching for the best combination of the parameter values. The first step is to find ranges of parameter values that probably contain the optimal combination of parameters. Finding these tighter bounds for the parameter values enables us to use formal optimization methods more effectively. First, we consider both of the parameter η and the respective smoothing parameter in isolation. After this, we use an optimization method that allows varying the value of the parameters simultaneously. We start by considering the effect of the parameter η. We fix the smoothing method, exponential smoothing with parameter $\alpha = 0.2$ and let the value of η vary. By discarding the data gathered during the warmup period we can reliably estimate the effect that the parameter η has on the average delay. This relation is shown in Figure 5.7.

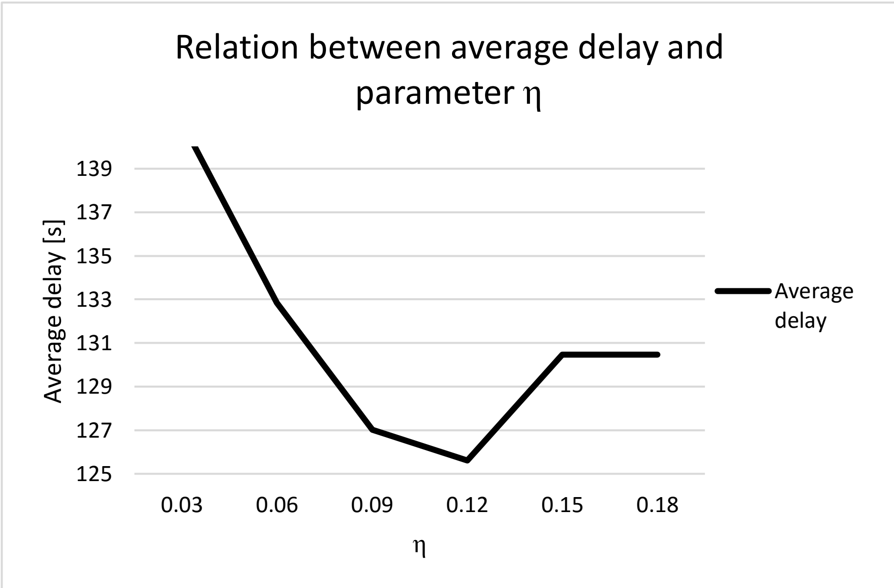


Figure 5.7 Average delay as a function of η

We see the average waiting time attains its minimum value when $\eta \approx 0.12$. Consequently, we assume that the optimal value of η lies somewhere in the range [0.09,0.15]. This narrows down the possible range of values, and makes sampling the search space more effective. Next question concerns the choice of the smoothing method and its respective parameter. We have two choices: exponential smoothing with parameter $\alpha \in [0,1]$ and moving average with window size $n \in \mathbb{N}$. One way to consider this design choice is to fix $\eta = 0.12$ and run simulations with both estimation methods. Because we use same traffic demand cases, we can assume that any differences in the average performance are caused by the adaptive and predictive capabilities of the models. For exponential smoothing, we use values sampled from the range $[0.05, 0.35]$. The effect of the window size used by the moving average method is simulated with values $w \in \{2,3,4,5,6,7\}$. The comparison of the performances of the smoothing methods is presented graphically in Figure 5.8.

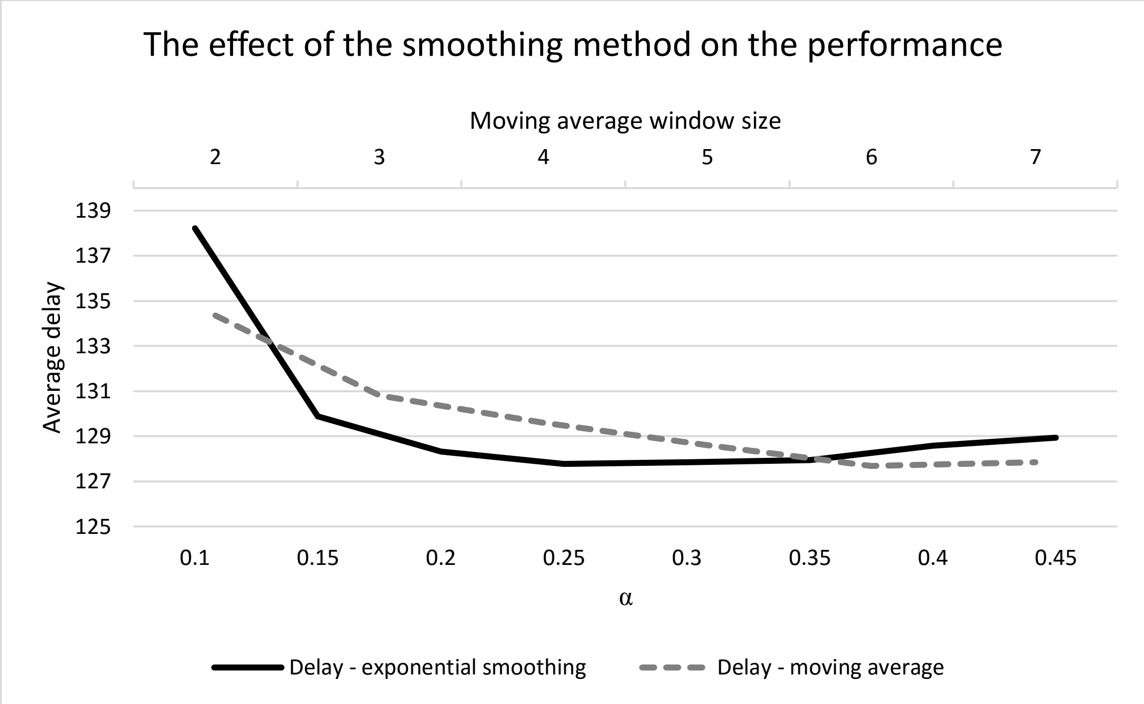


Figure 5.8 Average delay as a function of the smoothing method

The results indicate that by choosing the value of smoothing parameter optimally the smoothing methods can be made perform equally. The exponential smoothing attains its optimal performance with some value $0.15 \leq \alpha \leq 0.35$. This means that in the current network and with the current traffic demand a high amount of smoothing is desirable. The exponential smoothing method attains its best performance with parameter value $\alpha \approx 0.25$, which means that 75 % of the value of the queue length estimator is based on the earlier value of the estimator. It seems the that in this traffic case it is beneficial for the system to produce very stable queue length estimates. This result is to be expected because the arrivals to the system are identical stable Poisson processes. The results concerning the optimal window size of the moving average method are consistent with this notion. We see that the performance is optimized with window size 6. This means that last 6 queue measurements are taken into account with a 0.167 each. This is roughly in the same magnitude as the smoothing parameter of the exponential smoothing method. We see that the performance is improved by making the window longer, i.e. the more smoothing is used, but the performance plateaus after window size 6.

For simplicity, later in this thesis we decided to consider only the exponential smoothing as a design choice. We see that under optimal parameter value choice, there is no significant difference in the performances of the methods. Given that the methods perform equally well, the exponential smoothing is easier to implement and faster to compute only requiring the storage of one value at a time. The moving average requires storing historical queue lengths for the length of the window size. In addition, the warmup period is probably slightly longer for the moving average method as during the first 6 simulated traffic cycles the estimation method is not working as intended. Moreover, for the purpose of parameter optimization in the later parts of this chapter, it is beneficial to have parameter that attains continuous values.

We have now established coarse ranges that probably contain the optimal parameter values. Next step is to run experiments to find the actual best combination of parameter values. The earlier steps were used to reduce the size of the search space that we need to explore. Although the search space consisting of two continuous variables is still infinitely large, we can sample the search space more efficiently. Next step is to optimize the values of the control parameters. One common way is the use of metamodels and hill climbing algorithms. We do this by fitting a regression model to design points and the simulation response. After this we use the gradient descent method on the regression function to find new parameter settings that will probably decrease the average delay.

In order to fit a function to data, we have to run multiple simulations and for these simulations, we have to choose parameter values from the range we defined above. *Latin hypercube design* is a method that can be used to choose the design points of our experiment. In our case, we have to choose values for two control parameters from continuous closed interval. Essentially, the method discretizes the continuous range of both control parameter values into *m* equally spaced values. After this both parameters are permuted independently to receive one of the *m* possible values (Law, 2015). We use m = 20 and generate 20 design points. These design points are projected onto the (η, α) plane in Figure 5.9.

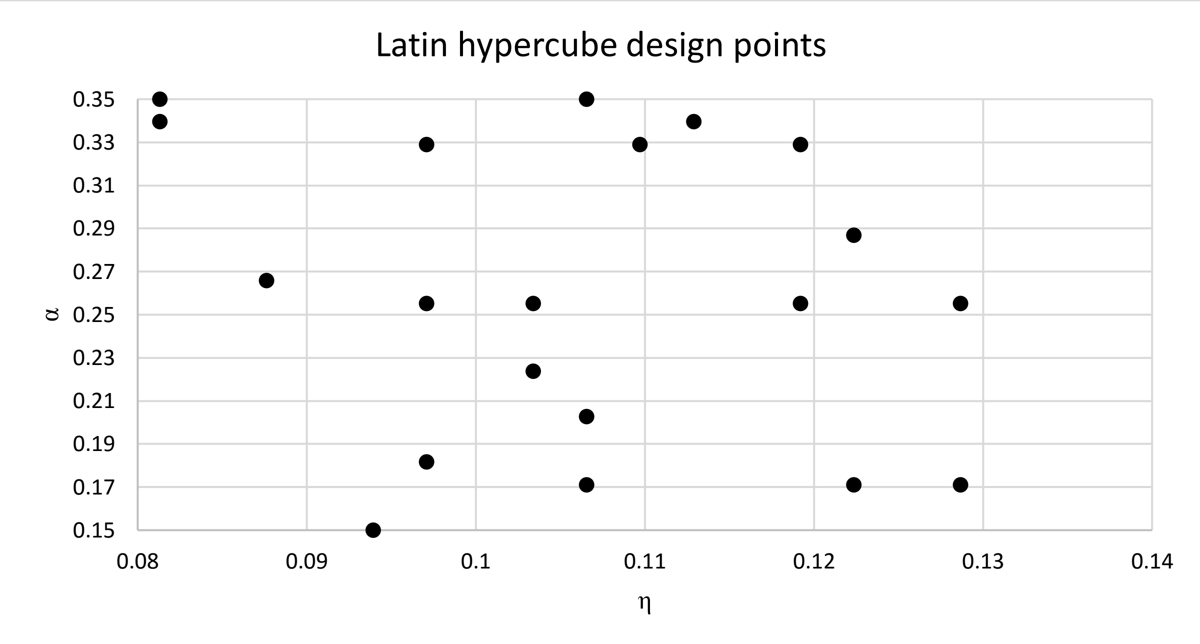


Figure 5.9 Projection of design points onto the (α, η) plane

We make 13 replications for each design point and calculate the average performance related to each combination of parameter values. After this, we conduct a linear regression analysis to fit a regression model to the data. $\hat{R}(\eta, \alpha)$ represents the average delay in the system as a function of control parameter values η and α. The stepwise regression analysis results in the following coefficients:

$$\hat{R}(\eta, \alpha) = 137.32 - 74.25\eta - 8.49\alpha \tag{9}$$

In general, the regression model does not fit the data well. R^2 value measuring the models ability to explain the variance in the dependent variable is 0.548 indicating a poor fit. However, the coefficients are statistically significant with risk level 0.05. The relation between the performance of our algorithm and the parameters is probably more complex than the linear regression model is able to estimate. As this regression model is the best estimation of how the parameter values affect the performance, we utilize it in improving the parameter settings. Notice that $\hat{R}$ represents the average delay, which we want to minimize, as a function of η and α. It is common knowledge that values of a function decrease the fastest when moving towards its negative gradient. The steepest descent method considers the currently best known combination of parameters, computes the negative gradient in this point and looks for a combination of values along the direction of this vector that reduce the average response of the simulation model. This is carried out iteratively until no decrease in the response is found. Because our regression model has only first degree linear terms, finding the gradient is straightforward

$$\nabla\hat{R}(\eta,\alpha) = -74.24\boldsymbol{e_\eta} - 8.49\boldsymbol{e_\alpha} \tag{10}$$

To reduce the average response of the simulation model, we want to move towards the opposite direction of this gradient vector, i.e. we choose the next parameter values by moving along the vector (74.24, 8.49). We consider the best parameter combination found using the Latin hypercube design, and generate new test points by sampling the search space along this vector. This, however, does not lead to improvement in the average delay. In order to test if using non-linear terms increases the fit and the predictive power of the regression model, we also fit models

$$\hat{R}(\eta,\alpha) = 116.86 + \frac{0.818}{\eta} + \frac{0.809}{\alpha} \tag{11}$$

and

$$\hat{R}(\eta,\alpha) = 145.99 - 48.406\eta - 98.601\alpha - 3.255\eta^2 + 188.159\alpha^2 - 54.435\eta\alpha \tag{12}$$

The R^2 values for these models are 0.666 and 0.758, respectively. Consequently, the models fit the data better than the original linear model. For the model described in Equation 12, this is probably due to the higher number of explanatory variables. Although not as straightforward as for the linear model, the gradient vectors for these models are easily found. We sample new design points along the negative gradient vectors, but none of the new trial points reduce the average delay. As none of the gradient descent trials result in improvement of the average delay in the system, we pick the best performing combination of parameter values from the original design points shown in Figure 5.9. Thus, we conclude that the near-optimal control parameters for the baseline demand case are:

$$\begin{cases} \eta = 0.1224 \\ \alpha = 0.2869 \end{cases}$$

5.2.2 Performance comparison

After finding good parameter values, it is interesting to see how our control algorithm performs against the fixed cycle traffic light control. This is important in deciding whether or not our control approach can be expected to perform better than other control approaches. It is also interesting to see, how varying the traffic load in the systems affects the relative performance of the different control methods. We simulate our traffic control algorithm with the parameters found above using the traffic demands described in Figure 5.4. Again, we run 13 replications of each demand level, detect the warmup period and remove the data that does not represent the long term average delay. Figure 5.10 presents the comparison of the performance of the fixed controller and our backpressure controller.

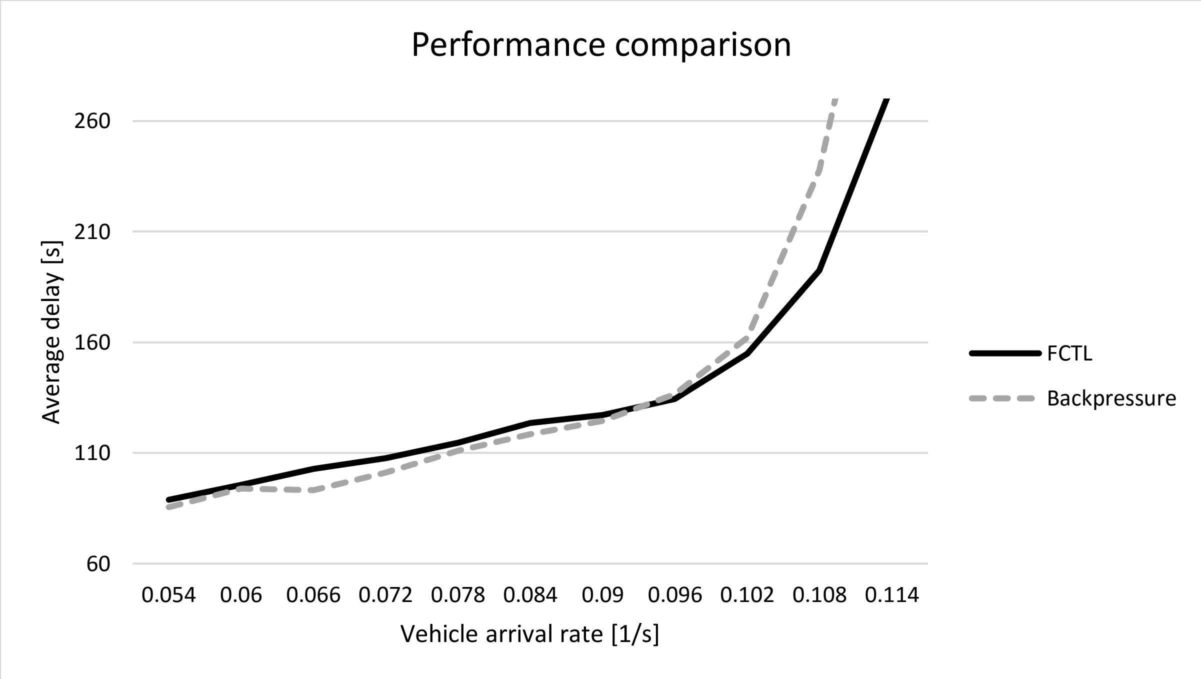


Figure 5.10 Comparison of the performance of control methods

When comparing the performance of the fixed cycle controller and our cyclic backpressure controller, we see that in the demand case $\lambda = 0.09$ that we used to optimize the parameters our control approach performs better. Interestingly, it seems that with lower vehicle arrival rates the control algorithm with the parameters found above performs better than the fixed control. When the demand is increased, however, the average delay deteriorates faster than it does with fixed control and the fixed control becomes the better choice. We conclude that using our control approach with appropriate control parameter values can lead to reduction of the average delay in the system. In addition, although both control methods share the cycle times for each of the arrival rates, the green splits for the fixed controller are separately optimized for each of the demand cases, whereas our control algorithm is able to find reasonably good green time allocations using parameters optimized using only one demand case. Consequently, we investigate next if the optimal values of the parameters η and α depend on the traffic load in the system, and the possible improvement in the performance gained by choosing the parameters of our algorithm as function of the traffic load. This is also suggested by the fact that the graph depicting the evolution of average delay as a function of the vehicle arrival rate is not convex. This could indicate that the parameters used yield sub-optimal control decisions for some of the traffic loads.

One way to link the parameter values to the prevailing traffic conditions is to consider the "regimes" of traffic that we described in Figure 5.4. We optimize the parameters separately for light, medium and heavy traffic loads. Such an approach is commonly used in practical control systems. For example, a clock or real-time measurements are used to determine whether the current traffic belongs to peak-hours or off-peak-hours and corresponding (pre-determined) control plans are implemented. In our first attempt to improve the reduce the average delay, we consider the mid-points of the traffic load regimes to be representative of the traffic conditions and, thus, indicate the magnitude of the optimal parameter values. Consequently, we consider traffic loads $\lambda_{Light} = 0.066$, $\lambda_{Medium} = 0.09$ and $\lambda_{Heavy} = 0.108$. We run the parameter optimization process described above for these demand cases. Obviously, the parameters found using the baseline demand case are the near-optimal parameters for the medium demand case. The near optimal parameter values for the three demand cases are:

$$light: \begin{cases} \eta = 0.261 \\ \alpha = 0.088 \end{cases}$$

$$medium: \begin{cases} \eta = 0.244 \\ \alpha = 0.150 \end{cases}$$

$$heavy: \begin{cases} \eta = 0.138 \\ \alpha = 0.269 \end{cases}$$

We see that the higher the traffic demand is the lower the optimal value of control parameter η is. This is caused by the fact that in heavy traffic most phases are fully utilized and the traffic flow exhibits fewer short term changes that the controllers can utilize to reduce the average delay in the system. In such situation, it seems to be beneficial to construct stable green light schedules with only minor adjustments. This is done by decreasing the preference of phases with higher pressures. The situation is the opposite in the case of light traffic. Now it is beneficial to have higher value of η. This leads to the system's ability to quickly react to short term variations in the traffic flows. Higher value of η is necessary as the difference in queue lengths may be only one vehicle, but network optimal decision might require serving the longer queue considerably longer time.

The way the optimal value of α behaves seems somewhat illogical. We would expect higher level of smoothing be beneficial in the heavy demand cases where the value of η is low. The optimal value of α, however, exhibits the opposite behavior. The smallest value of α, i.e. the highest amount of smoothing, is used in the light traffic case together with the highest value of η. Probably, the optimal control arises from the interaction of the values of these parameters. Consider, for example, the light traffic case. The high value of η already provides higher amount of adaptiveness to small changes in the traffic. Then, it seems, it is beneficial to use higher amount of smoothing to keep the system from oscillating too much. The opposite is true for the simulations concerning the heavy traffic. η is given a lower value leading to less difference in the green time allocation, but α is considerably higher allowing the system to still capture short term variations in the queue lengths.

Using the midpoint of the traffic load regime clearly indicates that different parameter configurations are necessary to respond to the differing control requirements of the demand regimes. Because the near-optima parameter values exhibit a consistent behavior, with η decreasing and α increasing with increasing traffic load, we consider interpolating and extrapolating good parameter values for the demand cases that were not considered in the optimization process. In addition to the probable increase in the performance of system, such approach also provides smoothness in the change of the parameter values. Relating the parameters of the control algorithm to the actual traffic load that can be measured by the detectors, instead of arbitrarily defined regimes, improves the stability of the controller and the control decisions made by our control system. For demands that are between the demands that were used to optimize the parameters we use linear interpolation to find values for η and α. The parameter values for the remaining demand levels are extrapolated assuming a linear dependence. In general, we expect the parameter values found through interpolation to perform better than extrapolated values. The comparison between fixed control and our control algorithm using both regime midpoint values and linearly interpolated values is presented in Figure 5.11.

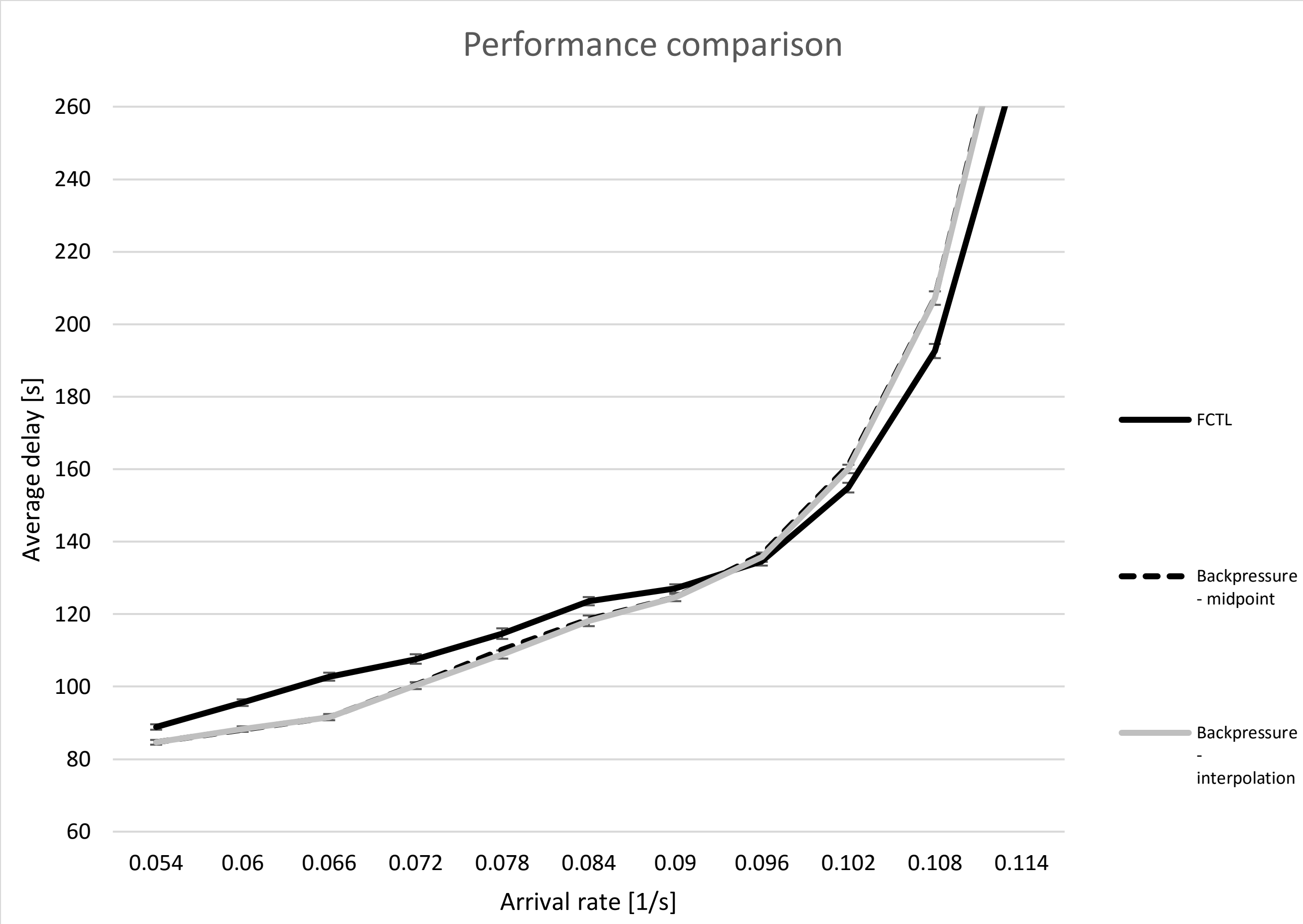


Figure 5.11 Performance comparison using optimized control parameters

We conclude that in average our control method performs better than a fixed cycle controller. Using regime midpoint optimized parameter values perform better than using a single parameter setting optimized for the baseline case, and the interpolation method performs slightly better than the regime midpoint method. The graph in Figure 5.11 shows also the 95 % confidence intervals of the average delay estimates. They show that for arrival rates $\lambda \leq 0.09$ our algorithm performs better than fixed controller and the difference is statistically significant with 95 % confidence. Our control algorithm performs slightly worse than a fixed controller for $\lambda = 0.096$ but this difference is statistically not significant. For demands $\lambda > 0.096$ our control algorithm performs worse than fixed control. Table 5.1 presents more accurate performance comparison between fixed traffic control and our backpressure controller.

One immediate conclusion is that in average our controller performs better than the fixed controller. The magnitude and sign of the performance improvement, however, depends heavily on the traffic load and the way that the control parameters are chosen. By using one set of control parameter values leads to improvement in the average delay for traffic loads that are lighter than the baseline demand. When the traffic load is higher than the value that was used to optimize the control parameters, nevertheless, the opposite is true and the average performance deteriorates significantly. When only one combination of control parameters is used, the average improvement is actually negative, meaning that in general this type of control makes the situation worse. Using the more complicated ways of choosing the control parameter values improve the performance of the control algorithm. Both of these methods now produce in average a positive reduction in the average delay in the system, with the interpolation/extrapolation method yielding slightly better results.

Table 5.1 Comparison of the performances of fixed control and backpressure controller

		FCTL	Backpressure - one value		Backpressure - regime mid		Backpressure - regime interpolation	
	Arrival rate	Average delay [s]	Average delay [s]	Improvement [%]	Average delay [s]	Improvement [%]	Average delay [s]	Improvement [%]
Light	0.054	88.87	85.51	3.77	84.67	4.72	84.65	4.74
	0.06	95.59	93.92	1.75	88.09	7.85	88.29	7.64
	0.066	102.75	93.27	9.23	91.59	10.86	91.59	10.86
	0.072	107.63	101.11	6.05	100.57	6.55	100.29	6.82
	0.078	114.63	110.93	3.23	110.19	3.88	108.88	5.02
Medium	0.084	123.58	118.52	4.09	118.52	4.09	118.17	4.38
	0.09	127.06	124.59	1.94	124.59	1.94	124.59	1.94
	0.096	134.53	136.50	-1.46	136.50	-1.46	135.71	-0.88
Heavy	0.102	154.90	162.18	-4.70	160.82	-3.82	160.07	-3.33
	0.108	192.59	237.84	-23.49	207.24	-7.61	207.20	-7.59
	0.114	276.01	384.51	-39.31	305.38	-10.64	303.87	-10.09
Average				-3.54		1.49		1.77

Table 5.1 presents the results of the Figure 5.11 in more detail. We see that with low traffic volumes our algorithm performs up to 10.9 % better than the fixed controller, and that in average our method is 1.77 % better than the benchmark controller. Under heavy traffic, however, our controller seems to perform worse than a traditional fixed cycle controller. As Figure 5.4 shows, this regime of traffic is characterized with exponential deterioration of the average delay in the system. In addition, the average queue lengths in the system grow unbounded, which indicates that the system is operating above its service capacity. Consequently, the concept of average delay becomes ambiguous as the average does not stabilize around a certain value. Due to the increasing number of vehicles in the system, the queue lengths are longer than the green times allocated to the traffic movement. This causes our control algorithm to underestimate the queue lengths on highly congested lanes leading to possibly detrimental control decisions. Besides, with higher demand the traffic flow does not exhibit considerable short term variations that the controller could utilize in optimizing the delay in the system. It seems that in these situations it is beneficial to implement stable green split schedules rather than try to do short term optimization that can lead to unrecoverable instabilities in the system.

5.2.3 Fairness of the traffic control

Although in our simulations the fixed cycle control and our control algorithm use the same cycle times, for most traffic loads, our algorithm is able to produce shorter average delays in the system. This could be possible due to a few reasons. Firstly, due to adaptiveness of our control approach, the controller can adjust to the actual traffic conditions, while a fixed controller only is configured to fit the average traffic situation. In addition, the adaptiveness can allow the controller to capture short term variations in the traffic flow that can be used to allow vehicles pass the intersection with less delay. Lastly, the backpressure policy is shown to guarantee network stability and, consequently, the control decisions might prefer network-wide benefit of optimizing the flow of high-intensity traffic flow over the less frequently used routes, or our control algorithm is able to find inefficiencies in the fixed timing plans that provide unfair advantages to some traffic flows leading to higher average delay in the system.

We characterize the effects that our control algorithm has on the traffic in the system by considering the way the delay is distributed in the system. First, we analyze the way the average delay is distributed over the different routes in the system. We call this the “equality” or “fairness” of the control. This analysis can point out, whether the system prefers optimizing a few high-volume routes or if it is better for the network-wide performance to guarantee a certain service level for multiple traffic flows. After this, we measure the effect that changing from fixed control to our backpressure control has on the average queue lengths in the system. Again, this analysis is used to consider the way that our control algorithm is able to reduce the average delay in the system.

In order to measure the “fairness” of the traffic control decisions, we propose using the so called *Gini coefficient*. It is a statistical measure that characterizes the dispersion of a probability distribution. It is often used to measure the economic inequality in a country. We extend this interpretation to measure the inequality in waiting times between vehicles choosing arbitrary routes through the network. We compute the average waiting time over the 13 replications for each route used by the vehicles in the network. We then interpret this waiting time as the “wealth” of the route, and compute the dispersion of this wealth among all used routes. The Gini coefficient for an empirical distribution with the wealth sorted in non-decreasing order, i.e. $y_i \leq y_{i+1}$, is given by Equation 13.

$$G = \frac{n+1}{n} - \frac{2\sum_{i=1}^{n}(n+1-i)\,y_i}{n\sum_{i=1}^{n} y_i}, \tag{13}$$

where *n* is the number of routes, y_i is the average delay on the *i*th route. With non-negative wealth values, or delays in our case, the Gini coefficient achieves a value between 0 and 1. Value of 1 represents extreme inequality, i.e., all wealth is owned by only one entity. The perfect equality of sharing the wealth equally among the entities is indicated by a Gini value of 0. In our case, we are interested in the way the delay in the system is distributed among the different routes the vehicles take. As our control algorithm is able to reduce the average delay in the system the distribution of the delay can indicate the way this is achieved. Reduction in the inequality could indicate that the fixed cycle control favors some traffic flows in a way detrimental to the network-wide performance, and our traffic control is able to correct this by providing more fair control decisions. An increase in the Gini value, in turn, could indicate that it is network-optimal to reduce the average delay by minimizing the delay along some high volume routes.

Table 5.2 summarizes some values that can be used to consider the fairness of the way the delay in the system is distributed over the vehicles and routes. In order to provide consistent results, we consider for both routes that pass through exactly three intersections. This means, that they have to visit one of the intersections on the main arterial route. We see that the coefficient of variation remains essentially unchanged. This would indicate that the way delay is allocated to the different routes does not change. A slightly contradicting result can be seen in the value of Gini-coefficient. Although the increase is very small, the value is slightly higher for our backpressure controller. This indicates that difference between routes with short delay and routes with longer delay is slightly bigger, indicating an increase in the inequality of the traffic control.

It seems that the backpressure controller is able to reduce the average delay in the system by finding network-optimal control decisions that lead to slight increase in the inequality in the service of the traffic flows. Reducing the delay along routes with higher traffic load leads to gains in the average delay that are higher than the slight increase in the average delay along routes with fewer vehicles. This is a common phenomenon in backpressure control, and one of the key reasons we decided to implement a cyclic version of the algorithm.

Table 5.2 Comparison of the distribution of delay in the system

Method	Average delay per vehicle	Average delay per route	Var(delay/route)	Coefficient of variation $= \frac{\sigma}{\mu}$	Gini coefficient
Backpressure	91.59	160.94	4226.55	0.404	0.229
FCTL	102.75	177.00	5141.79	0.405	0.221

Another approach to consider the way our control algorithm is able to reduce the average delay is to analyze the distribution of queues in the system. We rerun the simulation experiments used to analyze the fairness of the traffic. This time, we measure the lengths of the queues at the intersection at the moment of the green light initiation for the respective lane. We compute the average over the service cycles and the 13 replications to find the average queue length at each lane in the network. By running this analysis on both the fixed cycle controller and our backpressure controller, we can analyze the difference in the values. As we use exactly the same vehicle arrival processes for both cases any difference in the queue lengths is due to the difference in the performance of the control algorithms. In addition to considering the queue length, this analysis can provide indication of how delay is incurred in the different parts of the network. By assuming that the well-known queueing theory result, Little's Law holds at intersection level we can express the average delay at an intersection as a function of the average queue length.

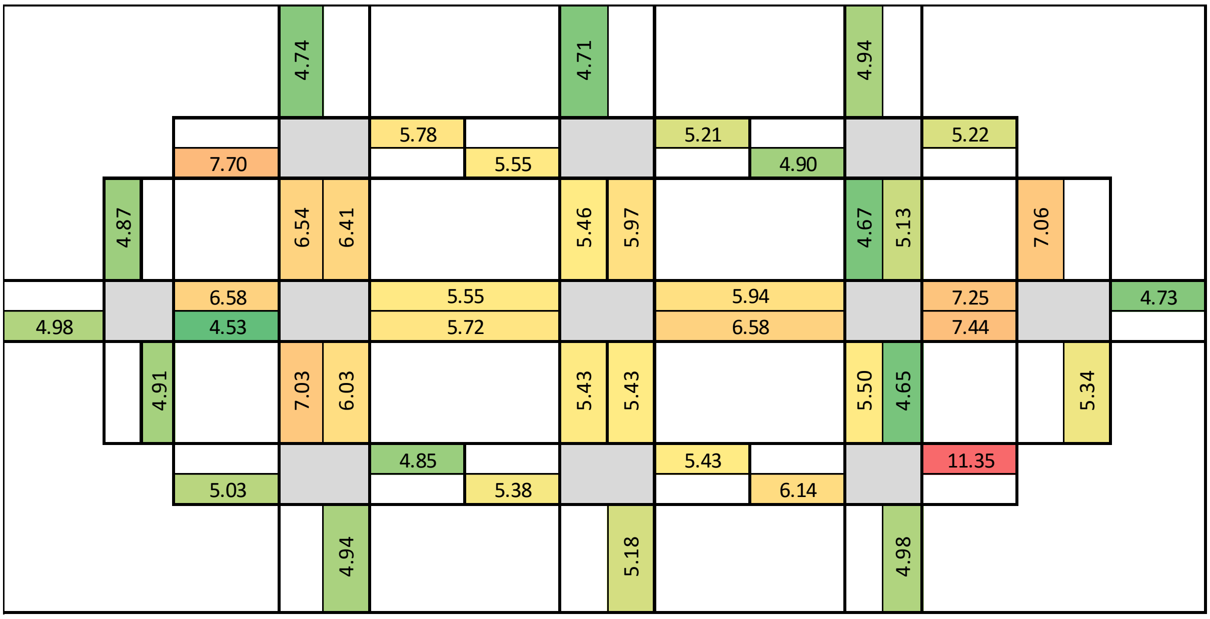


Figure 5.12 Average queue lengths per road under fixed cycle traffic control

Figure 5.12 presents the average queue length at the time of green light initiation on the roads in the system. Figure 5.13, in turn, shows the reduction in the queue lengths at the roads. A positive value indicates an increase in the queue lengths, and a negative value suggest a decrease in the average queue length on the respective road. Clearly, some of the roads have actually longer average queue lengths when controlled by our control algorithm. In total, our algorithm reduces the sum of all queue lengths by roughly 2 %. If we assume that the urban road network we simulate is a general queueing system, this result is to be expected as a consequence of the reduction in the average waiting time in the system.

Because the queue lengths on the roads are not universally decreased and some are increased by up to 20 %, our algorithm must find a way to disperse the queues that are critical in the reduction of the average delay by allowing some less critical queues to build up. Another way to consider the way our control algorithm is able to reduce the average delay is to reflect the reduction in queue lengths with the length of the queue shown in Figure 5.12 and the traffic demand per road segment shown in Figure 5.5. It seems that our algorithm reduces the queue length most, where the queues are already long. Moreover, it seems that the reduction is largest for roads that have long queue lengths but that have relatively low traffic load. This means that the queues are reduced mostly from where the extensive queue length cannot be justified by high traffic load. High traffic load on a road will inevitably lead to formation of queues no matter the control method. It seems, however, that the fixed Webster's method leads to unnecessarily long queue lengths on some roads because the major traffic flows are over prioritized. Because majority of the vehicles driving through the network visit only one of the peripheral intersections, the unnecessarily long queues on the minor traffic directions cause excessive delays deteriorating the network wide performance.

The way the Webster´s method is using the *max* operator to determine the most critical traffic movement in an intersection probably leads to very poor capacity allocation to traffic movements that have relatively low average demand. This then leads to longer queues than expected on some roads. Our control algorithm is able to correct this by finding a network throughput optimal way to allocate these traffic movements more green time.

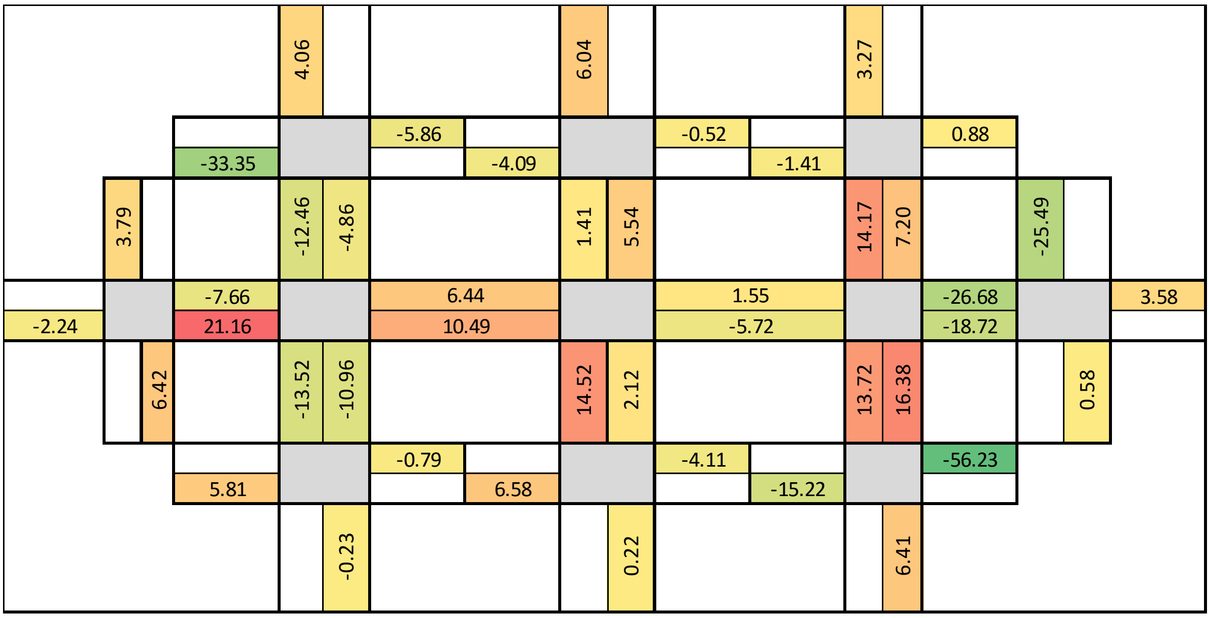


Figure 5.13 % of reduction in the average queue length per road achieved by using the backpressure controller

5.3 Grid network

Above, we considered an arterial network, where the controlled traffic lights form a chain. The network topology was chosen as it is commonly used in traffic control literature and such a topology can be found in multiple locations in the city of Patna. Another interesting network structure to be analyzed by simulating it is the so-called grid network presented in Figure 5.14. Now each intersection in the center of the network is connected to two other backpressure controlled intersections. In addition, each central intersection is connected to two peripheral intersections controlled by the simplified control algorithm proposed in Equation 8. This type of network is often used in the literature and the tightly spaced traffic lights in the city center of Patna form a grid-like network. We expect that the controlling of this network is more difficult than the controlling of the arterial system, because the intersections are more connected to each other.

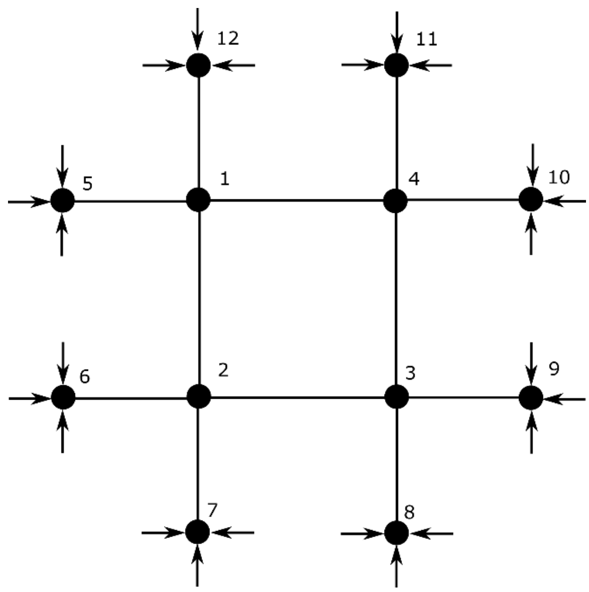


Figure 5.14 Topology of a grid network

Again, we simplify the simulation model by using identical intersection and phase structures as described in Figure 5.3. Because the network is larger than the linear network we simulated above, the base capacity of the network is slightly higher. We simulate various traffic demands by varying the arrival rate at the peripheral intersections between $\lambda_1 = 0.081$ and $\lambda_8 = 0.144$ vehicles per minute. Like in the earlier simulation, we divide the arrival rates to traffic demand regimes that describe coarsely the saturation of the traffic. Demands $0.081 \leq \lambda \leq 0.99$ belong to the light traffic regime, $0.108 \leq \lambda \leq 0.126$ form the regime of medium traffic load, and $\lambda \geq 0.135$ constitute heavy traffic.

We conduct similar parameter optimization as we did for the linear network. First, we establish coarse ranges that likely contain the optimal combination of the parameters. After this, we use the Latin Hypercube design to sample design points from the parameter space. We run simulation replications to determine the performance of the algorithm using the design points as the parameter combinations. Finally, we fit a regression model to the data and use the gradient descent approach to try find better performing parameter values.

Simulating the network using the parameter values optimized for the base demand case $\lambda = 0.177$, the midpoint of the medium traffic regime, results in average deterioration of 7.2 % in the average delay when compared to the fixed cycle controller optimized using the Webster's method. We consider optimizing the parameters for the midpoints of the traffic regimes, a choice that significantly improved the performance of the algorithm for the linear network. We have the following near-optimal control parameters:

$$light: \begin{cases} \eta = 0.07842 \\ \alpha = 0.29894 \end{cases}$$
$$medium: \begin{cases} \eta = 0.06947 \\ \alpha = 0.30158 \end{cases}$$
$$heavy: \begin{cases} \eta = 0.08474 \\ \alpha = 0.36526 \end{cases}$$

Unlike in the case of linear network these parameter values behave somewhat illogically. The value of the smoothing parameter behaves like we expected, having the higher value the more saturated the traffic is. We saw the similar relation with the linear network structure. The optimal value of the parameter η, however, behaves differently. The conclusion of the earlier simulation was that the optimal value should decrease with increasing traffic demand. Our simulations, nonetheless, indicate that when moving from light to medium regime in the current network, the optimal value of η does decrease but when we continue to increase the traffic load in the system the optimal value of the parameter increases again. For heavy traffic loads, the optimal value of the parameter is even higher than it is for the light traffic regime.

The grid network seems to be less sensitive on the parameter values than the linear network. Firstly, we see that the differences in the optimized parameter values are more minute than the differences in the parameter values for the linear network. Secondly, the performance of our control algorithm does not improve significantly by using the above optimized parameter values. In average, our control algorithm still leads to deterioration of the average delay in the system. This time, our algorithm performs in average 5.9 % worse than the fixed cycle controller. Next, we consider computing control parameter values for each of the traffic demands separately. Again, we use interpolation and extrapolation to determine near-optimal control parameter values for the arrival rates that are between the midpoints of the traffic regimes. The comparison of the performance of our control algorithm and the fixed cycle controller using the grid network of Figure 5.14 is presented in Figure 5.15.

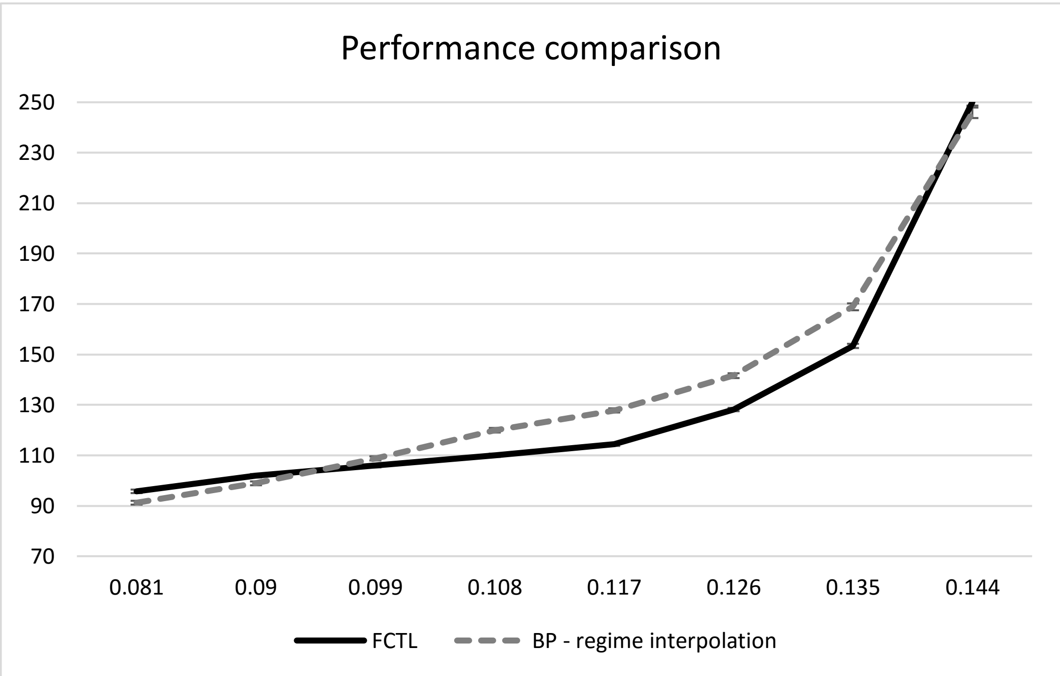


Figure 5.15 Performance comparison in the grid network using optimized control parameter values

We conclude that for the grid network our control algorithm fails to improve the average delay in the system. In average, the use of our control algorithm leads to an increase of 4.3 % in the average delay in the system. We see that our system improves the average delay for three levels of traffic load $\lambda = 0.081, \lambda = 0.09$ and $\lambda = 0.144$. The reductions in average delay for these cases are 4.7 %, 3.0 % and 1.7 %, respectively. For the remaining traffic loads our control algorithm performs worse than the fixed controller.

There are a few possible explanations on the discrepancy between the results concerning the linear and the grid networks. Firstly, the fixed control cycles we computed using Webster's method are probably very close to the optimal control of the network but our control algorithm is not able to reach this control with all traffic loads. This is suggested by the fact that we do not see a traffic demand after which the performance of our control algorithm deteriorates rapidly. In the case of the linear network, our controller initially performs better than the fixed controller but fails completely for heavy traffic loads. Now, our controller performs better for first two traffic loads and again for the highest traffic demand. The magnitude of the deterioration for the demand rates that are between these values is roughly constant, approximately 10 %.

A second explanation for the result is that something in the structure of the simulation network, the arrival rates and the turn fractions breaks the assumptions of our models or simply creates such illogical situation that our controller cannot come up with good green time allocation. For example, it is possible that the fixed cycle times accidentally and inadvertently create a green wave that allows the vehicles pass quickly through the network. The existence of such phenomenon would place our control algorithm in inferior position as we left the explicit coordination of the intersections outside of the scope of the algorithm development. It remains for the future research to consider different configurations, arrival rates, phase structure, turn fractions, to determine whether the results presented here reflect the actual performance of our controller or not. Such unexpected underlying phenomenon could explain also the weird results concerning the optimal values of the control parameters. The optimal value of the parameter η behaves differently than in the case of the linear network where our control algorithm managed to reduce the average delay.

5.4 Vehicle arrival distributions

So far we have used a Poisson process to generate the vehicle arrivals in the system. This means that the time between subsequent arrivals of vehicles are exponentially distributed. We know from queueing theory that for a queueing system with a fixed service process and given arrival rate, an increase in the variability of the arrival process leads to an increase in the average waiting time. If we consider an urban road network to be a general queueing system, we expect to see a similar relation between average delay and the variability of the vehicle arrival process. This is an important consideration as Poisson arrivals were chosen for modeling convenience to model a "fully random" arrival process. The actual arrival process to the urban network of Patna does not necessarily follow such a distribution. Consequently, it is possible that the performance improvement achieved is sensitive on the type of arrival process to the system. In order to evaluate this possible effect, we consider the linear network again and simulate other possible arrival processes. We use the near-optimal parameter values discovered earlier and quantify the effect that the variability of the vehicle arrival process has on the average delay encountered by the vehicles in the system. We also simulate the fixed cycle control approach to determine how the change in the arrival process affects the difference in the performances of the control methods.

A common way to characterize the variability of a distribution is to compute the so-called coefficient of variation, the ratio of standard deviation and mean:

$$CV = \frac{\sigma}{\mu} \tag{14}$$

In our simulations, the vehicles arrive individually at each peripheral inroad of the network. The arrival process follows Poisson distribution with parameter λ, i.e. the interarrival times are exponentially distributed with parameter λ. The mean of this exponential distribution is $\frac{1}{\lambda}$ and the variance $\frac{1}{\lambda^2}$. The coefficient of variation of the interarrival distribution is

$$CV = \sqrt{\frac{1}{\lambda^2}} \Big/ \frac{1}{\lambda} = 1$$

In addition to the Poisson process, we consider uniformly distributed interarrival times and arrivals following a fixed interval. To produce the desired long term demand, vehicles arriving at fixed intervals have to have $\frac{1}{\lambda}$ units of time between them. For the baseline case used in simulating network 1, this would mean that vehicles arrive $\frac{1}{0.06} \approx 16.67$ seconds between them. The variance, and the coefficient of variation, of these interarrival times is 0.

Another possible way is to simulate the arrival process is to assume that the interarrival time is uniformly distributed. The expected interarrival time has to be as computed above. It is common knowledge that the expected value of a uniformly distributed random variable is the midpoint of the range of possible values the random variable can attain. Consequently, we must have

$$X \sim uniform\left(0, \frac{2}{\lambda}\right),$$

where X represents the time between subsequent vehicle arrivals. The coefficient of variation of this random variable is

$$CV = \frac{\sqrt{\frac{\left(\frac{2}{\lambda} - 0\right)^2}{12}}}{\frac{0 + \frac{2}{\lambda}}{2}} = \frac{\sqrt{\frac{1}{3\lambda^2}}}{\frac{1}{\lambda}} = \frac{1}{\sqrt{3}} \approx 0.58$$

Considering the difference in the coefficient of variation of the input arrival distributions and the interpretation of urban road network as a queueing system, we expect the fixed interval arrival process to yield the lowest average delay in the system. The arrival process described by uniformly distributed interarrival times should result in higher amount of lost time but less than the original Poisson process. We simulate network 1 with the three different arrival processes and the near-optimal control parameters determined earlier. The results of this simulation experiment are shown in Figure 5.16.

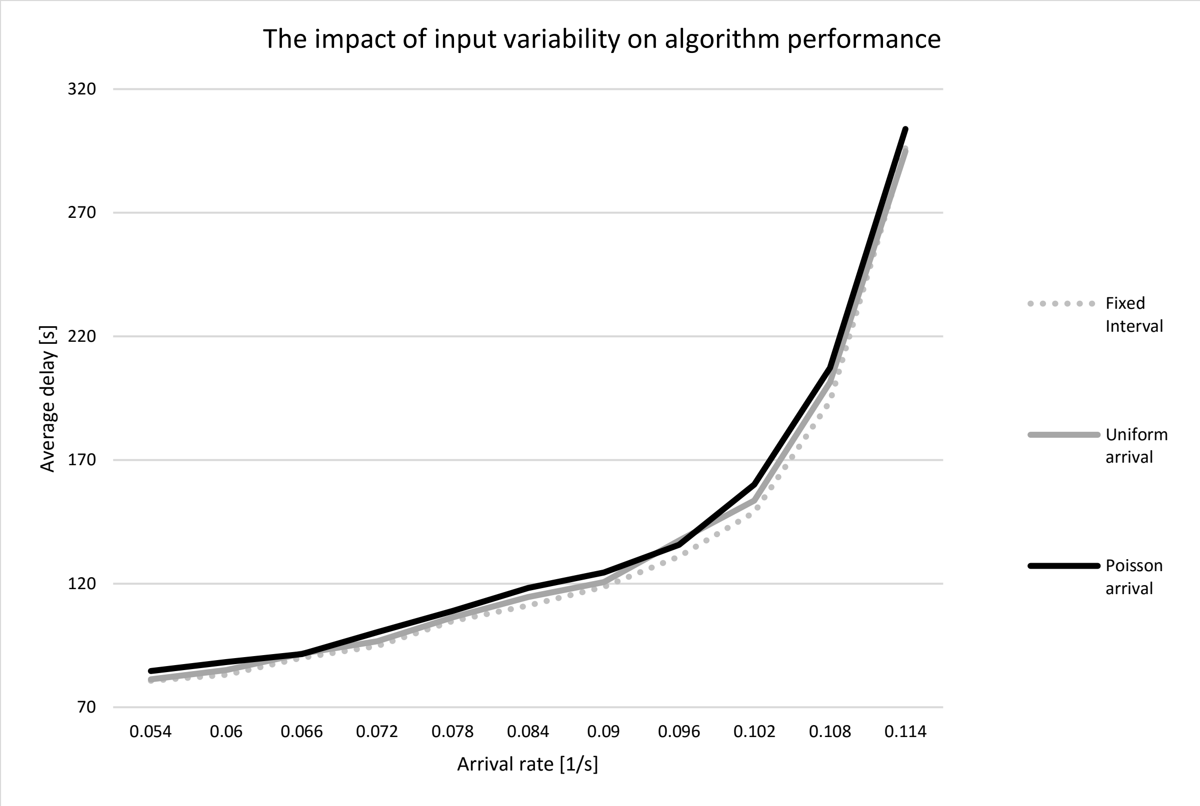


Figure 5.16 Comparison of average delay under different arrival processes under backpressure control

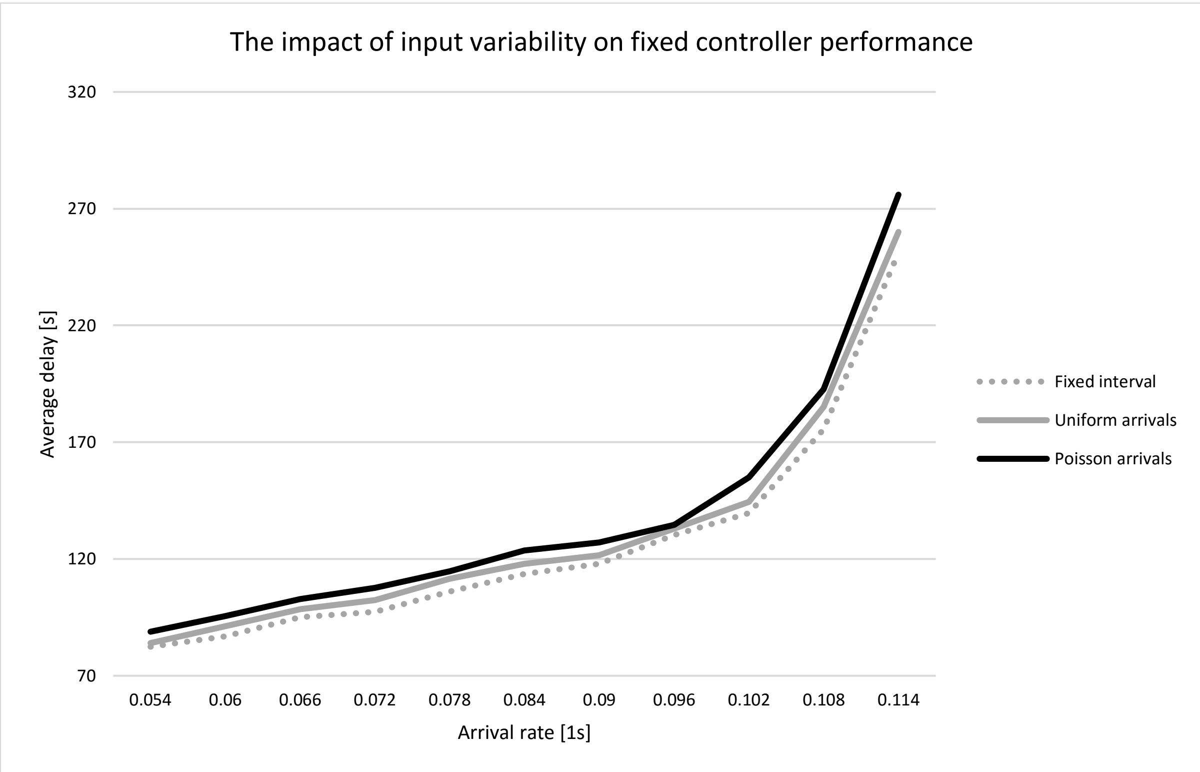


Figure 5.17 Comparison of average delay under different arrival processes and fixed control

We see that the average delay follows the relation we predicted by comparing the urban road network into a general queueing system. Keeping the arrival rate fixed, a decrease in the coefficient of variation of the arrival process leads to a decrease in the average delay in the system. The average delay under fixed cycle control behaves in a similar way as shown in Figure 5.17. Although both control methods exhibit the same phenomenon, it is important to consider how the coefficient of variation affects the relative performance of the methods. Table 5.3 presents the average improvement in average delay for light and medium traffic and for all the regimes achieved by switching from fixed control to our control algorithm.

Table 5.3 Relative performance of the control methods under different arrival processes

	Average improvement (light + medium)	**Average improvement (all demands)**
Poisson	5.07 %	1.77 %
Uniform	3.36 %	-0.16 %
Fixed interval	2.07 %	-1.67 %

In addition to reducing the average delay, the reduction in the coefficient of variation reduces the relative improvement gained by using the backpressure controller. For the combination of light and medium traffic regimes, we see the average improvement decreasing from 5.07 % to 2.07 %. When considering all levels of the demand, we see the improvement change sign from 1.77% for Poisson arrivals to -1.67 % for the fixed interval arrival. Clearly, there must be a minimum attainable delay that cannot be surpassed and with a shorter average delay to begin with, the improvements a smart controller can make become smaller. In addition, it seems obvious that the more irregular the arrival of vehicles is the better an adaptive controller will perform. Essentially, in the case of no variation in the vehicle arrivals, all information regarding the arrival process is contained in the mean interarrival time, the value that is used to generate the fixed timing plans. The adaptive nature of our control algorithm is not able to collect any additional information about the arrival process by monitoring the queue lengths. Moreover, when there are no short term variations in the arrival of vehicles, an adaptive controller loses the benefit it gains from the ability to capture short term dynamics in the traffic flow.

5.5 Control under perfect information

Throughout this research, the key problem and control limitation has been the heterogeneity of the traffic and the limitation of the currently available slightly inadequate sensor infrastructure. We chose to use an algorithm that utilizes the availability of the current queue lengths at an intersection and we then developed a measurement algorithm that can be used to estimate this value. We have now shown that depending on the network structure the use of this algorithm can lead to the reduction of the average delay in some systems. We noted, however, that the algorithm fails when facing very heavy traffic loads, and that the probable reason is that the queue length measurement algorithm is unable to estimate correctly the queue lengths when the intersections are constantly over utilized. Next, we consider a situation whereby our control algorithm has constantly access to perfect queue length information, i.e. our system knows the exact number of vehicles queueing at the incoming lanes. This analysis serves two purposes. Firstly, the accuracy of the queue length measurements is essentially an investment question. Good quality information can be achieved only through substantial investments in the sensor infrastructure. Secondly, the analysis serves as a theoretical insight on the information processing and traffic controlling capabilities of the underlying backpressure control principle. If the algorithm is able to perform well when given access to perfect information, we can conclude that the control principle of our algorithm is good, and the possible flaws in controlling heavy traffic loads are due to the limited information collected by our queue length measurement algorithm.

The recent literature has proposed methods that can be used to measure arbitrary and possibly heterogeneous vehicle queues. Narrow inductive loops can be placed vertically and tightly next to each other to allow detection of individual vehicles under heterogeneous traffic. Using such detector array could enable the use of vehicle counting and interpreting the queue length in a similar way as is currently done in the western traffic control. Another possible approach is the use of traffic measuring radars that can track multiple vehicles simultaneously and measure the length and composition of an arbitrary vehicle queue. Essentially, these methods could provide our control algorithm more accurate data on the prevailing traffic conditions. As our traffic control algorithm makes the control decisions based on the queue length measurement, we assume that more accurate information leads to better control decisions.

By quantifying the value of the more accurate information we can analyze the feasibility of investing in better traffic measurements. The cameras that are currently installed in Patna are cheap and easy to install and use. As they are placed above the road and share the poles with the actual traffic light equipment their installation requires minimal civil engineering and intrusion of traffic and the road infrastructure. Installing inductive loops requires intensive civil works as the loops are dug in the road and the road needs to be resurfaced. The radars compare with the cameras in the ease of installation but are more expensive to acquire and use. Clearly, there is a trade-off in the magnitude of the investment and the quality of the information acquired. The sensor infrastructure choices in possible future installations of this algorithm benefit from estimating the value of more accurate information. We analyze this trade-off by re-simulating the linear network and by giving the detectors the access to perfect queue length information. Although the additional information probably has an effect on the optimal values of the control parameters, we simulate the network using the control parameter values determined in Chapter 5.2.

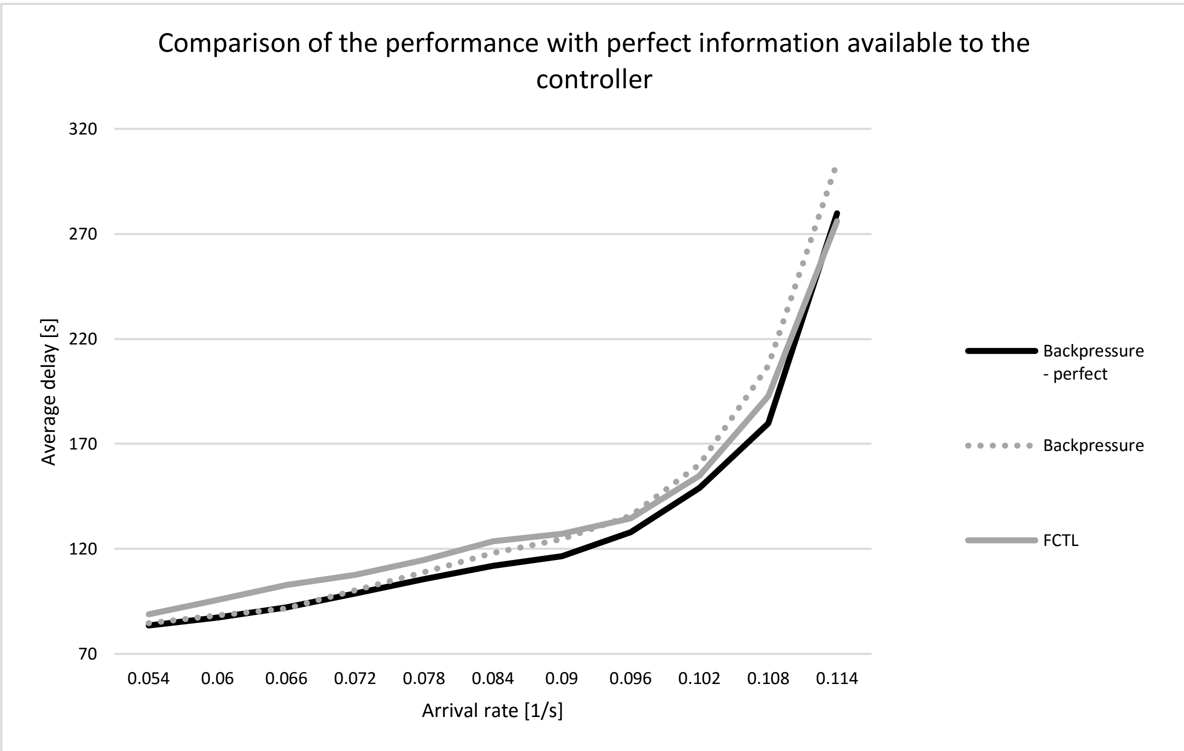


Figure 5.18 Performance of the control algorithm with perfect information

Figure 5.18 suggests that when our backpressure controller is aware of the exact queue lengths at the intersections the performance of the system is significantly improved. For lighter traffic demand, the system achieves same or slightly better performance than our original control method and the queue length estimation algorithm. With medium loads, the system with the additional information improves the average delay in the network significantly. The largest difference, nonetheless, is the performance under heavy traffic load. We see that the system that has the perfect queue length information is able to perform better than fixed traffic controller for all but the last demand case, where there is no significant difference between the performances of the two controllers.

The results suggest that the queue length measurement approach that we proposed fails to estimate the length of the queue for medium or heavy traffic loads. The underlying backpressure principles is applicable to all traffic loads and leads to improvement over the fixed controller if high quality information on the length of the queue is available. This justifies considering developing better queue length measurement methods and installing possibly better traffic measurement sensors. It remains for the future research, however, to determine the exact value of the additional information to support possible investment decisions.

5.6 Conclusions

In this chapter, we implemented a simulation model that can be used to test the behavior and measure the performance of our backpressure control algorithm. We used simulation based optimization methods to optimize the parameters η and α of our control algorithm. After this, we simulated the backpressure controller and a fixed cycle controller using the same traffic demand cases to compare their relative performance. We considered two network topologies: a linear networks represents an arterial route with intersections forming a chain with a clear direction of traffic progression, and a grid network representing intersections in the city center area that are highly connected to each other. The results of this chapter are twofold. By simulating the linear network, we found that when using the same cycle lengths, our control algorithm performs up to 7 % better and approximately 4 % better in average than a fixed cycle controller. The improvement in the performance is statistically significant. For the grid network, however, we found that our controller performs better than a fixed controller for some of the traffic demands, but performs 4.3 % worse in average than a fixed controller. In addition, we found out that the parameters used by our algorithm have to be optimized for the prevailing regime (light, medium, or heavy) of traffic.

By considering the so-called Gini coefficient and the coefficient of variation of the expected delay in the linear arterial network, we concluded that our backpressure controller is able to find network optimal green splits that slightly increase the variation in the service times between vehicles on major routes and vehicles on minor routes. We also analyzed the effect that the controller has on the average queue lengths at the intersections. We found that, although the algorithm reduces the average queue length in the system, some queue lengths increase. Somehow the algorithm is able to equalize the distribution of queue in a network-optimal way, reducing the queue lengths that are unnecessarily long and allowing queues grow on roads that are serviced “too well”. Such results are useful when considering the traffic conditions in Patna. One of the key challenges of traffic control in the highly saturated traffic conditions is to prevent the occurrence of excessively long queues that can cause gridlock situations.

Our analysis suggests that the performance of our traffic control algorithm deteriorates as the traffic becomes more and more congested. With the unstably busy traffic situation we simulated our control algorithm performed significantly worse than a simple fixed cycle controller. Although we showed that this is probably due to the queue length estimation method that we used, the deterioration of the performance leads to a more fundamental question that concerns the control of overly saturated traffic. The value of the optimal control parameters suggests that more stable and uniform distribution of green time becomes desirable. Essentially, our control algorithm tries replicating fixed cycle control as good as it can. Probably, in continuously over saturated traffic it is beneficial to provide constant and stable control that utilizes the intersection capacity as good as possible. Small changes that an adaptive controller makes possibly lead to instabilities that the heavy traffic load exaggerates and disperses through the network. For example, an opportunistic short term slight change in green time allocations perceived beneficial by an adaptive controller can lead to the deterioration of the service rate of another road. Because of heavy traffic load, this can lead to explosive growth of the queue length. If this queue grows long enough it can block upstream intersections leading to gridlocks that are unrecoverable by decentralized and adaptive decision logic.

This phenomenon is also identified in the literature. In general, it seems that various adaptive urban traffic control systems, even the existing commercial solutions, that provide significant improvement in performance for manageable traffic loads, cause deterioration of traffic conditions in the over saturated conditions. Consequently, our results agree with the results of the research recognizing that the control of over saturated traffic requires a completely different approach altogether. Most existing literature, like this research too, model the state of the traffic using the number of individual vehicles on the roads of the urban road network. Optimization methods are then used to compute optimal ways to guide these vehicles to their destinations through the network. In the overly saturated traffic conditions, however, the length of the queues grow unboundedly leading to the concept of a queue losing its interpretation. Probably, the control in these situations should concentrate on maximizing the capacity of the intersection and minimizing the risk of queue spillbacks and the formation of gridlocks. Such control, nevertheless, will prove difficult in Patna, as the current configuration of sensors enables only the collection of very simplistic stop line occupancy data.

We simulated both of the networks with stable arrival processes. Even though the arrivals are random, there is no medium to long-term variation in the arrival rates. We did this to simplify the simulation experiments but this choice probably prefers the fixed controller. The key idea behind our algorithm is that it is able to adapt to changes in the traffic, whereas a fixed controller is unable to react in a change in the long-term arrival rates. Probably, simulating the networks with a demand that is changing as a function of time, like the actual traffic in a real urban network does, our algorithm will probably perform better than a fixed controller.

In addition to comparing the performance of the backpressure controller and the fixed controller, we considered the impact that the variability of the input arrival process has on the average performance of the system. We showed that an urban road network behaves in a similar way to a general queueing system, i.e. the reduction in the coefficient of variation of the input process reduces the average waiting time in the system, and vice versa. This has an interesting connection to the recent traffic research that aims to aggregate regions of urban road network and describe their behavior and the traffic dynamics on a network level. We showed that, as expected, the magnitude of the performance improvement is sensitive to the choice of the input distribution. The results also indicated, however, that even with fixed deterministic arrival process our control algorithm can improve the traffic flow and that the more random the arrival process is the better our algorithm performs when comparing to simple fixed cycle controller. As the choice of the input distribution affects the control algorithm, future research is required to analyze the stochasticity of the arrival process of the vehicles entering the traffic light controlled network in Patna.

6 Conclusions and Discussions

In chapter 1.8, we formulated 4 research questions whose answers taken together satisfy our research goal. These research questions were answered in the subsequent chapters. In Chapter 2, we analyzed the traffic composition and behavior, the topology of the urban road network, and the available traffic sensors in the city of Patna. In Chapter 3, we conducted a structured literature review to analyze the current scientific knowledge concerning urban traffic control. We designed our traffic control algorithm in Chapter 4 and in Chapter 5 we tested its performance. For the detailed answers of each research question we refer to the conclusions of each of the prior chapters. In this chapter, we briefly summarize the answers of each of the research questions. After this we consider the theoretical results of this research and consider directions for future research. We conclude the chapter with a discussion of the practical results of this research from the perspective of ARS T&TT.

What is the current traffic situation in Patna?

Currently, Patna is suffering from severe traffic congestion problems leading to high amounts of lost time in the traffic and increased pollution. One of the causes of this problem is the total absence of traffic lights. Another constituent of the problem is the heterogeneity of the traffic. The traffic flow consists of many vehicle classes that differ in their physical and behavioral aspects. In addition, the local traffic culture differs from the traditionally associated with smooth traffic. This heterogeneity is key problem in controlling the traffic in Patna. Fundamentally, heterogeneous traffic responds to traffic light control input similarly to western homogeneous traffic. The control challenge is, however, that the traffic measurements traditionally used to implement adaptive controllers do not directly translate to the heterogeneous traffic. Consequently, our control approach has to operate with limited information about the prevailing traffic conditions. According to the requirements set by the problem owner, the algorithm should operate in real-time, be robust against hardware and communication network failures, adapt to changing traffic and provide stable control reducing the average delay in the urban network.

What solution approaches exist in literature to control a network of traffic lights?

The approaches used in recent literature can be categorized in three groups: decentralized approaches, computational intelligence methods, and multi-agent approaches. Decentralized control recognizes the difficulty of making optimal network-wide control decisions. Instead, the emphasis is placed upon designing control policies that are executed on local, intersection, level but are able to guarantee certain network-level performance. Computational intelligence methods try to emulate the way a smart human controller would make traffic light control decisions. Artificial intelligence and machine learning techniques are utilized to infer optimal control decisions from traffic data. Multi-agent control, a more recent approach, solves large decision problems through collaborative decision making. Each entity in the problem, intersections in the context of traffic control, makes its own decisions. In addition to simple decision making, these local controllers, called agents, communicate and share information with each other. For controlling the network of traffic lights in Patna, decentralized approaches are most useful. These methods score high on robustness, scalability and real-time decision making ability, key requirements for controlling the traffic in Patna.

How can we control the network of traffic lights in Patna?

Our control algorithm, called History-based Cyclic-phase Backpressure is based on a stochastic routing algorithm. The algorithm computes for each service phase in an intersection the difference in the queue length at the intersection and the queue lengths at the downstream intersections. This difference is interpreted as a “pressure difference” that the intersection controller wants to discharge by allocating green service time to the phases. We proposed using a cyclic version of this algorithm to reduce the possible inequality in the service rate, and to guarantee that also vehicles on minor routes with lower average demand rate are service in a reasonable amount of time. The cyclic algorithm is also easier to implement, as the completion of a service cycle constitutes a natural moment to share information between intersections and make the control decision for future phases.

In addition to the control algorithm, we proposed a way to estimate the current queue lengths on the inroads of the intersections. We solved the limitations of the traffic sensors and the challenge posed by the heterogeneity of the traffic by proposing a method based on measuring the length of a moving queue. This method is based on the fact that the interpretation of time gap between vehicles is invariant in the homogeneous and heterogeneous traffic cases. Our method also assumes that the length of queues in subsequent traffic light cycles resemble each other. In order to reduce the possible effects of cyclic changes or short-term variations in the queue lengths, we propose using a smoothing method to compute queue length estimates from historical data.

How does the proposed algorithm perform?

We used simulation based optimization techniques to optimize the parameters of the control algorithm. The optimal values of the parameters depend on the intensity of the traffic. For light traffic cases, high amount of adaptation is beneficial. The amount of smoothing used behaves in the opposite way. We compared the performance of the control algorithm by using it with the near-optimal parameters against a fixed cycle controller, a benchmark controller often used in the traffic literature. We conducted the performance comparisons using two different traffic network topologies. A linear network represents an arterial route with a clearly defined major traffic direction, and a grid network representing the tightly linked intersections in a city center area. We concluded that for the linear network our control algorithm reduces the average delay in the system by 1.8 % and for the grid network increases the average delay by 4.3 %.

We also concluded that the backpressure controller slightly reduces the “fairness” of the distribution of delay in the system. It seems that the algorithm is able to determine the existence of routes with major traffic flows and reduce the average delay encountered along them. This leads to the reduction of the average delay in the system. We measured the average queue lengths in the system to conclude that the algorithm is able to find more efficient distribution of queues in the network, and that the improved throughput of the system reduces the number of vehicles queueing in the system.

We also demonstrated that the performance of the system depends on the variability of the vehicle arrival process. The more random the arrival process of the vehicles to the system, the better our controller performs in comparison to the fixed controller. With completely deterministic arrivals, our algorithm fails to improve the average delay. Moreover, we established that giving our controller the access to more accurate information about the actual queue length at the intersection leads to an improvement in the performance. This suggests that the relatively poor performance of our control algorithm is due to the queue length estimation method that we used and the limited available detector information.

6.1 Theoretical results

Our research is one of the first reported attempts to control traffic under highly heterogeneous traffic. We conclude that the occupancy of a detector loop has essentially the same interpretation in both homogeneous and heterogeneous traffic and propose using this as an indicator of the current traffic conditions. We then proposed a way of estimating queue lengths using only the limited information of stop line mounted detectors. Our method relies on allowing the queue of vehicles to move and estimate the number of passing vehicles as they cross the stop line of the intersection. Whereas most existing methods conduct these measurements at an upstream location to provide the controller a "look-ahead horizon", we conduct the detection only at the stop line. Our approach generates an artificial look-ahead capacity by using the historical measurement of the queue length as an estimate of the current queue length.

Our research contributes also to the growing interest towards decentralized and agent based decision making. We show that decisions made locally using limited network information can provide desirable network-level effects and good global performance. In addition, we take a method proposed in the literature for a very simple network topology and control effort and generalize it to fit an arbitrary network and intersection topology and an arbitrary service phase structure.

6.2 Further development of the system

In this research we made multiple simplifying assumptions. They were necessary to restrict the scope of the project to fit the available amount of time. Urban traffic control systems, however, are complex and large computer systems. Essentially, the simplifying assumptions we made in this research were used to bore down to what we propose to be the core of the ARS urban traffic control system. More research is needed in the future to fill the gaps left by this research and provide the necessary functionality. Next, we propose a framework for the control architecture and then briefly discuss some possible and necessary research topics. This discussion serves the purpose of pointing out development steps that are necessary before the implementation of our traffic control system.

Currently, our traffic control algorithm makes green split decisions. From general planning perspective, split length decisions can be seen as capacity allocation problem. This means that with given limited capacity, i.e. available green time, the algorithm measures the prevailing traffic conditions and tries to allocate the green time to the phases in such a way that the waiting time is minimized. These green split time decisions are made autonomously and independently at each of the intersections, but theoretically, we know that the policy guarantees also network-wide performance.

The length of the cycle time, which was left outside of the scope of this research, is a capacity problem. The decision concerns setting the cycle time long enough so that the service capacity, i.e. the green time during the cycle, is long enough to satisfy the medium-term traffic demand. cycle time. Because average traffic demand changes slower than the queue lengths at the intersections these decisions are not made as often and belong to a higher level of decision hierarchy. Essentially, our traffic controller allocates green time *given* a cycle time. In the future, the controller should be supplemented with a module that is able to make cycle length decisions.

Coordination between the intersections is shown to reduce the average delays in urban traffic systems significantly but because of the limited scope we did not consider explicit coordination between the traffic lights in this research. The coordination between intersections is a more complex problem as it depends on both the length of the cycle time and the allocation of green splits. Intersections can be coordinated by computing offset values that allow the creation of continuous progression of vehicles. Knowledge of both cycle times and the green splits at the intersections is requires to estimate the time it takes for vehicles to reach the downstream intersections.

In Figure 6.1 we propose a schematic illustration of how the intersection controller proposed in Chapter 4 could be supplemented with additional decision logic to make our controller into a full-feature urban traffic control system. The boxes represent decision modules that will provide the required functionality, with the bolded boxes representing modules that are already implemented in the current control algorithm. Arrows in the drawing represent flow of information. The grey texts indicate the type of decision made or measurement collected by a module.

Like we did with the current intersection controller, we propose implementing this extended controller as a decentralized multi-agent system. Each intersection is given an individual controller with the task of determining locally optimal cycle length and green split. Depending on the choice of the network control approach, these intersections can either choose to use the locally determined cycle length, or communicate to choose a network-wide or sub-network-wide common cycle time that can improve coordination between the intersections. In order to determine optimal offset values, the intersection agents can communicate with each other. The agent can recognize the most optimal traffic direction for coordination and collaborate with the downstream intersection to determine optimal value for the offset.

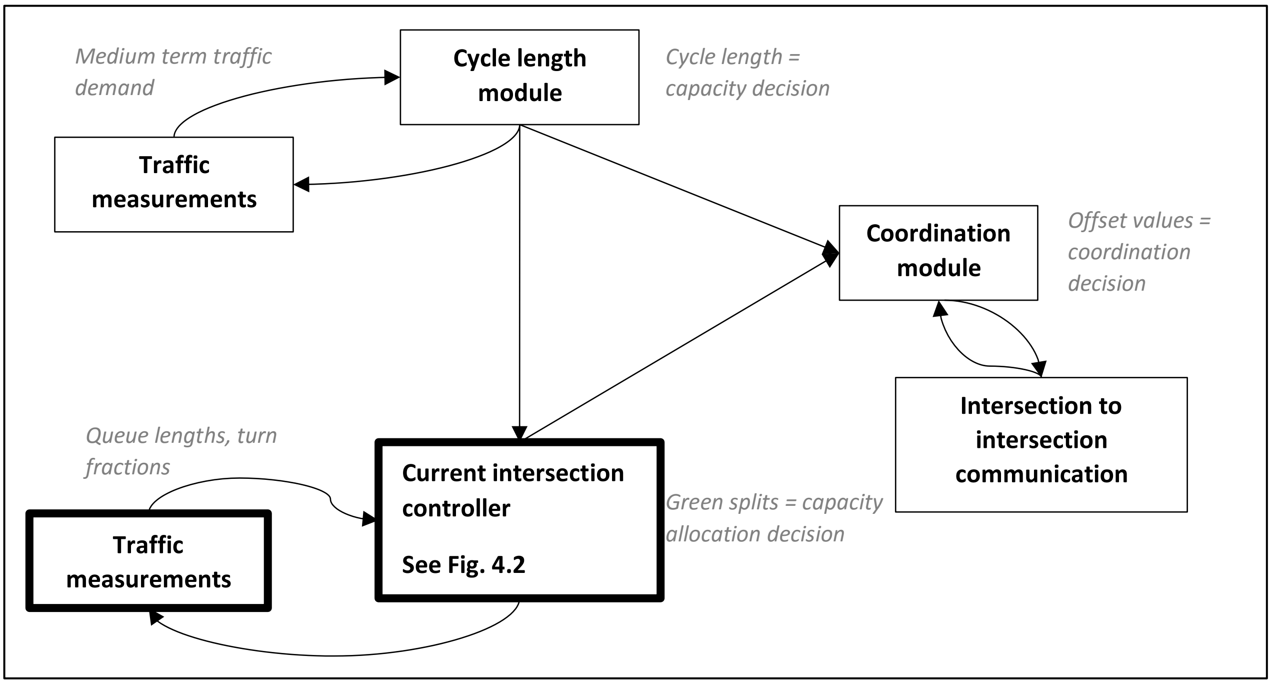


Figure 6.1 A proposed architecture of the future controller

6.2.1 Topics for future research

When it comes to implementing our traffic control algorithm, probably the most limiting assumption was the use of fixed network-wide cycle times. Although this is strictly not required by our algorithm, we used this simplification as our algorithm takes the intersection cycle time as input and is currently unable to estimate its goodness or propose modification to the cycle time. Existing commercial urban traffic control systems often use network-wide cycle times as a method of improving the coordination of the traffic lights. Some systems choose the optimal cycle time for subnetworks and might allow integer multiplies of cycle times at intersections where the traffic load differs significantly from the average load in the system.

Deciding the length of the cycle times is a capacity decision. During the full cycle of traffic light some of the cycle time is lost to phases that do not serve vehicles, i.e. the yellow and red phases. As we increase the length of the cycle time the proportion of the lost time during the cycle decreases and the service capacity of the intersection approaches the maximum capacity of the intersection. Consequently, the busier the traffic is, the longer the cycle time has to be to provide sufficient capacity.

One approach to do online optimization of the cycle time using our algorithm is to initialize the system with common, sufficiently long cycle time. After the system is running, we can use the platoon length measurements as indicators of the prevailing saturation of the traffic. If the complete green time is necessary to serve a lane, then this suggests that the service capacity allocated to this lane is possibly not sufficient. There are two ways to fix this problem. Either more of the cycle time is allocated to this lane, or the total cycle time has to be lengthened increasing the split time indirectly. As the former approach is considered automatically by our throughput optimal control algorithm, fixing possible lack of capacity should be done by increasing the total cycle time. As one possible approach we suggest the approach used in CoSiCoSt, namely, our system uses the virtual loops to determine whether or not queues exist on a lane after the end of the service phase. If the queue was not fully served during the phase, the capacity is not sufficient. If certain queue is continuously only partially served (say, during last three cycles) the cycle time of the intersection has to be increased. The extension of the cycle time could be done by lengthening the cycle time by a unit extension time.

Similarly, if all queues on each of the lanes are continuously completely served, some capacity is possibly wasted. If no queues are forming, as indicated by the existence of queues after completion of the associated service phase, the intersection can afford to reduce the length of the cycle time. Although making the cycle time shorter increases the proportional amount of lost time, the amount of time the vehicles wait of their service to start is reduced. The future research should address the optimal determination of the need of cycle time extension/shortening and the length of unit. In the architecture proposed in Figure 6.1 the determination of the optimal cycle time is implemented in the *Cycle length module.*

In addition to the cycle length decision, we scoped the explicit consideration of coordination between the intersections outside of this thesis. So far, our algorithm considers coordination indirectly by implementing intersection-to-intersection communication to share queue length measurements. In addition, our algorithm considers the state of the queues at the adjacent intersections in the decision making, and uses fixed, network-wide cycle times, the simplest and most widely used method of coordination. An often used method of improving the coordination between the intersections is the use of green waves. This is achieved by computing offset times between the initiation of green lights on major traffic directions. Green waves are known to reduce the delay on the key traffic directions on linear arterial routes, and possibly slightly deteriorate the delays on the routes that cross the major routes perpendicularly. Because our algorithm does not change the phase order at the intersections, and fixed cycle times are used, it is possible to extend our control algorithm to use offset values at the intersections. Care must be taken, however, in the determination of the offset values as our algorithm dynamically adjusts the split times. This slightly changes the moment of initiation of the phases that are meant to be coordinated in a green wave.

Additional research must be done to come up with a good real-time method for determining offset values between intersections. This method has to be able to consider the dynamic adjustment of the split times. Moreover, the algorithm should be able to determine the optimal direction of traffic to be coordinated. One possible way to approach this problem is to utilize the inherent multi-agent structure of our control approach. Each intersection can independently determine its locally most important traffic flow and "negotiate" with the downstream intersection controller to create a coordinated link between the intersections. In general, Lämmer and Helbing (2010) claims that with sufficient control incentive and communications protocol local controllers are able to create green wave like behavior. Additional research is required to determine the extent to which this is true for the cyclic variant of local control used in our control algorithm. The functionality of communicating with neighboring intersections and determining the optimal offset value for the intersection is implemented in the *Coordination module* in the architecture proposed in Figure 6.1.

Our analysis pointed out that the key reason that the algorithm we proposed does not work well with heavy traffic loads is the fact that the queue length estimation method fails to establish a good estimation of the number of vehicles waiting in the queue. We also showed that given perfect knowledge of the length of the queue at the intersections our control algorithm is able to perform significantly better than a fixed controller even for heavy traffic loads. Continuing this analysis is necessary to determine the extent to which additional information can be used to reduce the average delay in the system. Our simulations concerning the value of the additional information assumed that the system knows the exact number of vehicles in the queues. Although more sophisticated methods for determining the length of the queue exist or can be developed in the near future, an error is always associated to the measurements. It is important to consider the effect that the possible bias or error in the additional information has on the performance of our control algorithm before making sensor investment decisions.

Additional research has to be done to improve the performance of the queue length estimation method proposed in this research. If the virtual loops installed on the stop lines are used, allowing the queue to move like we proposed in this research is probably the only way to estimate the number of vehicles in the queue. In order to improve the performance of this algorithm for higher traffic loads, some additional considerations have to be made. One option is to use the approach we described as an option for the cycle length determination. Our queue length estimation method fails when the queue is longer than the allocated green time. We can use the cameras to determine whether vehicles are still queueing at the intersection at the moment serving the lane has ended. If vehicles still exist on the lane the platoon on that lane was not fully served and the queue length estimate underestimates the length of the queue. The estimate can then be corrected by, for example, adding a constant correction term. Research has to be conducted to determine the optimal way of correcting the queue length estimate.

In addition to the queue lengths, the turn fractions at the intersections are an important input value to our algorithm. In the thesis, we simplified the control problem by assuming that the system knows the long-term turn fractions and that the turn fractions remain constant. Obviously, such limitations should be solved before implementing our control method. Moreover, we assume that being able to estimate the short-term turn fractions at the intersection could improve the performance of the algorithm. Fortunately, our control algorithm simply uses the turn fractions as input value and is, thus, indifferent when it comes to the method determining the turn fractions. One possible way to determine the routes that the vehicles take through the network is to use automatic license plate recognition. By placing recognition cameras at certain points in the network, the system can partly reconstruct the routes that some vehicles take through the network. This, in turn, can be used to construct an estimate of the routing probabilities at the intersections. Technically, our control algorithm assumes that the estimator of the turn fractions is unbiased, but this is unattainable in a real world implementation. Future research should be conducted to establish the value, i.e. reduction in average waiting time, of having real-time measurements of the true turn fractions at the intersection. Moreover, future research should establish the sensitivity of the performance of our control algorithm to possible bias or error in the turn fraction values.

Probably the most important topic for the future research concerns determining the optimal values of η and α. Our simulation studies show that the performance of the system depends on choosing the values of these parameters to fit the current traffic load. We have also shown that linear interpolation and extrapolation can be used to smoothly transfer between traffic regimes. It is, however, important to find a better way to relate the optimal values of the parameters to the prevailing traffic load. In addition, the future research should come up with an algorithm of estimating the traffic load, and consequently justify the choice of the control parameter values, from the limited sensor measurements available. The ability to estimate the traffic load is also important for the determination of the cycle time and the offset values. Like stated above, the system has to choose the cycle time as a function of the traffic load. In addition, the optimal value of the offset values between the intersections depend on the traffic load as it affects the expected travel time between the intersections.

In addition to the development work that is required to implement the control algorithm, some additional research should be conducted with the current controller and the network structure. The grid network should be simulated with different demand and arrival patterns to determine whether the inconsistent results we had can be explained by a randomness or if the control algorithm fails to control that type of network altogether. Also the linear network should be considered with multiple other traffic demand cases to analyze the statistical strength of our conclusion. Moreover, so far we have considered only stable random arrival processes. In order to get a more accurate estimation of the real-world performance of the algorithm, the future research should simulate the networks with arrival rates that change as a function of time. This will make it more difficult for both of the controllers to determine good green time allocations. We expect that our adaptive controller will then perform considerably better than a fixed cycle controller.

7 References

Ahmad, A., Arshad, R., Mahmud, S. A., Khan, G. M., & Al-Raweshidy, H. A. (2014). Earliest-Deadline-Based Scheduling to Reduce Urban Traffic Congestion. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 4, pg. 1510 - 1526.

Arasan, V. T., & Koshy, R. Z. (2005). Methodology for Modeling Highly Heterogeneous Traffic Flow. *Journal of Transportation Engineering*, Vol 131, Issue 7, pg. 544 - 551.

ARS T&TT. (2014). Patna: Installation of Traffic Lights. *Internal Traffic Survey Report*.

ARS T&TT. (2016, February 2). *About ARS*. Retrieved from ARS T&TT Web site: http://www.ars-traffic.com/about-ars

Balaji, P. G., German, X., & Srinivasan, D. (2010). Urban traffic signal control using reinforcement learning agents. *IET Intelligent Transport Systems*, Vol. 4, Iss. 3, pp. 177 - 183.

Barcelos de Olivera, L., & Camponogara, E. (2010). Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C*, Vol. 18, pp. 120 - 139.

Bell, M. (1992). Future Directions in Traffic Signal Control. *Transportation Research Part A*, Vol. 26A, No. 4, pp. 303 - 313.

Benhamza, K., & Seridi, H. (2015). Adaptive traffic signal control in multiple intersection network. *Journal of Intelligent & Fuzzy Systems*, Vol. 28 pp. 2557 - 2567.

Burguillo-Rial, J. C., Rodríguez-Hernández, P. S., Costa-Montenegro, E., & Gil-Castiñeira, F. (2009). History-based self-organizing traffic lights. *Computing and Informatics* , Vol. 28 pg. 1001 - 1012.

Cascetta, E., Gallo, M., & Montella, B. (2006). Models and algorithms for the optimization of signal settings on urban networks with stochastic assingment models. *Annals of Operations Research*, Vol. 144, pg. 301 - 328.

CDAC. (2014). *Area Traffic Control System - CoSiCoSt Technical Specifications.*

Centraal Bureau voor de Statistiek. (2016, January 3). *Motor vehicles; type, age class, 1 January*. Retrieved from StatLine Database: http://statline.cbs.nl/Statweb/publication/?DM=SLEN&PA=82044ENG&D1=0-1,3-9,12&D2=0&D3=12&LA=EN&VW=T

Chin, Y. K., Yong, K. C., Bolong, N., Yang, S. S., & Teo, K. T. (2011). Multiple Intersection Traffic Signal Timing Optimization with Genetic Algorithm. *IEEE Conference on Control System, Computing and Engineering*, pp. 454 - 459.

Dai,, Y., Hu, J., & Zhao, D. (2011). Neural Network Based Online Traffic Signal Controller Design with Reinforcement Training. *14th International IEEE Conference on Intelligent Transportation Systems*, 1045 - 1050.

Department of Transport. (2016, February 11). *Flow weighted vehicle speeds (CGN02).* Retrieved April 6, 2016, from UK Government Web Site: https://www.gov.uk/government/statistical-data-sets/cgn02-flow-weighted-vehicle-speeds

Douma, A. (2008). Aligning the Operations of Barges and Terminals through Distributed Planning. *PhD - thesis*.

El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2013). Multi Agent Reinforcement Learning for Integrated Networks of Adaptive Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems.*, Vol. 14 (3), pp. 1140 - 1149.

García-Nieto, J., Alba, E., & Carolina Olivera, A. (2012). Swam intelligence for traffic light scheduling: Application to real urban areas. . *Engineering Applications of Artificial Intelligence*, Vol. 25, pp. 274 - 283.

Gartner, N. (1983). OPAC: A Demand Responsive Strategy for Traffic Signal Control. *Transportation Research Record*, January 1983, pp. 45 - 81.

Gartner, N. (1985). Demand-responsive traffic signal control research. *Transportation Research Part A*, Vol. 19A, Issue 5/6, pp. 369 - 373.

Geroliminis, N., & Daganzo, C. F. (2008). Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B*, Vol. 42, pg. 759 - 770.

Geroliminis, N., & Sun, J. (2011). Properties of a well-defined macroscopic fundamental diagram for urban traffic. *Transportation Research Part B*, Vol. 45, pg. 605 - 617.

Gershenson, C. (2007). Desing and Control of Self-Organized Systems.

Gregoire, J., Qian, X., Frazzoli, E., de La Fortelle, A., & Wongpiromsarn, T. (2015). Capacity-Aware Backpressure Traffic Signal Control. *IEEE Transaction on Control of Network Systems*, Vol. 2, No. 2, pp. 164 - 173.

Hamilton, A., Waterson, B., Cherret, T., Robinson, A., & Snell, I. (2012). The evolution of urban traffic control: changing policy and technology. *Transportation Planning and Technology*, Vol. 36, No. 1, pp 24 - 43.

Henry, J. J., Farges, J. L., & Tuffal, J. (1983). The PRODYN Real Time Traffic Algorithm. *IFAC Control in Transportation Systems*, pp. 305 - 310.

Hunt, P. B., Robertson, D. I., Bretherton, R., & Winton, R. I. (1981). SCOOT - A Traffic Responsive Method of Coordinating Signals. *Transport and Road Research Laboratory Report 1014*.

Khan, S. I., & Maini, P. (1999). Modeling Heterogeneous Traffic Flow. *Transportation Research Record*, 1678, pg. 234 - 241.

Lämmer, S., & Helbing, D. (2008). Self-Control of Traffic Lights and Vehicle Flows in Urban Road Networks. *Journal of Statistical Mechanics: Theory and Experiment*, P04019.

Lämmer, S., & Helbing, D. (2010, March 25). *Self-Stabilizing Decentralized Signal Control of Realistic, Saturated Network Traffic.* Santa Fe Institute Web Site www.santafe.edu/media/workingpapers/10-09-019.pdf. Retrieved from Santa Fe Institute Web site: www.santafe.edu/media/workingpapers/10-09-019.pdf

Le, T., Kovács, P., Walton, N., Vu, H. L., Andrew, L. L., & Hoogendoorn, S. S. (2015). Decentralized signal control for urban road networks. *Transportation Research Part C*, Vol 58, pp. 431 - 450.

Lighthill, M. H., & Whitham, G. B. (1955). On Kinematic Waves II: A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society London*, Series A: Mathematical and Physical Sciences. pp. 317 - 345.

Mirchandani, P., & Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C*, Vol. 9, pp. 415 - 432.

Mu, H. B., Yu, J. N., & Liu, L. Z. (2010). Traffic Signals Control of Urban Traffic Intersection Group Based on Fuzzy Logic. *Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 763 - 767.

Nair, S. (2016, February 16). Interview of the local project manager.

NCHRP-Synthesis. (2010). *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice.* National Cooperative Highway Research Program.

Nuli, S., & Mathew, T. V. (2015). Vehicle Actuated Control for Heterogeneous Traffic Using Stop Line Detection. *European Transport \ Transporti Europei*, Issue, 57, Paper 2, pg. 1 - 15.

Osorio, C., & Bierlaire, M. (2008). A multiple model approach for traffic signal optimization in the city of Lausanne. *8th Swiss Transport Research Conference*.

Papadimitriou, C. H., & Tsitsiklis, J. (1994). The Complexity of Optimal Queueing Network Control. *Ninth Annual Conference on Structure of Complexity* .

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., & Wang, Y. (2003). Review of Road Traffic Control Strategies. *Proceedings of the IEEE, vol. 91, No. 12*, 2043-2067.

Protham, H., Branke, J., Schmeck, H., Tomforde, S., Rochner, F., Hähner, J., & Müller-Schloer, C. (2009). Organic traffic light control for urban road networks. *International Journal of Autonomous and Adaptive Communications Systems*, Vol. 2, Iss. 3, pp. 203 - 225.

Richards, P. (1956). Shock Waves on the Highway. *Operations Research*, Vol. 4, Issue 1, pg. 42 - 51.

Robertson, D. I., & Bretherton, R. (1991). Optimizing Networks of Traffic Signals in Real Time - The SCOOT Method. *IEEE Transactions on Vehicular Technology*, Vol. 4, Iss. 1 pp. 11 - 15.

Roozemond, D. (1999). Using Intelligent Agents for Dynamic Urban Traffic Control Systems.

Salomi, V. (2015, February 18). *Patna 2nd most polluted city after Delhi?* Retrieved from Times of India: http://timesofindia.indiatimes.com/city/patna/Patna-2nd-most-polluted-city-after-Delhi/articleshow/46972132.cms

Singh, S. (2012). Urban Transport in India: Issues, Challenges, and the Way Forward. *European Transport*, Issues 52. Paper 5.

Smith, S. F., Barlow, G. J., Xie, X.-F., & Rubinstein, Z. B. (2013). Smart Urban Signal Networks: Initial Application of the SURTRAC Adaptive Traffic Control System. *Proceedings of the 23rd International Conference on Automated Planning and Scheduling*, pp. 434 - 442.

Srinivasan, D., Choy, M. C., & Cheu, R. L. (2006). Neural Networks for Real-Time Traffic Signal Control. *IEEE Transactions on Intelligent Transportion Systems*, Vol. 7, Iss. 3, pp. 261 - 272.

Talab, H. S., Mohammadkhani, H., & Haddadnia, J. (2013). Controlling multi variable traffic light timing in an isolated intersection using a novel fuzzy algorithm. *Journal of Intelligent & Fuzzy Systems*, Vol. 25, pp. 103 - 116.

Tassiulas, L., & Ephremides, A. (1992). Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, Vol. 37, No. 12.

Turky, A. M., Ahmad, M. S., & Yusoff, M. Z. (2009). The Use of Genetic Algorithms for Traffic Light and Pedestrian Crossing Control. *IJCSNS International Journal of Computer Science and Network Security*, Vol. 9, No. 2, pg. 88 - 96.

van Katwijk, R. (2008). *Multi-Agent Look-Ahead Traffic-Adaptive Control.* PhD thesis.

Varaiya, P. (2013). Max pressure control of a network signalized intersections. *Transportation Research Part C*, Vol 36, pp. 177-195.

Webster, F. (1958). Traffic Signal Settings. *Road Research Technical Paper No. 39*.

Woensel, T. v., & Vandaele, N. (2007). Modelling traffic flows with queueing models: A review. *Asia-Pacific Journal of Operational Research*, Vol. 24, nr. 4, pp. 1 - 27.

Wooldridge, M., & Jennings, N. (1995). Intelligent Agents: Theory and practice. *The Knowledge Engineering Review*, Vol. 10, Iss. 2. pp. 115 - 152.

Xie, X. F., Smith, S. F., & Barlow, G. J. (2012). Schedule-driven intersection control. *Transportation Research Part C*, Vol. 24, pp. 168 - 189.

Yeh, E. M., & Berry, R. A. (2007). Throughput Optimal Control of Cooperative Relay Network. *IEEE Transactions on Information Theory*, Vol. 53, No. 10, pp. 3827 - 3833.

8 Appendix A

Webster's (1958) method uses an empirical formula to determine a near-optimal cycle time and the allocation of that cycle time into green splits. His approach starts by determining the degree of the saturation of each service phase. This is done by computing a critical flow factor for each of the lanes. Consider lane *l* with average demand d_l and saturation flow rate, or the service capacity, s_l. Then the degree of average utilization of the lane is:

$$y_l = \frac{d_l}{s_l}$$

Now, consider a phase *p* that serves a set of lanes $\mathcal{L}_p$. The maximum utilization of the phase is given by:

$$y^p = max_{l \in \mathcal{L}_p} y_l$$

Because the lanes served during one phase do not interact, this value describes the "critical" utilization of the phase. The degree of utilization of the intersection is then given by

$$Y = \sum_p y^p$$

Essentially, this value describes the utilization of the intersection. It is the sum of the utilization rates of each of the critical lanes per service phase. Based on simulation studies, Webster (1958) proposed a formula to compute the optimal cycle time based on the value *Y.* The optimal cycle time is:

$$C^* = \frac{1.5L + 5}{1 - Y},$$

where C^* is the optimal cycle time and L is the lost time in the cycle. Clearly, the formula can be used only if Y < 1, i.e. the intersection is able to serve the average demand that it faces. The lost time *L* is the fixed sum of the phases during which no vehicles are served. In our case, this is the sum of the duration of the yellow and red phases.