Multi-Agent Look-Ahead Traffic-Adaptive Control

R.T. van Katwijk

Multi-Agent Look-Ahead Traffic-Adaptive Control

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College van Promoties,
in het openbaar te verdedigen op dinsdag 29 januari 2008
om 15:00 uur door

Ronald Theodoor VAN KATWIJK
informatica ingenieur

geboren te Rotterdam.

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. ir. J. Hellendoorn
Prof. dr. ir. B. De Schutter

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. J. Hellendoorn	Technische Universiteit Delft, promotor
Prof. dr. ir. B. De Schutter	Technische Universiteit Delft, promotor
Prof. dr. H.J. van Zuylen	Technische Universiteit Delft
Prof. ir. L.H. Immers	Katholieke Universiteit Leuven
Prof. dr. ir. E.C. van Berkum	Universiteit Twente
Prof. dr. ing. F. Busch	Technische Universität München
Ing. P. van Koningsbruggen	Technolution
Prof. dr. R. Babuška, M.Sc.	Technische Universiteit Delft, reservelid

TRAIL Thesis Series T2008/3, The Netherlands TRAIL Research School
TRAIL Research School
P.O. Box 5017
2600 GA Delft
The Netherlands
T : + 31 (0) 15 27 86046
F : + 31 (0) 15 27 84333
E : info@rsTRAIL.nl

Published and distributed by: R.T. van Katwijk
E-mail: ronald.vankatwijk@tudelft.nl

ISBN 978-90-5584-098-4

Keywords: traffic, control, adaptive, look-ahead, coordination

Printed in The Netherlands

Abstract

The objective of this thesis is to create a distributed, multi-agent, approach to traffic control. This PhD thesis' focus is on the control of a network instrumented by traffic signals.

A thorough literature study has been performed, reviewing the current state of the art in traffic signal control. On the basis of this literature survey, a taxonomy of existing systems was constructed. The design of a traffic-adaptive control system is as well a science as an art. Along the way compromises have to be made in order to end up with a workable system that is not only able to come up with good signal timings, but is also able to deliver them on time. The taxonomy constructed of the various traffic-adaptive control algorithms is based both on the underlying principles and on the compromises that were made to come up with a workable, albeit less optimal system.

A new adaptive control algorithm is subsequently developed that incorporates the strong points of each of the algorithms reviewed. The algorithm determines a short term policy on the basis of a long-term analysis and considers the individual signal groups as the smallest controllable entity. Although state of the practice in vehicle-actuated control, look-ahead adaptive control still use stages as the smallest controllable entity, which reduced the flexibility of this approach. The developed algorithm is capable of controlling a single intersection, but can be configured for use in a network.

When configured for use in a network the controller shares its intentions regarding its control plan with nearby intersection controllers and informs them of traffic that it plans to release. In order to enable cooperation controllers must be willing to adjust their locally optimal control plan for the benefit of the network. In order to achieve this controllers are informed about the cost inflicted by them to nearby controllers. Using this information, intersection controllers can iteratively adjust their plan to the benefit of the network.

In order to evaluate the developed control algorithms a test bed was developed during the course of this thesis. The test bed was essential in the development and testing of the algorithm. The test bed was also used in a proof-of-concept study for the N470, whereas the performance of the algorithm was benchmarked for a corridor against freshly optimized traffic-actuated controllers.

Preface

As a fan of fantasy I am well aware that a story does not necessarily have to finish when the book ends. As books can be part of a series there is ample opportunity to finish unexplored story lines in subsequent parts of the series. The book that lies in front of you similarly does not finish the story I would like tell, but nevertheless provides closure to some of the more important story lines.

Pursuing a PhD is first and foremost a humbling experience. Having worked for some time in the field of traffic management I thought I would be off to a running start. However, as it turned out there is a large gap between the state of the practice and the state of the art in traffic control. Also, whereas traffic management and traffic control are used interchangeably they really denote completely different worlds that have yet to be brought together.

Nevertheless I can wholeheartedly recommend anyone that considers pursuing a PhD next to their regular work to do so. Right up to the end, when you are required to stop doing research and start writing up what you have done, I have always considered the time spent on research as a hobby. A hobby that at times went out of control, but a hobby nevertheless.

The writing part is the part that has been most stressful, especially since at that time I got the opportunity to apply the research done at the university into some TNO projects. Although fun, this clearly illustrated that there is too little time in a working week. Of course, time is relative. However, I have always tried to maintain the proper balance between my working life and my personal life. When the end of PhD drew nearer, this proved to be increasingly more difficult. Therefore I am not unhappy with the closure the thesis that lies before you provides.

There are a number of people that have made this thesis possible. Of these, I would like to thank Paul van Koningsbruggen and Ben Immers for seeding and nurturing my interest in possible applications of multi-agent systems for traffic management. I am furthermore grateful to my promoters Hans Hellendoorn and Bart De Schutter, as without their support I would not have started let alone finished this thesis. A special gratitude goes out to Bart De Schutter as he, being my daily supervisor, had to go through some of my more unstructured writings. I furthermore would like to thank the members of my committee for their interest in my PhD Thesis and taking the time and effort to review it.

I am indebted to my wife, Debbie, for her understanding, patience and continual support during the entire period of my PhD research, and especially in the difficult time of writing the thesis. I am also indebted to my now 1-year old son, Yoeri, small as he is, for being able to put things in the right perspective. After all, this thesis, although important, in the end is just another object to play with and to drool on.

Contents

Chapter 1

Introduction

The negative economic and societal consequences of the traffic congestion keep increasing. A reliable and efficient operation of the traffic network is thus of crucial importance for our society. In order to deal with congestion there are a number of options available: increasing the capacity of the roads (increasing the traffic supply), promoting the use of alternative modes of transportation (limiting the traffic demand), or making more efficient use of the existing infrastructure. The expansion of the existing infrastructure is however often prevented by spatial limitations and by European and national guidelines regarding the air quality.

Traffic management and control are a key tool to obtain a reliable and efficient traffic network. The "traffic market" functions optimally when there is a perfect matching between traffic supply and traffic demand. Currently, the traffic supply mainly consists of fixed infrastructure. However, the set of tools that enables the traffic operators to dynamically influence the traffic supply, such as ramp metering, route guidance, dynamic speed limits, VMS, etc., is still expanding, both in diversity of control measures and in number of installed traffic control set-ups. Additionally, more and more real-time information becomes available in the traffic control centers about the current traffic situation on the network. This implies that the traffic operators get more and more handles to influence and to control the traffic situation.

In cooperation with amongst others TNO, the Dutch Ministry of Transportation has developed the "Architectuur voor Verkeersbeheersing (AVB)" [10] (Traffic Control Architecture, see Figure 1.1), a conceptual and operational procedure that supports traffic operators. The AVB provides a structured decision approach that ultimately results in a set of specific traffic control measures that should be deployed. The common objective is to provide the best possible "quality" of performance of the traffic network for all road users, given operational constraints. To this aim the AVB tries to express the quality of performance of the traffic network in various objectives and target levels or constraints for the different parts of the network. In practice, this is a hard and difficult task due to the dynamics and complexity of the traffic process. The matching between the traffic demand and the available supply can therefore only partially be determined at the strategic and tactical level: the major part of the matching should occur at the operational level.

The majority of the traffic management instruments functions fully autonomously. As local disturbances in the traffic flow can be handled without appealing to other traffic con-

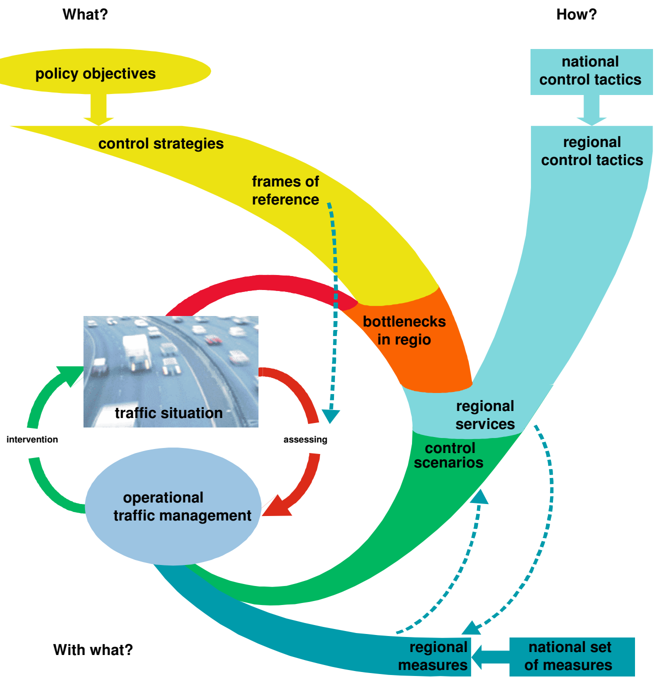


Figure 1.1: "Architectuur voor Verkeersbeheersing (AVB)" (Traffic Control Architecture)

trol instruments or to a higher traffic management level these disturbances can be handled more timely and thus more effectively. As more and more instruments are deployed, the probability of interference between control tools that are applied in the same area increases. Although a traffic system consisting of fully autonomous traffic-control instruments is able to deal with disturbances more timely, it cannot guarantee the effective functioning of the network as a whole. Most traffic control instruments are focused on promoting the traffic flow in their own region of influence. It is very well possible that disturbances are dealt with by moving them instead of solving them.

1.1 Main objective and research questions

In order to be able to guarantee the effective functioning of the network as whole, often a centralized or top-down approach to traffic management is employed. Theoretically this allows for a network-optimal setting of each traffic control instrument. The downside of the "top-down" approach is however that it is impossible to regularly fine-tune each individual control measure. This is largely due to the complexity of the control instruments and the frequency with which disturbances occur. It is intractable to effectively respond to all minor disturbances from a centralized traffic management center even with the support of advanced support systems.

As a centralized control approach is often not feasible in practice due to computational complexity, communication overhead, and lack of scalability, a distributed control approach is considered. A distributed approach can solve the shortcomings of the centralized ap-

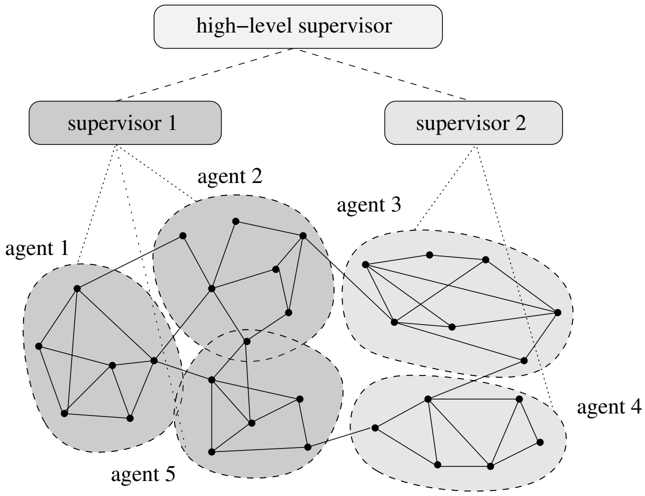


Figure 1.2: Representation of a hierarchical, multi-agent structure for traffic management and control.

proach to a large extent. Furthermore, a distributed approach is less susceptible to failures, and thus more robust.

A distributed approach requires that the control problem can be subdivided into several loosely coupled (or even independent) subproblems, such that the combination of all the solutions of the subproblems together approximate the solution of the original control problem. In our approach each of the subproblems is solved by a local control agent that has a large degree of autonomy to determine the most optimal control measures for the region it manages and within the set of control measures that it has at its disposition.

Of course, in order to prevent negative effects of the control measures in one region on the traffic situation in a neighboring region, there should be coordination among the neighboring control agents. This coordination is obtained through direct communication and negotiation between neighboring agents. The more global coordination of the agents on the regional or network level is then effectuated by supervisors, that operate at a higher level of the control architecture (see Figure 1.2).

The objective of this thesis is to create a distributed, multi-agent approach to traffic control. The focus of this thesis is on the distributed, multi-agent, control of a network instrumented by traffic signals as opposed to a network instrumented by many different types of control instruments. Traffic signals are one of the oldest and most common traffic control instruments available to the road operator, and they are also the most heavily researched and thus optimized traffic control measure. Traffic signal installations have evolved from fairly simple installations that operate under a fixed-time regime to complex installations that optimize the switching of traffic signals using information from multiple detectors located at the approaches to the intersection. Without underestimating the complexity of different types of control instruments, one could argue that when it is possible to create a distributed, multi-agent control system for this type of traffic control instrument, it can also be extended to include the other types of traffic control instruments.

1.2 Contributions to the state of the art

In this thesis we will develop a multi-agent framework for the distributed and coordinated control of a traffic network. The main theoretical and innovative contributions of this thesis with respect to the current state of the art are:

- Development of a taxonomy of current approaches to traffic signal control.
- Development of a test bed for distributed multi-agent traffic control concepts that complies to the standards as set forth by the Foundation of Intelligent Physical Agents (FIPA) and that supports state of the art microsimulation models (i.e., Paramics and AIMSUN).
- Development of a look-ahead traffic-adaptive signal controller that uses the more flexible movement-based approach as opposed to the stage-based approach used by the state of the art in look-ahead traffic-adaptive control.
- Development of a distributed and coordinated traffic control approach for networks controlled by a heterogeneous set of traffic control instruments. Coordination procedures are developed on the basis of which the actions of nearby controllers can be coordinated on the microscopic level (i.e., on the level of individual vehicles) and on the macroscopic level (i.e., on the level of flows).

1.3 Traffic signal control

Traffic signals have a long history. On 10 December 1868, the first traffic signals were installed outside the British Houses of Parliament in London. They resembled the railway signals of the time, with semaphore arms and red and green gas lamps for night use. The gas lantern was turned with a lever at its base so that the appropriate light faced traffic. Unfortunately, it exploded on 2 January 1869, injuring the policeman who was operating it.

The modern electric traffic signal is an American invention. As early as 1912 in Salt Lake City, Utah, policeman Lester Wire set up the first red-green electric traffic signal. The first patent in the field of traffic signals was filed by James Hoge, on 22 September 1913. The first interconnected traffic signal system was installed in Salt Lake City in 1917, with six connected intersections controlled simultaneously from a manual switch. Detroit saw the first traffic signals that used a yellow signal in addition to the red and green signal in 1920.

Automatic control of interconnected traffic signals was introduced March 1922 in Houston, Texas. In 1952, in Denver analogue computers were first used to switch between different control plans on the basis of detector information. The first use of a digital computer for controlling traffic occurred in 1959, in Toronto. For more information regarding the history of traffic signals see [82, 125].

Nowadays, in The Netherlands, almost 85% of the traffic signal controllers are of the vehicle-actuated type. These controllers operate in real-time by applying a control action in response to the current traffic state. A traffic-actuated controller operates based on traffic demands as registered by the actuation of vehicle and/or pedestrian detectors. There are several types of traffic-actuated controllers, but their common feature is the ability to adjust the length of the currently active phase in response to traffic flow. The green time for a phase is then a function of the traffic flow, and can be varied between pre-timed minimum and maximum lengths depending on flows. Although vehicle-actuated controllers operate in real-time, they attempt no systematic optimization. With recent advances in communication networks, computers, and sensor technologies, there is an increased interest in the development of optimizing traffic signal control systems. The literature review in Chapter 2 focuses on these new optimizing traffic signal control systems.

1.4 Outline of the thesis

In this section an outline is given of the chapters in this thesis. The relations between the chapters are illustrated in Figure 1.3.

In Chapter 2 an overview is given on the state of the art in traffic signal control. For this, information is used both from the scientific literature and from the system documentation of proprietary systems. On the basis of this literature survey a taxonomy of existing traffic signal control systems is proposed in Chapter 3. The taxonomy compares the relative strengths and weaknesses of each control system. The material presented in Chapter 3 has been reported on in [109, 110].

Subsequently, in Chapter 4 a new algorithm is defined that takes the strongest points identified in Chapter 3, and that is capable of incorporating the movement-based approach used in The Netherlands. This work has been published in [109, 111]. On the basis of this algorithm look-ahead traffic-adaptive control of a single or multiple independent intersections is made possible. Furthermore, the algorithm can be configured for use in a network configuration. When configured for use in a network the controller shares its intentions regarding its control plan with nearby intersection controllers and informs them of traffic that it plans to release. In Chapter 5 we describe how local control instruments can be made aware of the network in which they operate and how they are able to coordinate their actions. This work has partly been published in [112].

Chapter 6 summarizes the contributions of this thesis and the areas in which additional research is needed to further improve on the applicability of the developed algorithms. In Appendix A the multi-agent traffic management test bed is described that was developed during the course of this thesis. The development of this testbed has been reported on in [113–116]. The test bed was used in the development and the analysis of the algorithm and the coordination procedures described in Chapter 4 and Chapter 5.

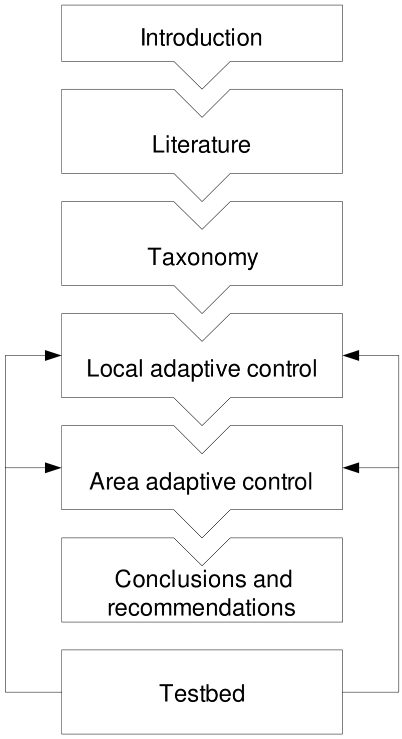


Figure 1.3: Outline of the thesis

Chapter 2

Literature survey on traffic signal control

2.1 Introduction

The installation of traffic lights at an intersection is done to solve problems in the area of traffic safety or capacity and delay. Signalized intersections permit conflicting traffic movements to proceed efficiently and safely through space that is common to those movements. This is accomplished by separating the individual movements in time rather than in space. The basic unit of a traffic control scheme is the signal group, a collection of lights that always show the same color. In most cases a signal group corresponds to a single movement on the intersection. Figure 2.1 shows the standard coding of signal groups as used in The Netherlands. A phase is a set of signal groups that can proceed concurrently without conflicts between major movements. Some movements are allowed to proceed during a phase even though they cause conflicts. Pedestrians are commonly allowed to proceed across intersections even though right-turn movements are occurring. These movements are called permitted, while protected movements are those without any conflicts.

Traffic controllers can be classified according to the method in which they allocate green time for each phase and can be roughly classified into the following types of control:

- *Fixed-time control*: A signal timing plan is selected according to a fixed schedule (e.g., time-of-day, day-of-week) from a set of predetermined plans, which were developed off-line on the basis of historical traffic data. The duration and order of all green phases remain fixed and are not adapted to fluctuations in traffic demand.
- *Actuated control*: In order to adapt the control scheme to fluctuations in traffic demand, traffic detectors are placed that indicate the presence or absence of vehicles. Using this information green phases are extended or terminated depending on the current traffic demands.
- *Adaptive control*: A traffic control system that continuously optimizes the signal plan according to the actual traffic load is called an adaptive traffic control system. Changes to the active signal plan parameters are automatically implemented in response to the current traffic demand as measured by a vehicle detection system.

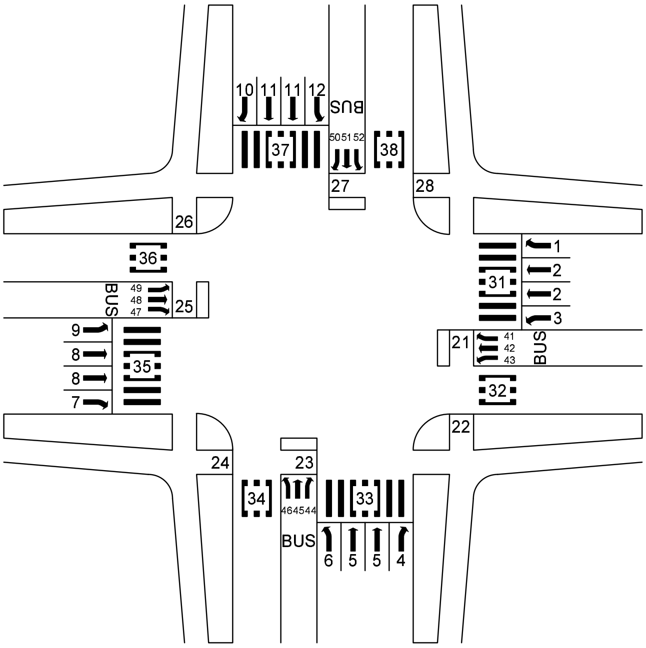


Figure 2.1: Standard coding of signal groups as used in The Netherlands

These types of control can be applied both at the local and at the network (area) level. In this chapter an overview is given of the state of the art in traffic signal control. For this, information is used both from the scientific literature and from the system documentation of proprietary systems. In Section 2.2 the basics underlying all types of intersection control are first introduced. In Section 2.3 current approaches to local intersection control are surveyed. Section 2.4 subsequently surveys current approaches to network (area) wide intersection control. Section 2.5 concludes this chapter and contains some concluding remarks.

2.2 Basics

The basic timing elements within each phase for each type of traffic control include the green, yellow and all-red interval. In order to ensure a safe traffic operation traffic signal control systems have to respect certain requirements regarding the minimum length of these intervals. Traffic signal control systems also have to take into account how these intervals affect traffic flow to also be able to realize an efficient operation. Traffic control systems differ in how they allocate green time for each phase. This will be the subject of the rest of this chapter. This section however focuses, for each of the basic timing elements, on what they have in common.

2.2.1 Green time

The green time such as it is displayed starts with a short time which is effectively unused by vehicles because the queue has to start up. Also at the end of green phase, traffic continues to enter the intersection during the first part of the yellow phase. The lag in the beginning is called the green start lag λ_s^{start}, and the time of the yellow phase that is still effectively green is called the green end lag λ_s^{end}.

The time that starts after the green start lag and ends after the end of the green end lag is called the effective green time. The requirements for guaranteed or minimum green times are normally defined for the displayed green times. However, the green time requirements for the minimum time with regard to the capacity are calculated for the effective green time. The difference in the length of the displayed green time (g_s) and the effective green time (g_s^{eff}) is equal to the difference between the green start lag and the green end lag.

$$g_s^{\text{eff}} = g_s - \lambda_s^{\text{start}} + \lambda_s^{\text{end}}$$

The green start lag is partly due to the reaction time of the first driver, but most of the time it is a consequence of the fact that vehicles have to accelerate which makes the speed of the first few vehicles lower than in the middle of the green phase. At the stop line the time headway of departing vehicles decreases from the first vehicle up to the third or fifth, because the first few vehicles are still accelerating while they pass the stop line. After a few vehicles the time headway becomes constant, h. The green start lag λ_s^{start} is the sum of the time headways at the beginning minus the constant time headway h.

$$\lambda_s^{\text{start}} = (h_1 - h) + (h_2 - h) + \dots$$

The headway h and the saturation flow q_s^{sat} (the extrapolated number of vehicles that can pass the stop line if there is a permanent queue) can be measured from direct observation of the departure times of vehicles or by counting the number of vehicles that pass the stop line in a certain time interval.

A road user who sees a traffic signal turn green will not look at the light in the first seconds after the start, because (s)he expects that the green phase will not be terminated straight away. A guaranteed green time is given as a minimum duration of the green phase. Apart from the term guaranteed green time, often the term minimum green time is used. This term has the same meaning, it is a lower limit to the green time.

The guaranteed green interval has a fixed length and should be minimal in order to prevent lost time.

2.2.2 Yellow time

The red traffic signal always means that the driver has to stop. Therefore the yellow phase has to be long enough to let all drivers pass that are too close to stop at the start of yellow. A driver can stop at an intersection if the distance at the onset of the yellow signal between the front of the vehicle and the stop line allows a driver to recognize the yellow signal, decide on the appropriate action, execute that action, and stop the vehicle in time. The decision to stop has to be made before it reaches a minimum distance d_0 from the stop line that depends on the speed of the vehicle, the driver's reaction time, and the deceleration rate of the vehicle.

The minimum distance is computed using the following equation:

$$d_0 = v\delta + \frac{v^2}{2a}$$

where:

$$\begin{aligned} v &= \text{speed of the approaching vehicle} \\ \delta &= \text{perception-reaction time of the driver} \\ a &= \text{maximum deceleration rate of the vehicle} \end{aligned}$$

A driver can clear the intersection if the distance at the onset of the yellow signal can be covered within the yellow period. The maximum distance d_c from the stop line at which the vehicle can pass the stop line prior to the red phase irregardless of acceleration is computed using the following equation.

$$d_c = v y_s - (w + l^{\text{veh}})$$

where:

$$\begin{aligned} y_s &= \text{duration of yellow interval} \\ l^{\text{veh}} &= \text{length of vehicle} \\ w &= \text{width of the intersection} \end{aligned}$$

When a vehicle needs only to enter the intersection prior to red presentation rather than cross through the intersection, as is the case in The Netherlands, the $(w+L)$-term can be safely ignored.

A successful clearing maneuver can be represented as:

$$d + (w+L) - v\delta \leq v_0(y_s - \delta) + \frac{1}{2}a(y_s - \delta)^2$$

The right-hand side of the equation represents the distance traveled from an initial speed (v) at a constant acceleration (a) during the time interval $(y_s - \delta)$ subsequent to perception-reaction time and before the onset of the red signal. This can be realized if the yellow time is chosen larger than $y_s > \delta + \frac{v}{2a}$.

2.2.3 All-red time

Traffic signals give green to conflicting movements sequentially. Between the green phase of one movement and the next green phase of a conflicting movement there should be a transition time in which care is taken that the last vehicle of the movement with the finishing green time can leave the intersection safely. It has to be assumed that vehicles continue to enter the intersection during the whole yellow phase, such that the clearance time starts at the end of the yellow phase. The all-red time is the time between the end of yellow and the start of the next green phase. It is determined by the time needed by the last vehicle of the finishing green phase to exit the conflict area and the time needed by the first vehicle of the starting green phase to reach the conflict area. The conflict area is the area that is used by both movements when traversing the intersection (see Figure 2.2 for an illustration).

The distance that has to be traveled to exit the conflict area is $l^{\text{veh}} + d_{r,s}^{\text{exit}}$, where l^{veh} is the length of the vehicle and $d_{r,s}^{\text{exit}}$ is the distance between the stop line and the farthest point of the area where the path chosen by the road user for which the green phase ends (signal

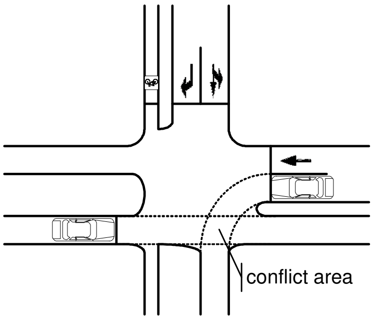


Figure 2.2: Conflict area

r) and the path of the road user for which the green phase starts (signal s) intersect. Given a speed for the vehicle (v) the exit time can be determined:

$$t_{r,s}^{\text{exit}} = \frac{l^{\text{veh}} + d_{r,s}^{\text{exit}}}{v} \tag{2.1}$$

The distance that has to be traveled to enter the conflict area is $d_{s,r}^{\text{entry}}$, which is the distance between the stop line and the nearest point of the area where the path chosen by the road user for which the green phase starts and the path chosen by the road user for which the green phase ends intersect. Given a speed for the vehicle for which the green phase starts (v) the entry time can be determined:

$$t_{s,r}^{\text{entry}} = \frac{d_{s,r}^{\text{entry}}}{v} \tag{2.2}$$

The all-red time or clearance time can subsequently be determined as follows:

$$r_{r,s} = \max\left\{0, t_{r,s}^{\text{exit}} - t_{s,r}^{\text{entry}}\right\} \tag{2.3}$$

The all-red time has to be at least 0 seconds, i.e., conflicting flows should never have simultaneous green or yellow lights. The all-red times have to be calculated for all conflicts. The numbers are often represented as a matrix, the all-red matrix. An empty entry in the matrix means that there is no conflict.

2.3 Local control

Local traffic signals are the basic signal systems, which operate under either "fixed-time", "actuated", or "adaptive" modes.

2.3.1 Fixed-time control

Under fixed-time control, the duration and the order of all green phases is fixed. Fixed-time control assumes that the traffic patterns can be predicted accurately based on historical data. Because the traffic situation changes over time, a clock is commonly used to replace one fixed-time control plan with another. As fixed-time controllers can operate without traffic detectors installed at the intersection, the construction cost is much lower than with traffic-actuated and traffic-adaptive control. The main drawback of fixed-time control is that it is not able to adapt itself as it is based on historical rather than on real-time data. Historical data is often not representative for the current situation as:

- traffic arrives at the intersection randomly, which makes it impossible to predict the traffic demand accurately.
- demand changes on the long term leading to "aging" of the optimized settings.
- demand may change due to drivers' response to the new optimized signal settings.
- Events, accidents, and other disturbances may disrupt traffic conditions in a non-predictable way.

In fixed-time control the signal cycle is divided over the various phases according to historical volumes. As a consequence of the time needed to clear the intersection when changing phases and for traffic to start-up, a fixed amount of time during the signal cycle can be considered lost, as it cannot be used for traffic flow. The amount of time lost (per hour) increases when the duration of the signal cycle is chosen shorter. Intersections with a shorter signal cycle therefore have a lower overall capacity. However, longer signal cycles also lead to longer waiting times and longer queues. In order to find an optimal value for the cycle duration and the green durations of the separate phases when the flow rates are known a formula has been derived which minimizes the average delay for all passing vehicles. This formula is presented below.

Webster's formula

Due to the fact that a theoretical calculation of delay is very complex and that direct observation of delay in the field is complicated by uncontrollable variations, Webster [120] pioneered in using computer simulation to help derive the well-known Webster's delay formula, assuming unsaturated, random arrivals for isolated, fixed-time signals. He detailed a procedure of how to calculate the optimum cycle length and green times (splits) based on minimum overall intersection delay estimated by his delay formula [120, 121]. Webster designated for each signal phase a critical lane (or critical movement) as the one with the highest ratio of flow to saturation flow (or flow ratio). Suppose that r_i is the critical flow ratio for phase p. By definition,

$$r_p = \max_{l \in L_p} \left\{ \frac{q_l}{q_l^{\text{sat}}} \right\} \tag{2.4}$$

where L_p is the set of lanes of phase p, and q_l and q_l^{sat} are the flow rate and saturation flow rate, respectively, for lane l. The optimum cycle length, C^{opt}, can be approximated as:

$$C^{\text{opt}} = \frac{1.5L + 5}{1 - R} \tag{2.5}$$

where L is the total intersection lost time per cycle and R is the summation of all critical flow ratios corresponding to each of the n phases in the cycle:

$$R = \sum_{p=1}^{n} r_p \tag{2.6}$$

For each phase p, its optimum green time, g_p^{phase}, is calculated by distributing the total available green time, i.e., $C^{\text{opt}} - L$, in proportion to its critical flow ratio:

$$g_p^{\text{phase}} = \frac{r_p}{Y}(C^{\text{opt}} - L) \tag{2.7}$$

Notice that Webster's optimum allocation of green times is based on an equal degree of saturation for all phases in the cycle.

This formula is derived for intersection control with a fixed timing. Because the various vehicle-actuated control schemes change to fixed-time control for high flow rates, this method can also be used for these intersection controls.

A second use for the Webster method is as a reference method to compare the efficiency of different intersection control methods including network control methods.

2.3.2 Traffic-actuated control

For vehicle-actuated and traffic-actuated control programs detectors are needed to get information about the actual traffic situation. The detectors that are used most frequently are inductive loop detectors. In order to decide whether it is efficient to terminate the green phase the traffic-actuated controller should be able to determine whether the last vehicle of the queue that has build up at the stop line during the red phase has passed. Measuring the gap between vehicles does this. If the gap between vehicles is larger than some maximum gap, the control program may decide to stop the green phase. Additionally many traffic-actuated controllers also extend the green to ensure that the green phase is terminated comfortably and safely.

These extensions will continue until the intervals between vehicles are long enough for the signal to decide that it would be more efficient to terminate the current green phase or until a pre-specified maximum green time has been reached.

The zones that should be monitored to ensure an efficient, safe, and comfortable termination of the green phase are depicted in Figure 2.3 along with the typical detector configuration used to monitor these zones. Zones 1 and 2 are used to determine whether it is efficient to terminate the green phase. The option zone is used to determine whether the green phase can also safely be terminated. If desired, the comfort zone can subsequently be monitored to determine whether the green phase can also be terminated comfortably for any approaching vehicles. These zones are further defined below.

Efficient termination of green

In order to maximize the green time that is fully utilized, the green time that is unused should be minimized. This is why in traffic-actuated control the last vehicle of a moving queue determines the moment at which the signal is switched from green to yellow. To determine whether the green time should be extended the controller has to ascertain whether

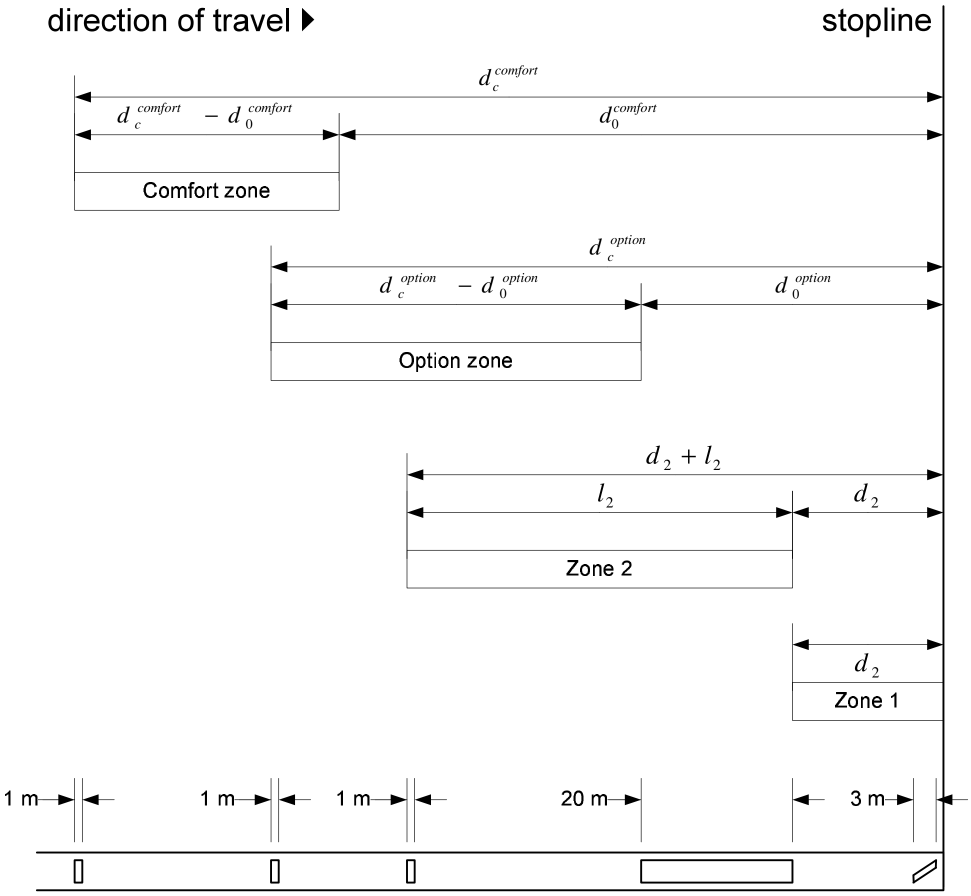


Figure 2.3: Maximum detection configuration for a vehicle-actuated controller

the distance between successive vehicles is larger than the maximum distance between two queued vehicles.

The maximum distance between two vehicles in a queue and thus the length of the zone to be monitored can be calculated as follows:

$$l_2 = t^{\text{gap}} v^{\text{max}} - l^{\text{veh}}$$

where:

$$\begin{aligned} l_2 &= \text{The length of the zone to be monitored} \\ v^{\text{max}} &= \text{The local speed limit} \\ l^{\text{veh}} &= \text{The average local vehicle length} \\ t^{\text{gap}} &= \text{The maximum gap time between two consecutive vehicles} \end{aligned}$$

To ensure that the given green time is efficiently used, the zone to be monitored should be located at a distance from the stop line which will allow the last vehicle to pass the stop line just after the effective green time has ended. The last vehicle clearing the detection zone then drives toward the stop line during yellow. This distance is determined as follows:

$$d_2 = \lambda_s^{\text{end}} v^{\text{min}} + l^{\text{veh}} \tag{2.8}$$

where:

$d_2 =$ The distance from the stop line of the zone to be monitored
$v^{\text{min}} =$ The lowest speed with which vehicles clear the intersection
$\lambda_s^{\text{end}} =$ The green end lag (the time of the yellow phase that is still effectively green)

Traffic-dependent extension of the green phase is only possible after the queue is moving. Gap measuring by detectors for a standing queue might give a premature end of the green phase, if the gap detector is just between two standing vehicles. The gap timer might then have already reached the maximum gap time at the start of the green phase. This is one of the reasons why a fixed green time is applied before gap measurement can start. The fixed green time makes it possible for the standing queue to come into motion. As the detection zone used to determine whether the queue has dissipated is located some distance from the stop line this causes a great deal of lost time.

In order to be able to guarantee that the zone just in front of the stop line is cleared a guaranteed green time is needed that can be calculated as follows:

$$g_s^{\text{min}} = \left(\frac{d_2}{l^{\text{veh}}}\right) t^{\text{gap}} - \lambda_s^{\text{end}} + \lambda_s^{\text{start}}$$

where

$g_s^{\text{min}} =$ The guaranteed green time
$\lambda_s^{\text{start}} =$ The green start lag (the time needed for a queue to come into motion)

To be safe this calculation is often done assuming longer headways. This guaranteed green time is applied even when there is only a single vehicle in front of the stop line. The green time needed by this vehicle is $1 \cdot t^{\text{gap}} - \lambda_s^{\text{end}} + \lambda_s^{\text{start}}$.

In order to limit this loss, the green time for vehicles between the stop line and the detection zone is also made traffic-dependent. The zone to monitor starts at the stop line (0) and ends at d_2. Only a limited amount of time is needed for this zone as it applies only to the first couple (i.e., 4) vehicles.

Given the following representative values [2, 122] for through traffic approaching an intersection at 80 km/h:

$$\begin{aligned} v^{\max} &= 22.2 \text{ m/s} \\ v^{\min} &= 15 \text{ m/s} \\ l^{\text{veh}} &= 5 \text{ m} \\ t^{\text{gap}} &= 2.5 \text{ s} \\ \lambda_s^{\text{start}} &= 1.5 \text{ s} \\ \lambda_s^{\text{end}} &= 1 \text{ s} \end{aligned}$$

The zone that starts at the stop line has a length of 20 meters. As the end of this zone is determined by the distance from the stop line of the second zone the second zone starts at 20 meters (d_2) and ends after 51 meters (l_2).

Gap reduction is a means of reducing the passage time or gap on the basis of the time that opposing vehicles have waited. In effect, it benefits the waiting vehicles by reducing the time allowed between vehicles arriving on the green phase before that phase is terminated. It starts with a large gap to allow the traffic to reach a constant flow. The gap time will linearly decrease in time. The longer the duration of the extension period, the smaller the gap will be. The green phase will after some time only continue when the vehicles are following each other with short intervals. There are three timing parameters associated with the gap reduction feature: Time-before-reduction, Time-to-reduce, and Minimum gap, which are described below:

- *Time-before-reduction*: This parameter determines the length of the time period after which the linear reduction of the passage time begins. The period begins when the phase is green and there is a serviceable call on a conflicting phase.
- *Time-to-reduce*: This parameter controls the rate with which the gap time is reduced. The gap time will be reduced until the minimum gap time has been reached.
- *Minimum gap*: This parameter determines the minimum time with which the green interval is extended for each vehicle actuation up to the maximum green. The gap timer starts when the vehicle actuation is removed. This extension period is subject to termination by the maximum green or a force-off.

Safe termination of green

At high speeds there is a possibility that a dilemma occurs. The dilemma situation is that drivers are too close to stop but the distance is too large for them to cross before the start of the red phase, i.e., when $d_0 > d_c$. This zone is illustrated in Figure 2.4. This situation can only occur for speeds higher than $v > 2a \cdot (y_s - \delta)$. If y_s = 4 s, a = 3.86 m/s^2 and δ = 1 s, this speed is 83 km/h, which means that only fast driving cars will encounter the dilemma situation.

When $d_c > d_0$, the driver has the option to either stop or proceed. This zone is called the option zone and is illustrated in Figure 2.5. Large option zones increase the probability for rear-end collisions. This occurs when a lead vehicle decides to stop, while the following vehicle decides to go.

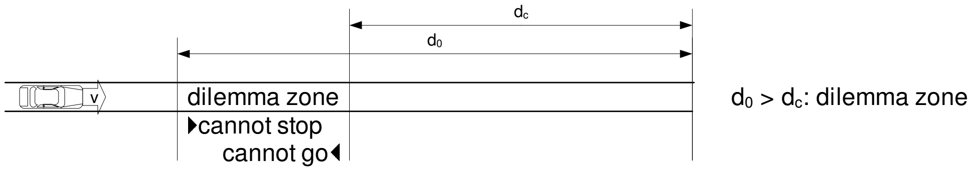


Figure 2.4: Dilemma zone

Looking in the direction of travel this zone starts at the point where the driver, if he decides to continue with a normal speed (80 km/h), exactly crosses the stop line at the end of yellow. For 4 seconds yellow, this point is located at 4 s · 22.2 m/s = 89 m. The zone ends at the cross section where the vehicle with the highest possible deceleration (5 m/s^2) is unable to stop in front of the stop line. The protection of this zone entails that the switch from green to yellow takes place when there is at most one vehicle in this zone.

Vehicles have a speed that varies between a maximum and a minimum value. For the start of the option zone calculations are done on the basis of the maximum speed (80 km/h) for the end of the dilemma zone the minimum speed is used (54 km/h).

At 54 km/h and a deceleration of 5 m/s^2 the end of the option zone is located at 34 m distance upstream from the stop line. At this speed it is unlikely that a driver decides to stop with maximum deceleration. At this lower speed it is more likely to assume for instance a slower deceleration of 3.86 m/s^2, which corresponds to the legally required deceleration in The Netherlands, instead of a maximum deceleration of 5 m/s^2. The end of the option zone is then located at 40 m distance upstream from the stop line. The option zone covers the area from 89 to 40 m upstream from the stop line. This is shown in Figure 2.5

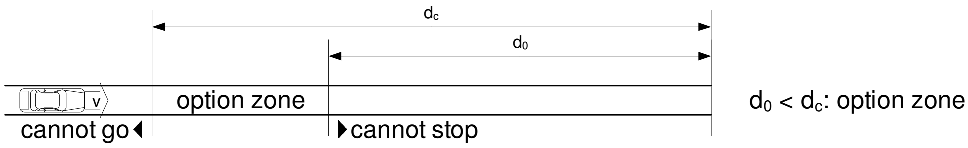


Figure 2.5: Option zone

Comfortable termination of green

Another zone, closely related to the option zone is the zone in which a driver, at the moment that yellow appears, has to decelerate stronger than what is perceived as comfortable (2.5 m/s^2), but less than what is legally required (3.86 m/s^2) to come to a full stop in front of the stop line. This is the comfort zone. The comfort zone covers the area from 115 to 80 m upstream from the stop line.

2.3.3 Traffic-adaptive control

Under traffic-adaptive operation the state of the entire intersection is taken into account in the decision to either continue the current green phase or to switch to a different phase.

This in contrast to traffic-actuated control where the decision to switch or to extend is based purely on the presence of demand on the active green phase. The traffic-adaptive control approaches described here apply solely to the level of a local intersection. The systems that will be described in Section 2.4.3 also apply to the local level, but are also able to control the intersections in a larger area. The adaptive systems described in this section are MOVA, CRONOS, and SPPORT.

MOVA

MOVA (Microprocessor Optimised Vehicle Actuation) [25] is a signal control strategy that has been developed by TRL (Transport Research Laboratory) in the mid-eighties. It was developed to overcome the problems of aging traffic-actuated signal plans. MOVA has two operational modes; the first deals with uncongested conditions, the second with situations when the junction becomes overloaded/congested with large queues on one or more approaches. MOVA determines which mode is appropriate and which approaches, if any, are overloaded.

In the uncongested mode, MOVA seeks to disperse any queue which has built up on red, and then carries out a delay-and-stops minimizing procedure every half a second. If there would be a benefit from extending the green, then the green is continued and the calculations are repeated. If no benefit is predicted, the signals change to the next stage. The delay-and-stops minimizing procedure is based on Miller's algorithm [78]. Miller's algorithm calculates (under certain simplifying assumptions) the time gains and losses caused on all approaches if the decision to switch to a different phase is postponed. If the amount gained is larger than the amount lost, the switch takes place immediately, otherwise the decision is postponed.

In the congested mode, MOVA operates a capacity-maximizing routine. This routine takes into account which approaches are overloaded, the efficiency of green use, the amount of use made of any flare lanes on the approach, and determines the signal timings that will maximize the junction throughput under the actual flow conditions prevailing.

Flare lanes are additional lanes located near the junction that are used to discharge vehicles faster than normally possible with just the main lanes. The actual benefits that a flare lane has on capacity depend on the number of extra vehicles stored on the flare lane. MOVA defines "bonus green" as the additional green time that is required if the flare lanes would not have been available. When the "bonus green" is small (i.e., little use is made of the flared area), then capacity increases as cycle time rises. When the "bonus green" is large, then capacity increases as cycle time reduces. The optimum cycle is a function of junction geometry and lost time, flows, and turning movements. MOVA continually monitors conditions during oversaturated periods and will, when appropriate, select and enforce the cycle time which maximizes capacity.

CRONOS

CRONOS (ContROl of Networks by Optimization of Switchovers) [11–13] is a real-time traffic control algorithm, that has been developed at the Centre d'Etudes et de Recherches de Toulouse (CERT). Its traffic modeling has been designed for using video measurements. For that purpose, the storing zones inside the junction and the spatial extension of the queue are also modeled.

Firstly, the traffic prediction model of CRONOS can take into account the queue spatial extension in each controlled link based on real-time image based detection and past information. In addition, it can re-actuate and memorize the left-turn vehicles stored in the intersection at each time step in order to model the departures from the links. Secondly, CRONOS applies a rolling-time horizon (80 seconds) concept and a revised Box algorithm in the system optimization process. This method is based on successive trials where the solution giving the highest performance value is modified until convergence.

CRONOS uses a stream-based approach in which the signal group (a set of signals that together control one traffic stream) is the smallest possible entity. No cycle duration nor stages are defined a priori. This approach is supposed to be more flexible than the stage-based approach because the choice of solutions is greater, but the complexity increases. The admissible set of solutions is defined by the safety constraints between signal groups and the constraints on minimum and maximum green times for each signal group. CRONOS is the only real-time operational system employing a stream-based approach.

SPPORT

SPPORT (Signal Priority Procedure for Optimization in Real Time) [33, 34] is primarily developed in response to concerns that exhaustive optimization procedures such as dynamic or linear programming may be too computationally demanding for real-time applications in networks with highly variable demands [33]. SPPORT makes signal-switching decisions using a heuristic rule-based optimization procedure. The procedure is based on the recognition that signal switches usually occur after the realization of specific discrete traffic events. By ignoring all events that have no importance for the signal operation, the procedure specifically allows for a significant reduction in the number of potential switching combinations that need to be considered to find solutions to traffic control problems. To account for the fact that different traffic events do not carry the same importance, SPPORT requires the user to prioritize the events. The higher an event is on the list, the more likely it is to receive a green phase. The program is able to pre-evaluate each of the phase sequences generated from the respective priority lists by using a predefined cost function. Also, it can dynamically select the most promising plan on-line for immediate short-term application.

As with most traffic-responsive signal control systems, the SPPORT model relies heavily on projected vehicle arrival information to make signal-switching decisions. This information is obtained from traffic detectors installed at strategic points along the approaches to the intersection under control. Each time a vehicle passes over a detector, the detection time and type of vehicle are recorded by SPPORT. This information is then used to project vehicle arrival times at the intersection stop line of every approach link. Predictions are made within SPPORT using a discrete-event microscopic simulation model that has been explicitly designed for SPPORT [24].

2.4 Area control

In a network of closely space controlled intersections, the coordination between intersections has a large influence on the performance. Vehicles departing from a queue at a traffic signal typically travel in a platoon that disperses as vehicles travel further downstream. When signal-controlled intersections are located closely together, a platoon of vehicles released from an intersection will not completely disperse before it arrives at the next intersection. The movement of a platoon of vehicles through several signalized intersections is referred to as progression. By properly coordinating the traffic signals in a network platoons of vehicles can keep progressing. Signal coordination would enhance the overall traffic operation. Signal coordination can be achieved in "fixed-time", "actuated", and "adaptive" modes.

2.4.1 Fixed-time control

Coordination between intersections can be achieved by two means. The first is time-based control. Under time-based control, the signal timing relationship is maintained by very accurate time clocks internal to each controller. The clocks in each controller are set to the same time of day. In theory, with all the controllers set to the same time of day, the offset relationship between the green indications at each successive intersection can be maintained.

The other means of achieving coordination between intersections is through the use of an interconnection. With an interconnected system, the controllers at each individual intersection (commonly referred to as the local controller) are connected to a master controller or a central computer either by a physical link, or by the use of a radio or other airways communication media. A primary function of the master controller is to ensure that the individual intersection controllers stay in sync with each other (usually by sending a synchronization pulse through the interconnection). The pulse provides a common reference point from which all the intersections can time their offsets.

Regardless of the type of mechanism used by the controller to achieve coordination, every coordinated system has a set of requirements for establishing the timing plans inside the controller. The first requirement is that all the traffic signals have to operate with the same cycle length. The intersection in the system that requires the greatest cycle length to accommodate the traffic is dominant in the design of the progression scheme. Once the system cycle length has been determined, the phase sequences and lengths (or split times) can be determined for each intersection in the system. The final signal parameter that must be determined is the offset. The offset is usually defined as the time difference between the initiation of green indications of the coordinated movements relative to the master intersection (i.e., the intersection dictates the signal timing requirements of the other intersections). The offset value is derived based upon the distance between the master intersection and the desired travel speed of traffic on the arterial.

Figure 2.6 shows a time-space diagram that shows an arterial of nine coordinated intersections. The width of the band bordered by the green line indicates the length of time available for vehicles traveling at a certain speed that allows them to continue without stopping. In this example traffic going from left to right can be seen to be favored over traffic going from right to left as the width of the green band for the latter is smaller.

Generally, there are two approaches that are employed to compute timing plans for an arterial street:

- Progression-based methods, which maximize the bandwidth of the progression, or
- Disutility-based methods, which minimize a performance measure such as the overall delay and stops.

Because these two approaches attempt to develop signal timing plans to achieve different objectives (maximize bandwidth versus minimize delay), they can result in significantly different signal timing plans for similar traffic conditions. The selection of which philosophy to use in an area is determined by local policy.

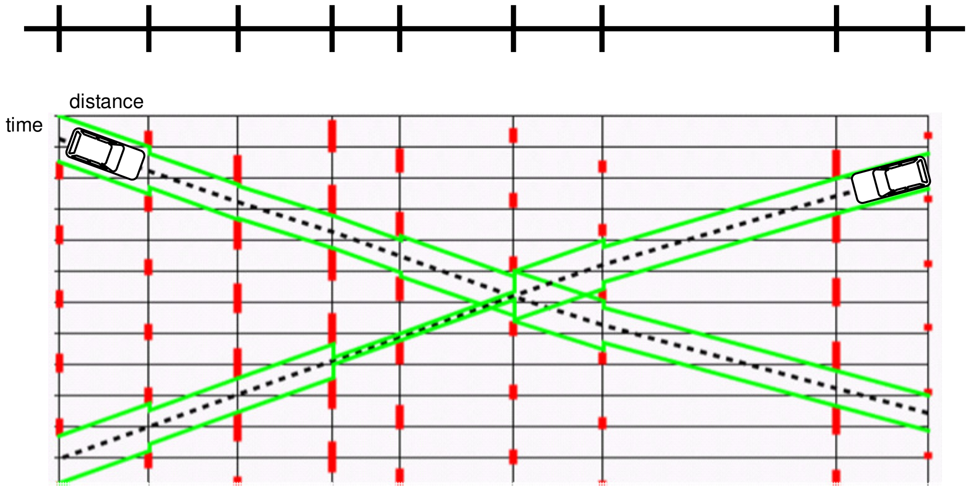


Figure 2.6: Vehicles are able to continue through the next intersection without stopping

Progression-based methods

Bandwidth optimization techniques, such as MAXBAND, PASSER II, and PASSER IV, use traffic volumes, signal spacing, and desired travel speed to determine the optimum width of the progression band that can be accommodated on an arterial. Because bandwidth optimization techniques attempt to provide the widest progression band possible, they generally result in longer cycle lengths so as to permit larger amounts of traffic to pass through an intersection during the green interval.

The first version of MAXBAND was developed by J.D.C. Little [69]. MAXBAND considers a two-way arterial and specifies the corresponding offsets so as to maximize the number of vehicles that can travel within a given speed range without stopping at any signal (green wave). Splits are considered in MAXBAND as given (in accordance with the lateral street demands); hence the problem consists in placing the known red durations of the arterial's signals so as to maximize the inbound and outbound bandwidths. For an appropriate problem formulation, it is necessary to introduce some binary decision variables, which leads to mixed-binary linear-programming problem. Little extended the basic MAXBAND method via incorporation of some cycle constraints to render it applicable also to a network of arterials [21, 69]. A number of significant extensions have been introduced in the original method in order to consider a variety of new aspects such as: time of clearance of existing queue, left-turn movements, and different bandwidths for each link of an arterial (MULTIBAND [44–46]).

PASSER (Progression Analysis and Signal System Evaluation Routine) [106, 119] was developed by the Texas Transportation Institute. The PASSER algorithm optimizes signal control in a multi-stage process. First green splits are calculated for each signal via Webster's formula, then a bandwidth optimization stage determines maximum bandwidth offsets and cycle times, and finally a fine-tuning stage is used to further reduce delay. PASSER was the first algorithm to consider more than two phases in the phase sequence and to explicitly optimize over the set of possible phase sequences. Earlier versions of PASSER were based on interference minimization. Bandwidth interference can be characterized as the difference between an actual bandwidth solution and the idealized one-way bandwidth possible in a given direction along the arterial. A one-way progression can be easily attained and is bounded by the minimum green split in the given direction. However, when synchronizing traffic signals in both the inbound and the outbound direction, some signal will likely interfere with the green band in one direction if providing an ideal progression in the other direction. The algorithm minimizes this interference by adjusting phasing sequences and offsets.

Minimizing interference is analogous to maximizing bandwidth as presented by Little, and stems from the same analytic result of half-integer synchronization. In a simplified explanation, the half-integer synchronization result concludes that two intersections with the same cycle time can be synchronized to produce the maximal equal bandwidth in both directions by either exactly synchronizing their cycles or by exactly alternating cycles, i.e., a 50% cycle time offset. Half-integer synchronization relies on the inbound and outbound directions being served simultaneously with splits of equal duration, whereas PASSER considers multiphase operation where inbound and outbound directions may be served at different times with different durations.

Disutility-based methods

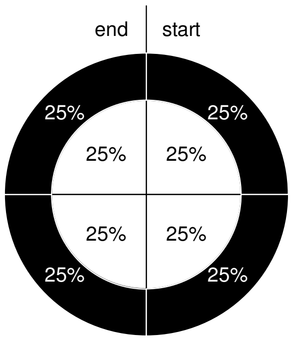
The second approach uses models to minimize the delay, the number of stops, or another measure of disutility. Examples of these types of techniques include TRANSYT-7F and SYNCHRO. These models generally attempt to find a common cycle length that minimizes the amount of overall delay in the system and then compute the offset required for progression. As a result, these optimization techniques generally produce cycle lengths that are shorter than those produced by bandwidth optimization techniques.

TRANSYT (TRAffic Network StudY Tool [50]) is a computer model to optimize the linking and timing of traffic signals in a network. TRANSYT was first developed by D.I. Robertson [97], but was substantially extended and enhanced later [50]. It is the best known and most frequently applied signal control strategy, and it is often used as a reference method to test improvements enabled by real-time strategies. TRANSYT is fed with initial signal settings such as pre-specified staging, minimum green times, and an initial choice of splits, offsets and cycle time. A unique cycle or half-cycle time is considered for all network intersections in order to enable offset coordination. A heuristic hill-climbing procedure is used to determine values for the decision variables that minimize a performance index calculated by the computer model, e.g., total number of vehicle stops.

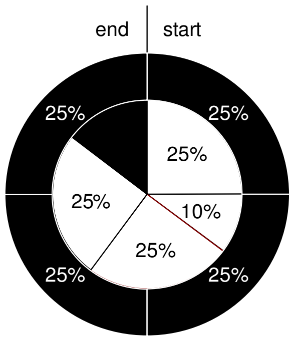
2.4.2 Traffic-actuated control

Coordination between traffic-actuated controllers is achieved on the basis of the same principles through which coordination between fixed-time controllers is achieved. In order to ensure that traffic-actuated controllers return to the coordinated phase in time a mechanism must be in place to force non-coordinated phases to terminate. Two types of force-off modes are used [104]: floating and fixed force-offs. The primary difference in these modes is in the manner the excess time from one non-coordinated phase is used by another non-coordinated phase. Typically, coordinated phases will not gap out. The non-coordinated phases can gap out if they have detectors and are operated in an actuated manner. A force-off point for each non-coordinated phase is the point in the cycle where the respective phase must terminate to ensure that the controller returns to the coordinated phase at the proper time in the cycle.

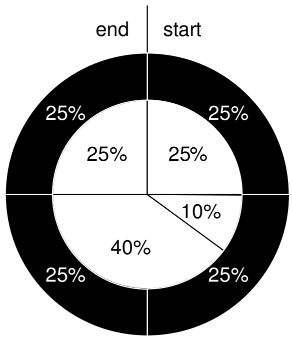
- *Floating force-off*: In floating force-off mode, the duration of the non-coordinated phases is limited to the splits that were programmed in the controller. As a consequence floating force-off does not allow for any time from phases with excess capacity to be used by a phase with excess demand. This means that phases that are allowed to start earlier as a consequence of an excess of capacity on phases earlier in the cycle will be forced to terminate before their force-off point in the cycle. This results in an early return to the coordinated phases.This is illustrated in Figure 2.7. This figure shows by means of two pie-charts how green is split over the phases in the cycle for both force-off modes. Suppose that green is split equally over all four phases (the outer ring of the pie-chart) and that the second phase needs only 10 % (the inner ring of the pie chart) of the allocated split. In that case the third phase can start earlier. In the case of a floating force-off the third and fourth phase are terminated after they have spent the 25 % of green allocated to them. This results in an early return to green to the first, coordinated phase. This is illustrated in Figure 2.7(b).
- *Fixed force-off*: Fixed force-off, on the other hand, allows the transfer of excess capacity from one phase to a subsequent phase with excess demand. This means that phases with excess demand will terminate at the force-off point irrespective of when the phase starts. The controller only allows the use of excess unused capacity and ensures that coordinated operations are not disrupted. This is illustrated in 2.7(b), where the third phase can make use of the green unused that remains after the second phase has finished until it reached its fixed-force off point.



(a) Maximum green times



(b) Floating force-off



(c) Fixed force-off

Figure 2.7: Force-off modes

Some of the advantages and disadvantages of fixed force-off are:

- Fixed force-off is useful to allow better utilization of the time available from phases operating below capacity by phases having excess demand, which varies in a cyclic manner. This is the case when phases earlier in the phasing sequence operate below capacity more often than phases later in the phasing sequence.

- Fixed force-off minimizes the early return to coordinated phases, which can be helpful in a network having closely spaced intersections. An early return to the coordinated phase at a signal can cause the platoon to start early and reach the downstream signal before the onset of the coordinated phase, resulting in poor progression.

- Fixed force-off minimizes the early return to the coordinated phase, which can be a disadvantage. Under congested conditions on the arterial, an early return can result in the queue clearance for coordinated phases. Minimizing early return to coordinated phases can cause significant disruption to coordinated operations. This disadvantage can be overcome by adjusting the splits and/or offsets at the intersection to minimize disruption.

Overall, fixed force-off has the potential to improve signal operations by better utilization of any excess capacity. However, fixed force-off will only benefit if the phases that are more likely to be below capacity are earlier in the phasing sequence. Hence, this excess time can be available to be used by a subsequent phase with a higher demand.

2.4.3 Traffic-adaptive control

Traffic load is highly dependent on parameters such as time, day, season, weather, and unpredictable situations such as accidents, special events, or construction activities. These factors are taken into account by a traffic-adaptive control system, so that bottlenecks and delays can be prevented. Adaptive traffic control systems continuously sense and monitor traffic conditions and adjust the timing of traffic signals accordingly.

Adaptive systems, like SCOOT and SCATS, have been around since the mid 70's, and have proven their worth in various places around the world. Using real-time traffic information, an adaptive system can continuously update signal timings to fit the current traffic demand. The aging of traffic signal plans, with a gradual degradation of performance as traffic patterns drift away from those in place during implementation, is well documented [4]. Many agencies have no program for monitoring the applicability of signal timing to the current traffic patterns, and it is not uncommon to find agencies that have not re-timed their signals in years. The benefits of an adaptive signal control system are apparent, since both traffic operations and staff can be made more efficient since a better performance can be gained with the same level of effort [37].

Adaptive traffic control systems are often categorized according to their generation. First-generation traffic-adaptive systems employ a library of pre-stored signal control plans, which are developed off-line on the basis of historical traffic data. Plans are selected on the basis of the time of day and the day of the week, directly by the operator, or by matching from an existing library a plan best suitable for recently measured traffic conditions. First-generation traffic-adaptive systems are often referred to as traffic-responsive signal control.

A limitation of traffic-responsive signal control is that by the time the system responds, the registered traffic conditions that triggered the response may have become obsolete. Second-generation traffic-adaptive systems therefore use an on-line strategy that implements signal timing plans based on real-time surveillance data and predicted values. The optimization process can be repeated every five minutes. However, to avoid transition disturbances, new timing plans cannot be implemented more than once every 10 minutes. Third-generation traffic-adaptive systems are similar to the second-generation systems, but differ with respect to the frequency with which the signal timing plans are revised. The third generation of control allows the parameters of the signal plans to change continuously in response to real-time measurement of traffic variables, which allows for “a-cyclic” operation.

A significant part of the literature reviewed for this section reports on benchmark studies meant to ascertain the added value of the proposed system with respect to a reference system. In most papers that report on field tests it is the current system at that time against which is benchmarked. As it is entirely conceivable that that system has not been well maintained, it is hard to judge the new system appropriately.

In cases where simulation is used as a benchmark environment it is easier to benchmark against an optimized system. However, also in those cases a bias is conceivable as the available expertise on and the effort put into the optimization of the existing system might not have been on par with the expertise available as well as the effort put into the optimization of the proposed system. Simulation is furthermore only partly capable to represent reality. For example, pedestrian activity and side-street parking are often not modeled accurately.

Benchmarks are furthermore predominantly performed against fixed-time or actuated systems and not against other adaptive systems. Generally speaking, given time-varying unpredictable demand patterns, a traffic-adaptive system should be able to outperform a fixed-time or actuated system. The margin of improvement demonstrated by a traffic-adaptive system over a fixed-time or traffic-actuated system cannot be compared easily to that determined for another adaptive system as it is strongly related to the network geometry and traffic demand chosen in the benchmark study. For a fair comparison the systems should be benchmarked using the same test environment and an equal amount of effort should be put in the optimization of the different systems by people that are knowledgeable.

As so little comparative benchmarks between traffic-adaptive systems are available care was taken not to judge these systems with respect to one another. The systems described in this section are systems from the, proven, second generation (SCATS, SCOOT, MOTION) and from the, younger, third generation (OPAC, PRODYN, RHODES, UTOPIA/SPOT, TUC).

SCATS

SCATS (Sydney Coordinated Adaptive Traffic System) [71] was developed in the early 1970’s by the Roads and Traffic Authority of New South Wales, Australia. The system utilizes a distributed, three-level, hierarchical system employing a central computer, regional computers, and local intelligent controllers to perform a large-scale network control. The regional computer can execute adaptive control strategies without any aid from the central computer, which only monitors the system performance and equipment status. The control structure enables SCATS to expand easily and suitably for controlling any size of traffic area.

SCATS employs a strategic optimization algorithm and a tactical control technique to perform system-wide optimization. The optimization philosophy contains four major modules: (1) cycle length optimizer, (2) split optimizer, (3) internal offset optimizer, and (4) linking offset optimizer.

SCATS selects combinations of cycle, splits and offset from predetermined sets of parameters with few on-line calculations. Maximum freedom consistent with good coordination is given to local controllers to act in the traffic-actuated mode. The system is designed to automatically calibrate itself on the basis of data received, minimizing the need for manual calibration and adjustment.

For control purposes, the total system is divided into a large number of comparatively small subsystems varying from one to ten intersections. As far as possible, the subsystems are chosen so that they can be run independently for many traffic conditions. For each subsystem, minimum, maximum, and geometrically optimum cycle lengths are specified. To coordinate larger groups of signals, subsystems can link together to form larger systems, operating on a common cycle length. Linking plans manage the linking between subsystems. When a number of subsystems are linked together, the cycle time becomes that of the linked subsystem with the longest cycle time. The combination of subsystem plans, link plans between subsystems, variable cycle length, and variation of offsets provides an infinite number of operating plans.

Four background plans are also stored in the database for each subsystem. The cycle length and the appropriate plan are selected independently of each other to meet the traffic demand. For this purpose, a number of detectors in the subsystem area are defined as strategic detectors; these are stop-line detectors at key intersections. Various system factors are calculated from the strategic detector data, which are used to decide whether the current cycle and plan should remain the same or be changed.

Strategic options, minimum delay, minimum stops, or maximum throughput can be selected for the operation. These options can be permanent or dynamically changed at threshold levels of traffic activity.

Four modes of operation are included in SCATS:

- Masterlink Operation — This is the normal mode of operation which provides integrated traffic-responsive operation. There are two levels of control in this mode: strategic and tactical. The strategic control determines the best signal timings for the areas and subareas, based on average prevailing traffic conditions. The tactical control is concerned with the control of the individual intersections within the constraints imposed by the strategic control. This lower-level control deals with termination of the unnecessary green phases when the demand is below the average. The basic traffic measurement used by SCATS for strategic control is a measure analogous to the degree of saturation on each approach. This measure is used to determine cycle length, splits, and the direction and magnitude of offset.

- Flexilink Operation — In the event of failure of a regional computer or loss of communications, the local controllers can revert to a form of time-based coordination. In this mode, adjacent signals are synchronized by reference to the power mains frequency or an accurate clock, and signal timing plans are selected by time-of-day. The local controller operates under a vehicle-actuated or a fixed-time control system.

- Isolated Operation — In this mode, the controller operates under independent vehicle actuation or a fixed-time control system.

- Flash Operation — This is a manual mode in which normal automatic operation is overrided. It incorporates flashing yellow display for the major approaches and flashing red display for the minor approaches.

SCATS has also been widely used in several cities in Australia, New Zealand, USA, China, Singapore, Philippines, and Ireland.

SCOOT

SCOOT (Split, Cycle, and Offset Optimization Technique) [17, 55, 98, 99] was initiated by the British Transport and Road Research Laboratory (TRRL) in the 1970's, with its first commercial system installed in 1980. SCOOT is a centralized system based on a traffic model with an optimization algorithm adapted for on-line application. Optimization takes place by incrementally updating a fixed-time plan. The benefit of this approach is that changes are gradual. The transition is less disruptive and less prone to overreacting than the transition between distinct plans as is typical in a time-of-day scheme.

SCOOT performs optimization at three levels: Split, Cycle and Offset. SCOOT measures vehicles with a detector at the upstream location of the stop line. SCOOT predicts the profile of arrivals to the intersection based on the updated flow information collected by the upstream detectors. This arrival profile is compared with the departure profile, and the differences represent those vehicles delayed and queued at the intersection. These flow profiles are estimated for each cycle from a combination of the vehicles approaching, the time to clear the queue, the impact of offset and split adjustment.

The split optimizer in SCOOT evaluates the projected arrival and departure profiles every second. A few seconds before each change of signals, the system adds the delay from all movements that will end or begin at that change of signals. This delay is compared against the delay calculated with the change of signals occurring either a few seconds earlier or later. Then, the optimized scenario that provides the best balance of movement delay will be implemented.

The offset optimizer operates on each node pair and searches for the best-offset timing to improve traffic progression on the basis of the cyclic profile. Based on the profile measured in the previous cycle, the offset optimizer minimized the delay for all movements of the intersection by incrementing or decrementing the current offset with a few seconds. With the offset optimizer in the SCOOT systems, green waves can be imposed along the coordinated signal controlled corridor. After this offset adjustment, the split optimizer may further adjust the signal timings based on traffic actually approaching the stop line at that time.

The cycle optimizer looks at the saturation levels of all intersection movements once each cycle-control period. At critical intersections with low reserve capacity, the cycle optimizer will extend the length of the cycle. It does so in different increments of time (e.g., 4, 8, or 16 seconds) depending on the current cycle length. If an intersection is operating below capacity, the cycle optimizer will reduce the length of the cycle.

SCOOT has been widely used in several cities in UK, USA, Canada, China, South Africa, Cyprus, Pakistan, United Arab Emirates, Chile, and Spain.

MOTION

MOTION (Method for Optimization of Traffic signals In Online-controlled Network) [8, 18] is a traffic signal control strategy developed by Siemens, Germany. The system operates on three functional levels: on the strategic level, every 5, 10 or 15 minutes (cycle time, average green time distribution, basic stage sequences and network coordination); on the tactical level, every 60 to 90 seconds (cycle, current stage sequence); and on the operational level, every second (green time modification).

Starting with the dominant traffic stream through the network, a grid of green waves is constructed, taking into account modeled (or if available, measured) platoons in the links. For each intersection, the optimum sequence of stages is identified, and the basic split of green times is fixed. Depending on the remaining spare time per intersection, and on the constraints of the optimized offsets, a certain amount of bandwidth is available for the subsequent local optimization. Optimization normally aims at minimizing delays and stops in the network. In the final step the decision is made to change the signal programs at the intersections. To avoid frequent minor changes, changes are only implemented if calculation determines a significant improvement in the overall optimization objective. Depending on the type of local controller and on the local control method used, the signal programs are then converted and implemented. To avoid severe disruptions in traffic flow due to the plan switch, a smooth (gliding) transition from the running to the new plan is performed. Until the next optimization run of the network model, the local controllers operate on their own and modify their plan according to the local situation, but always staying within the given bandwidth.

TUC

TUC (Traffic-responsive Urban Control) [9, 31, 32, 89, 90] employs a store-and-forward based approach to road traffic control, which introduces a model simplification that enables the mathematical description of the traffic flow process without the use of discrete variables. This opens the way to the application of a number of highly efficient optimization and control methods (such as linear programming, quadratic programming, nonlinear programming, and multi-variable regulators), which, in turn, allows for coordinated control of large-scale networks in real-time, even under saturated traffic conditions.

The critical simplification is introduced when modeling the outflow of a stream suggests that there is a continuous (uninterrupted) outflow from each network link (as long as there is sufficient demand). The consequences of this simplification are:

- The time step t of the discrete-time representation cannot be shorter than the cycle time C, hence real-time decisions cannot be taken more frequently than at every cycle.
- The oscillations of vehicle queues in the links due to green/red-commutations are not described by the model.
- The effect of offset for consecutive intersections cannot be described by the model.

Despite these consequences, the appropriate use of store-and-forward models may lead to efficient coordinated control strategies for large-scale networks.The three main modules of TUC are the split, cycle, and offset control modules that allow for real-time control of green times, cycle times and offset.

The basic methodology employed for split control by TUC is the formulation of the urban traffic control problem as a Linear-Quadratic (LQ) optimal control problem based on a store-and-forward type of mathematical modeling. The control objective is to minimize the risk of oversaturation and queue spill-back, and this is achieved through the appropriate manipulation of the green splits at signalized junctions for given cycle times and offsets.

Longer cycle times typically increase the capacity of the junction as the proportion of the lost time caused by switching signals becomes accordingly smaller. Longer cycle times may however increase vehicle delays at undersaturated junctions with longer waiting times during the red phase. The objective of cycle control is to increase the capacities of the junctions as much as necessary to limit the maximum observed saturation level in the network. Within TUC this objective is achieved through the application of a simple feedback-algorithm that uses the maximum observed saturation levels of a pre-specified percentage of the network links as the criterion for the increase or decrease of the cycle.

Offset control is achieved through the application of a decentralized feedback control law that modifies the offsets of the main stages of successive junctions along arterials, so as to create "green waves" when possible, taking into account the possible existence of vehicle queues. To implement a new offset in TUC, a transient cycle time is temporarily implemented at all but the first junction along an arterial. The transient cycle time is implemented one single time, after which all the junctions along the arterial are coordinated according to the new offset.

UTOPIA/SPOT

UTOPIA/SPOT (Urban Traffic Optimization by Integrated Automation/Signal Progression Optimization Technology) [74, 75] is a traffic signal control strategy developed by Mizar Automazione in Turin, Italy. UTOPIA/SPOT calculates optimal control strategies for subareas of the network, with each subarea having the same cycle length. While operating, the system maintains a historical database of measured flows, turning percentages, saturation flows, and cycles in use.

The system utilizes a distributed, two-level, hierarchical system employing a central, area-level computer, and intersection, local-level computers to perform large-scale network control.

SPOT is a fully distributed, traffic-adaptive signal control system. It operates by performing a minimization of local factors such as delays, stops, excess capacities of links, stops by public or special vehicles, and pedestrian waiting times. With each repetition, all SPOT units exchange information on the traffic state and preferential policies with their neighboring SPOT units. This permits the application of look-ahead (each SPOT unit receives realistic arrival predictions from upstream intersections) and strong interaction (each controller considers, in the local optimization, the adverse effects that it could have on downstream intersections). Data is exchanged with neighboring intersections every few seconds.

As each SPOT-unit communicates with surrounding units, the system can be programmed to prioritize public transport and emergency vehicles by giving early warning of these vehicles or by allowing them to be quickly cleared through the intersection. SPOT can also prioritize traffic on the basis of adherence to timetables, number of passengers, etc.

SPOT allows a staged system implementation over time starting with a few intersections. It can be implemented without a central computer for small systems of typically six intersections or less. However, for larger intersection networks, the UTOPIA central PC-based control system should be added.

At the area level, the UTOPIA-module provides a mechanism to handle critical situations in the form of two actions that a signal controller may request of adjacent signal controllers. Thus, a controller may cope with congestion by requesting that a downstream signal increase throughput or that an upstream controller decrease demand. These requests are realized by respectively relaxing or tightening green time constraints . For the area level, UTOPIA, the model can: (1) analyze area-wide traffic data and make predictions for main street flows over time, (2) apply its internal macroscopic model to entire area network and traffic counts, and (3) optimize the total travel time with constraints of average speed and saturation flows.

SPOT has been used in several cities in Italy, The Netherlands, USA, Sweden, Norway, Finland, Denmark, and the UK.

OPAC

The OPAC (Optimized Policies for Adaptive Control) algorithm [43, 47–49] has gone through several development cycles ranging from OPAC I through OPAC-VFC. OPAC maintains the specified phase order. For uncongested networks, OPAC uses a local level of control at the intersection to determine the phase on-line, and a network level of control for synchronization, which is provided either by fixed-time plans (obtained off-line), or a "virtual fixed cycle". A virtual fixed cycle is a cycle that although fixed between intersections (to enable synchronization), is determined on-line (hence virtual). Predictions are based on detectors located approximately 10-15 seconds upstream. After the initial 10-15 seconds, a model predicts traffic patterns (typically 60 seconds).

OPAC breaks the signal optimization problem into subproblems using dynamic programming, an approach that leads to a more efficient computation. At the same time it determines a virtual cycle. These are implemented for a time-step (roll period) of about 2-5 seconds. The length of the virtual cycle is varied according to the needs of either the critical intersection or the majority of intersections. The virtual cycle is allowed to change by typically one second per cycle. Within this limitation, OPAC provides local coordination by considering flows into and out of an intersection in selecting its offset and phase lengths.

The congestion control process in OPAC generally attempts to maximize throughput, by selecting the phase that will allow the maximum number of vehicles to pass the intersection. OPAC does this by considering saturation flows and space available to store vehicles on each link. The first step of congestion control involves determining the next phase given that there is not a critical link that is on the verge of or currently experiencing spill-back. On the basis of these calculations, the algorithm determines whether it is necessary to revisit the timings at neighboring intersections in light of throughput constraints that their physical queues impose on each other's effective service rates.

OPAC-I assumes an infinite horizon and uses dynamic programming to optimize the performance index. OPAC-I cannot be implemented on-line in real-time because of the extensive time required to compute the optimal settings. OPAC II used an optimal sequential constraint search (OSCO) to calculate the total delay for all possible phase switching options. The optimal solution was determined as the phase switching that produces the lowest total delay values, and OPAC-II was found to derive solutions with performance indexes within 10% of those generated with OPAC-I. Although OPAC-II was faster than OPAC-I, it still suffered from the need for vehicle arrival information for the entire planning stage, which was 50-100 seconds in length. OPAC-III was the first version of OPAC that featured the rolling horizon approach and was developed at first for a simple two-phase intersection, but later extended to an eight-phase intersection, which allowed phase skipping. OPAC-VFC added the algorithm used to coordinate adjacent signals.

PRODYN

PRODYN (Programmation Dynamique) [3, 36, 53, 54] is a real-time traffic control algorithm, which has been developed by the Centre d'Etudes et de Recherches de Toulouse (CERT), France. PRODYN evolves from two stages of development: two-level hierarchical control (PRODYN-H) and then decentralized control (PRODYN-D). The former offers the best result; however, its applicability is restricted due to the complex computations involved and the network size (limited to about 10 intersections). The latter, on the other hand, alleviates those limitations. Two approaches have been studied for PRODYN-D: no exchange (PRODYN-D1) versus exchange (PRODYN-D2) of information between the intersections.

At the intersection level, the optimization model's aim is to minimize delay by using improved forward dynamic programming with constraints on maximum and minimum greens. At the network level, the network coordination optimization is performed by a decentralized control structure. The procedure includes: (1) simulating a specific intersection output for each time step as soon as the intersection controller finishes its optimization over the time horizon, (2) sending the simulation output to each downstream intersection controller, and (3) using the output message from upstream controllers at the next time step to forecast arrivals.

RHODES

RHODES (Real-Time, Hierarchical, Optimized, Distributed, and Effective System) [79] is a hierarchical control system that uses predictive optimization, allowing intersection and network levels of control. RHODES includes a main controller, a platoon simulator (APRES-NET [30, 52]), a section optimizer (REALBAND [29]), an individual vehicle simulator (PREDICT [30, 52]), and a local optimizer (COP [101]). RHODES requires upstream detectors for each approach to the intersections in the network. RHODES also can use stop-line detectors to calibrate saturation flow rates and to improve traffic queue estimates. RHODES is entirely based on dynamic programming, and it formulates a strategy that makes phase switching decisions based on vehicle arrival data.

The design of RHODES is based on dividing the traffic control problem into subproblems by use of a network hierarchy. The subproblems include the network-loading problem, the network flow control problem, and the intersection control problem.

At the top of the hierarchy is the network-loading problem. At this level, link loads and the prediction of the trends in the change of loads from real-time data are estimated. RHODES uses this information pro-actively to predict future platoon sizes near the boundaries of the system.

The middle level consists of the network flow problem and involves the selection of signal timing to optimize the overall flow of vehicles in the network. The decisions are made in this level every 200-300 seconds. A platoon prediction logic model called REALBAND is used at this level. Network optimization is also established at this level and its results are used as constraints for the decision made in the next level.

The lowest level of the control strategy is the one at the intersection and it is responsible for making the final second-by-second decisions regarding traffic signal operation. This level uses two sublevels of logic. The first is the Link Flow Prediction Logic which uses data from detectors on the approach of each upstream intersection, together with information on the traffic state and planned phase timings for the upstream intersection, to estimate vehicle arrivals at the intersection being optimized. The other level is the Controlled Optimization of Phases (COP), which uses the information from the network flow problem, in addition to the results from the link prediction logic, to determine whether the current phase should be extended or terminated.

2.5 Conclusions

In this chapter an overview was given of current approaches to traffic signal control. The continuing growth in computational power enables control systems to further cater to the dynamics of the traffic system. Traffic-adaptive systems are currently the most advanced and complex systems available. Whereas the working of fixed-time and traffic-actuated control systems is generally well-understood and is more-or-less standardized, this is not yet the case for traffic-adaptive systems. As traffic-adaptive systems operate on the forefront of what computers, monitoring equipment, traffic prediction, and optimization are capable of, these systems significantly differ in their approach of traffic-adaptive control. In the next chapter a taxonomy of traffic-adaptive systems will be created comparing the different adaptive control systems with respect to the design choices made in the development of each system.

Chapter 3

A Taxonomy of Look-Ahead Traffic-Adaptive Control Approaches

Traffic signal control essentially comes down to making the right decisions at the right time. As such the traffic signal control problem solved by all traffic-adaptive systems can be formulated in the form of a general decision problem. This general decision problem in turn can be represented as a simple decision tree such as the one shown in Figure 3.1.

The root of the decision tree represents the current state $s_i \in S$, where i is the current time step and S is the set of all states. States in the decision tree of Figure 3.1 are represented by open circles and decisions by solid circles. The cost involved in order to switch to the subsequent state, s_{i+1} when deciding for an decision u_i is denoted by c_i. All traffic-adaptive systems support a generalized cost function that can linearly combine stops, delay, and travel time. Figure 3.1 depicts two possible successor states for each decision, illustrating

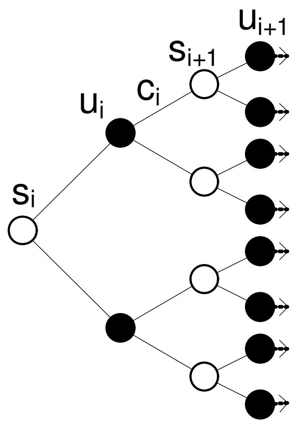


Figure 3.1: Decision tree

the fact that the result of implementing control decision u_i might not be deterministic.

In general, the nodes of a search tree represent decisions. These decisions are mutually exclusive and therefore partition the search space into two or more simpler subproblems. At each time step, the system observes the system's current state s_i, and selects a control decision, $u_i \in U_i$, where u_i is the decision and U_i is the finite set of decisions available to a controller in state s_i. When the controller chooses an decision $u_i \in U_i$, the cost incurred by taking that decision, is denoted by c_i. After the implementation of decision u_i the system has switched to state s_j with probability $p_{i,j}(u_i)$. The objective of each traffic-adaptive system is to find an optimal sequence of decisions.

The objective of the system is to operate such that the total cost over the entire planning horizon is minimized. Thus, the task of the system is to obtain a sequence of control decisions $(u_0, u_1, \ldots u_K)$, also referred to as a policy or control trajectory, such that the expected cost over a finite horizon of length K is minimized. In the case of an infinite planning horizon, a discount factor $\gamma < 1$ is typically applied to future costs to obtain a finite estimate of the *cost-to-go* from the current state i, denoted by $f(i)$. The optimal cost-to-go value, denoted by $f^*(i)$, is a function of the immediate cost of implementing the control decision plus the expected cost-to-go from the subsequent state, a relationship encapsulated in the following recursive expression which is also known as Bellman's Equation [35]:

$$f^*(i) = \min_{u \in U(i)} \left\{ c_i(u) + \gamma \sum_{j \in S} p_{i,j}(u) f^*(j) \right\} \tag{3.1}$$

Computational complexity unfortunately still prevents the configuration of a traffic-adaptive system in which no compromises have to be made. As traffic-adaptive systems operate on the forefront of what computers, monitoring equipment, traffic prediction, and optimization are capable of, these systems significantly differ in their approach. There are many different ways in which a traffic-adaptive system can be configured in order to end up with a workable system that is (a) able to come up with good signal timings and (b) able to deliver them on time. In this chapter a taxonomy of traffic-adaptive systems will be created comparing the different adaptive control systems on the design choices made while creating the system.

Looking at the various traffic-adaptive systems we can discern the following features in which they differ:

- the *architecture*: Does the system control intersections from a central location or can the system partially be distributed to the intersections?
- the *search algorithm*: Is the optimal sequence of decisions found by using move-based search method, or by using a constructive search method?
- the different *decisions* (u_i) considered in the optimization: Is the order in which phases can be given green to predetermined or can this be determined (and optimized) on-line?
- the *prediction models*: How is the performance (c_i) of each decision (u_i) evaluated? How accurate is the model used in optimizing the signal timings? Is a fast vertical queuing model used instead of a slow but possibly more accurate simulation model?
- the length and resolution of the *planning horizon* over which an optimal sequence of decisions is sought (i.e., the depth of the decision tree): Is the length of the horizon

fixed (e.g., 2 minutes) or dependent on current traffic conditions? Is the resolution static (e.g., is the horizon divided into 5 seconds intervals) or is it dynamic (e.g., dependent on projected arrival times)?

- the *update frequency*: How often can the optimization be done (i.e every 0.5 seconds or every 5 seconds)?

The following sections elaborate on each of these features and on how each traffic-adaptive system (i.e., in order of their earliest reference found in literature SCOOT [98], SCATS [71], PRODYN [54], OPAC [43], UTOPIA-SPOT [74], RHODES [52], and ALLONS-D [92], TUC [31]) differs in the way these features are filled in.

3.1 Architecture

The architectural approaches to the traffic-adaptive control problem can roughly be divided into centralized and distributed approaches. Centralized systems are those that rely on a central computer to make control decisions and direct the actions of individual controllers whereas distributed systems on the other hand are those that have the intersection controller to be responsible for operation decisions. As each approach has its specific strengths and weaknesses in practice often a hybrid, hierarchical approach is used.

3.1.1 Centralized systems

Centralized systems are those that use a central computer to make control decisions and to direct the actions of individual controllers. Each intersection requires only a standard controller and interfacing unit and does not perform any optimization.

Centralized systems depend on reliable communications networks. Because real-time control commands are transmitted from the central computer to the local intersection, any interruption in the communications network forces the local controller to operate without that real-time control and revert to its fall-back plan. In traditional centralized systems, the fall-back operation is usually stand-alone traffic control under actuated operation.

In more recent systems, time-based coordination is incorporated into the fall-back operation, but this still requires a transition period for the system to switch from central to local control mode. During this short period of time, signal coordination is usually lost. For this reason, the state of the art in centralized systems usually employ some form of land-based communication infrastructure and preferably operate under their own communication network.

Centralized systems depend on reliable central computers. When the central computer is down the coordination between intersections will be lost. The impact of this would be worse than in case of an often localized communication failure, as the problem will affect all coordinated signal-controlled intersections. In order to tackle this concern, some of the centralized systems employ two identical central computers, containing the same software and share a joint operation system by networking together with a high-speed network connection. Under this arrangement, it would be possible for one computer to continue to operate the system if the other fails.

Centralized systems are often not easily expanded. Many traditional centralized systems are designed around a maximum network size. Increasing the size of the network requires a significant investment in central computer upgrades and often, upgrades in the software as well.

Centralized systems allow centralized control algorithms. This is the one area where centrally controlled systems have a distinct advantage over distributed systems. Some control algorithms require a central computer to calculate the optimization algorithm for the entire network. Only a centrally controlled system can provide this capability. Examples of systems that use a centralized architecture are SCOOT, SCATS, and TUC.

3.1.2 Distributed systems

In distributed systems the intersection controllers are responsible for operation decisions. These systems rely on powerful local controllers; as the power is built into the local controller, these controllers must have all the features desired for signal control at the intersection.

Distributed systems do not transmit mandatory real-time control commands over the communications network. Consequently, the intended operation of the system can be maintained even during communications breakdown and central computer downtime. True distributed systems incorporate this characteristic more effectively than centralized systems with a time-base backup.

Distributed systems are usually operating under time-based coordination, with the central computer and communications network used only to synchronize the internal clock. Distributed systems are relatively easy to expand. Each time a new traffic signal controlled intersection is built, the computing capacity of the system, which is stored in the local controllers, is expanded to incorporate the new intersection.

Distributed systems do not provide for centralized adaptive-control algorithms, and would be inapplicable for some adaptive-control algorithms that require centralized control to optimize the traffic operation. An example of a system that uses a distributed architecture is PRODYN. Although PRODYN originally used a hierarchical architecture [54] later versions use a decentralized architecture. In [3, 36] the decentralized version of PRODYN is described. In this version adjacent intersections share information regarding pending arrivals. Preference was given to the latter version in subsequent developments [36]. This approach is also taken by the SPOT-units of UTOPIA-SPOT.

3.1.3 Hierarchical systems

Hierarchical systems combine the principles of centralized and distributed systems. The central computer and local controllers perform specific calculations and optimize the traffic operation based on different objective functions. Examples of hierarchical systems are UTOPIA-SPOT, RHODES, and OPAC-VFC. At the local level intersections are respectively controlled by the SPOT, COP [101], and OPAC look-ahead adaptive control algorithms. At the network level UTOPIA-SPOT uses the UTOPIA model to further extend the optimization horizon of each local controller. RHODES takes a different approach and uses the REALBAND algorithm [29] to optimize the movement of observed platoons in the network. The REALBAND decisions are used as constraints to the intersection control logic. The network level of OPAC is not publicly documented.

3.2 Search algorithms

As the decision space has a tree-like structure, the search for the optimal sequence of decisions corresponds to building the tree. An exhaustive search of the entire decision space results in a full tree being built. Since search space size grows exponentially with problem size, it is not possible to explore all assignments except for the smallest problems. The only way out is to not look at the whole search space. Efficiency in searching the decision space is considered by the degree to which the entire tree will not have to be built to find an optimal solution. Finding an optimal solution basically comes down to assigning values to the decision variables and determining the cost of this assignment. Search methods now differ in the way in which these assignments are visited. We can classify search methods according to different criteria:

- *Complete versus incomplete exploration*: Complete search means that the search space is investigated in such a way that all solutions are guaranteed to be found. This is necessary when the optimal solution is needed (one has to prove that no better solution exists). Incomplete search may be sufficient when just some solution or a relatively good solution is needed.

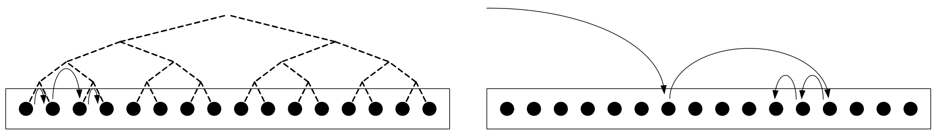
- *Constructive versus move-based*: This indicates whether the method advances by incrementally constructing assignments, i.e., control decisions (thereby reasoning about partial assignments which represent subsets of the search space), or by moving between total assignments, i.e., control policies (usually by modifying previously explored assignments).

- *Randomness*: Some methods have a random element while others follow fixed rules.

Move-based methods are usually incomplete. This is not surprising given typical sizes of search spaces. A complete exploration of a huge search space is only possible if large subspaces can be excluded a priori, and this is only possible with constructive methods which allow one to reason about whole classes of similar assignments. Moreover, a complete search method must remember which parts of the search space have already been visited. This can only be implemented with acceptable memory requirements if there is a simple structuring of the space that allows compact encoding of subspaces.

Figure 3.2 shows two ways to search a search space with N (here 16) possible total assignments, some of which are solutions. The constructive search methods usually organize the search space by partitioning it systematically. This can be done naturally with a search tree (Figure 3.2(a)). The nodes in this tree represent decisions that partition the remaining search space into two or more (usually disjoint) subspaces. Using such a tree structure, the search space can be traversed systematically and completely (with as little as $O(N)$ memory requirements). Figure 3.2(a) shows a sample tree search, namely a depth-first incomplete traversal. As opposed to that, Figure 3.2(b) shows an example of an incomplete move-based search, which does not follow a fixed search space structure. Of course, it will have to take other precautions to avoid looping and to ensure termination.

From a technical point of view, the main difference between constructive and move-based search is that constructive search extends the control policy while local search changes it. During constructive search assignments are made when going down the tree, and this is undone in reverse order when backing up the tree to a parent node. It is characteristic of



(a) A structured tree search (b) A move-based search

Figure 3.2: Different approaches to not having to build the entire search tree

move-based search that a move produces a small change, but it is not clear what effect this will have on the quality of the solution.

3.2.1 Move-based search

A move-based or local search algorithm starts from a candidate solution and then iteratively moves to a neighbor solution. This is why move-based search algorithms are predominantly used by systems that adhere to the concepts of cycle, splits, and offsets, because common practices exist to determine acceptable initial values (see for instance [2, 107]). Typically, every candidate solution has more than one neighbor solution; the choice of which one to move to is taken using only information about the solutions in the neighborhood of the current one, hence the name local search.

Termination of local search can be based on a time bound. Another common choice is to terminate when the best solution found by the algorithm has not been improved for a given number of steps. Local search algorithms are typically incomplete algorithms, as the search may stop even if the best solution found by the algorithm is not optimal. This can happen even if termination is due to the impossibility of improving the solution, as the optimal solution can lie far from the neighborhood of the solutions crossed by the algorithms.

In the literature, e.g., in [22], local search methods are often characterized by the nested-loop program scheme of Algorithm 3.1.

Algorithm 3.1 Local Search

```
set candidate solution
while global_condition do
    while local_condition do
        select a move
        if acceptable then
            do the move
            if new optimum then
                remember it
            end if
        end if
    end whileset new candidate solution
end while
```

The algorithm starts from an initial solution and iteratively updates this solution until it no longer significantly (specified as a local condition by the user) improves. In order to prevent that the algorithm gets stuck in a local optimum this process is repeated for different initial solutions until enough (specified as a global condition) parts of the search space have been searched.

We discuss two examples of local search methods that follow this scheme that are predominantly used in traffic signal control: hill climbing, and genetic algorithms.

- *Hill climbing*: A hill-climbing (or valley-descent) algorithm starts with a candidate solution, specified by the user, selected by the program using a fixed criterion, or selected randomly. Traffic-adaptive systems often start out with the currently implemented solution. Then, the algorithm selects a variable to be optimized (i.e., offset, cycle length, etc.) and creates two additional candidate solutions for this variable, one by increasing the values of that variable and the other by decreasing the value. Initially, the value of the selected variable is increased or decreased by a specified amount called the step size. Following this, the algorithm uses a traffic simulator to calculate the fitness values for each of the two new scenarios and compares them with the base scenario. These evaluations identify the two best scenarios and, consequently, a direction of further search. For instance, if increasing the value of the selected variable resulted in a better fitness value, the search algorithm will mark this new scenario as the current best and continue searching in the direction of increasing values for the variable. In the next iteration, the search algorithm generates a new scenario by increasing or decreasing the value of the selected variable in the selected search direction, calculating the new fitness value, and comparing it with the two current best values. The algorithm continues in this manner until the fitness value for the new scenario ceases to be better than the current best. Hill-climbing methods guarantee optimal solutions only when the function to be optimized is unimodal[1](has one peak or valley). For multi-modal functions, the hill-climbing method may terminate with a suboptimal solution depending on how good the candidate solution is.

- *Genetic Algorithms*: Genetic algorithms maintain a pool of solutions rather than just one. New candidate solutions are generated not only by "mutation", but also by "combination" of two solutions from the pool. Probabilistic criteria are used to select the candidates for mutation or combination, and for discarding excess solutions from the pool. Genetic algorithms belong to a class of algorithms known as evolutionary algorithms which have been developed fairly recently. A genetic algorithm starts with a subset of scenarios (some members of a population) and applies principles of natural selection (mating, gene mutation, etc.) to generate a new or revised set of scenarios (called the next generation). A genetic-algorithm-based optimization model uses a specified traffic simulation model to evaluate the fitness of each member (i.e., a signal timing scenario) in the current population. Then, it generates a new population by combining the characteristics of (that is, by mating) selected pairs of scenarios (members). The principles of natural selection ensure that the characteristics of the fittest members (i.e., those with higher bandwidths or lowest delays, depending on the objective of optimization) have a high probability of transmission to the next generation. A genetic algorithm terminates when either no more improvements occur, or a certain number of user-specified generations are complete, whichever occurs first.

[1]a function $f(x)$ between two ordered sets is unimodal if for some value m (the mode), it is monotonically increasing for $x \leq m$ and monotonically decreasing for $x \geq m$. In that case, the maximum value of $f(x)$ is $f(m)$ and there are no other local maxima.

In TRANSYT the optimization can be done both by using a hill-climbing algorithm and a genetic algorithm. In [50] the genetic algorithm is said to be mathematically better suited for determining the absolute or global optimal solution, compared to hill-climbing optimization. However, it is also said that the genetic algorithm generally requires longer program running times on the computer, compared to hill-climbing optimization. This is why the SCOOT system, which is essentially an on-line version of the TRANSYT-model, still uses a form of hill climbing, making a series of frequent small adjustments to signal timings to minimize the chosen performance index throughout the network. SCATS also uses this method of optimization. As to date none of the applications of genetic algorithms as found in literature [50, 91, 105, 119] have been implemented in a system that performs on-line optimization.

SSPORT [24] can also be said to use a move-based search method. However the method employed is strictly unique in the sense that is has been specifically targeted for application in the optimization of traffic signals. It is based on the recognition that signal switches occur after the realization of specific discrete events such as after a queue of vehicles has reached a certain size, after a queue has just finished dissipating, or after the detection of an incoming transit vehicle. By ignoring all events that have no importance for the signal operation, the number of potential switching combinations that need to be considered can be significantly reduced.

To account for the fact that different traffic events do not carry the same importance, SSPORT assigns a priority or weighting to each event. While the use of prioritized lists of events allows the SPPORT model to determine the relative importance of various traffic events, it is often difficult to determine beforehand which event should have the highest priority. To solve the above problem, the user is permitted to provide more than one prioritized list of events for consideration by the model. When more than one list is provided, the signal optimization algorithm generates a candidate timing plan for each list and then selects for implementation the one yielding the best performance.

3.2.2 Constructive search

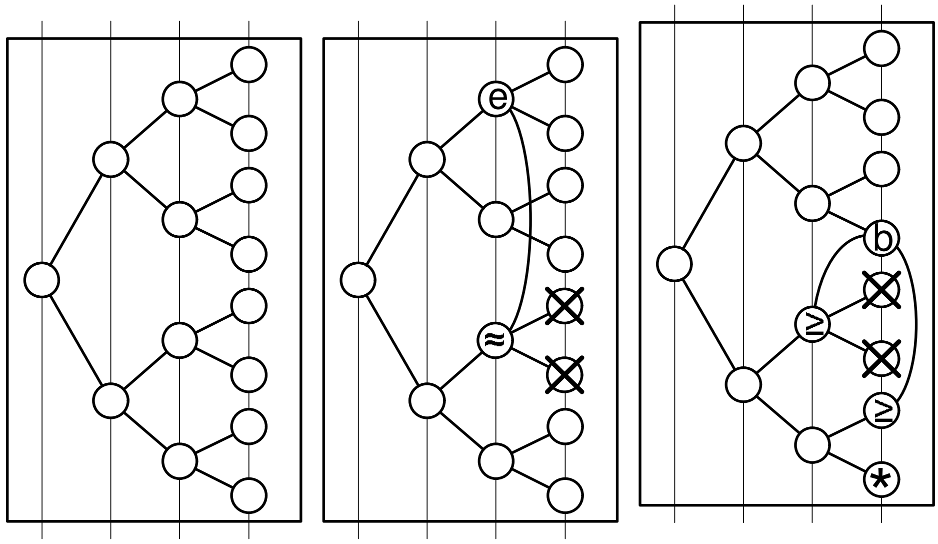
Dynamic programming and branch-and-bound (and combinations thereof) are the constructive search techniques that are predominantly used in traffic-adaptive systems.

Dynamic programming

Dynamic programming is a method for solving problems exhibiting the properties of overlapping subproblems and optimal substructure, described below:

- *Overlapping subproblem*: A problem is said to have overlapping subproblems if the problem can be broken down into subproblems that are reused several times. For example, the problem of computing the Fibonacci sequence exhibits overlapping subproblems. The problem of computing the n-th Fibonacci number, $F(n)$, can be broken down into the subproblems of computing $F(n-1)$ and $F(n-2)$, and then adding the two. The subproblem of computing $F(n-1)$ can itself be broken down into a subproblem that involves computing $F(n-2)$. Therefore the computation of $F(n-2)$ is reused, and the Fibonacci sequence thus exhibits overlapping subproblems.

- *Optimal substructure*: A problem is said to have an optimal substructure if its optimal solution can be constructed efficiently from optimal solutions to its subproblems.



(a) decision tree (b) equivalent states (c) bounding

Figure 3.3: Different approaches to not having to build the entire search tree

> For example the solution of the n-th Fibonacci number, $F(n)$, can be constructed efficiently from the solution of subproblems $F(n-1)$ and $F(n-2)$ by adding the two. Typically, a greedy algorithm is used to solve a problem with optimal substructure wherever such an algorithm can be found; otherwise, providing the problem exhibits overlapping subproblems as well, dynamic programming is used. If there are no greedy algorithms or overlapping subproblems, often a straightforward search of the solution space is the best possible solution.

The applicability of the approach depends on the opportunities for state aggregation within the decision tree. The strength of dynamic programming is that it can prevent that optimal solutions to subproblems it has already solved are recomputed. A naive approach may waste time recomputing optimal solutions to subproblems it has already solved. In order to avoid this, the solutions to already solved problems are saved. Then already-computed solutions can be retrieved and used. This approach is called memoization (not memorization, although this term also fits). It is however only possible to reuse a previous solution when states and thus the corresponding subproblems can be considered equal. RHODES, PRODYN, and OPAC all employ dynamic programming as their method of optimization.

In order to attain greater efficiency OPAC, RHODES and PRODYN use an approximate state equivalence relation. OPAC reduces the state space by constraining the number of phase changes or switches allowed during a planning horizon to at least one and no more than three. COP reduces the state space by indexing the states by their time and the number of phase switches evaluated and only keeping the sequence of phase switch time points that minimized the performance index until that time point and pruning the rest. In PRODYN the state space is reduced by limiting the maximum queue length (a value of 14 was chosen). Furthermore an equivalence class is defined, whereby the state space is drastically reduced.

This is accomplished by partitioning the set of possible queue lengths into intervals such as $[0,1);[1,3);[3,6);[6,10);[10,14]$. In comparing two states, each 4-tuple of queue lengths is mapped into a 4-tuple of corresponding queue length intervals. If two states are mapped into the same 4-tuple of queue intervals, they are deemed equivalent.

Figure 3.3(b) shows how dynamic programming assists in not having to build the entire search tree as depicted in Figure 3.3(a). It shows that the optimization algorithm recognizes that the state marked "$\approx$" is equivalent to the state marked "e" and is able to prune the evaluation of the control decisions leading to the states marked "X" as a result.

Branch-and-bound

Branch-and-bound is a general method for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization. It belongs to the class of implicit enumeration methods.

A branch-and-bound procedure requires two tools. The first one is a smart way of splitting the decision space into several smaller feasible subspaces. This is called branching, since the procedure is repeated recursively to each of the subspaces and all produced subspaces naturally form a tree structure, called search tree or branch-and-bound-tree. Its nodes are the constructed subspaces.

Another tool is bounding, which is a fast way of finding upper and lower bounds for the optimal solution within a feasible subspace. The core of the approach is a simple observation that (for a minimization task) if the lower bound for a subspace A from the search tree is greater than the upper bound for any other (previously examined) subspace B, then A may be safely discarded from the search. This step is called pruning.

Branch-and-bound is usually implemented by maintaining a global variable that records the minimum upper bound seen among all subregions examined so far; any node whose lower bound is greater than this upper bound can be discarded. It may happen that the upper bound for a node matches its lower bound; that value is then the minimum of the function within the corresponding subspace. In both these cases it is said that the node is solved. Note that this node may still be pruned as the algorithm progresses.

Figure 3.3(c) shows how branch-and-bound can assist in not having to build the entire search tree depicted in Figure 3.3(a). It shows that the optimization algorithm recognizes that the cost of reaching the state marked "$\geq$" exceeds the current upper bound and is able to prune the evaluation of the control decisions leading to the states marked "X" as a result. The upper bound was determined by the algorithm after having evaluated the then optimal solution marked "b". The actual optimal solution is marked "*". Of the adaptive systems reviewed only ALLONS-D and SPOT employ the branch-and-bound method in its pure form.

In order to obtain a tight upper bound an initial path must be established through the search tree for which it is most likely to obtain a good solution. This involves that initially parts of the search space that are unlikely to contain good solutions are ignored. This is done by using *heuristics*. Heuristics are used to explore promising areas of the search tree first. This can be done by using problem specific knowledge (often borrowed from current practices in tuning traffic responsive and vehicle-actuated controllers) or by reusing information gained from previous optimizations. Each algorithm published in the literature is believed to apply heuristics to further accelerate the process of finding the lowest cost control trajectory. However, in the literature no specific references could be found regarding the heuristics applied by each algorithm. Generally the heuristics that can be applied fall into two categories: heuristics that are based on the efficiency of green usage and heuristics that are based on the reuse of prior information. These categories of heuristics are described below:

- *Efficiency of green usage*: Vehicle-actuated controllers keep the active phase green as long as it is expected to be saturated. As a search heuristic, if the controller expects to be able to disperse traffic at the saturation flow rate, then it will extend the phase unless the current phase has been served for the maximum duration and switching is the only feasible option.

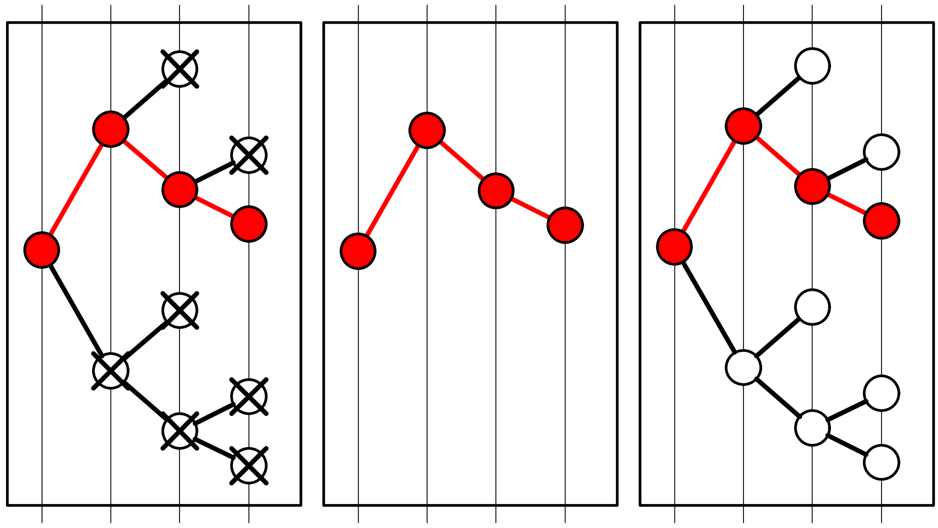
  When the active phase is no longer serving traffic at the saturation flow rate another heuristic can be used to determine whether the phase should be extended. This heuristic was first proposed in [78]. The heuristic begins by determining (a) the savings in delay for the vehicles that would be served if the active phase were extended another time step, and (b) the delay incurred to currently queued or arriving vehicles that are not served if the active phase is extended. If the approximated delay saved by extending the current green phase outweighs the extra delay imposed on opposing traffic, then the decision to extend the phase is considered first.

- *Reuse of prior information*: This heuristic uses information retained from the previous decision-making effort and applies it to improve decision-making in the subsequent look-ahead. Depending on the accuracy of the prediction model used in the look-ahead optimization this can considerably speedup the optimization process. However, when there are significant changes between update intervals, the use of prior information can send the optimization process into the wrong branch of the search tree, thus leading to a decrease in speed. In [102] Shelby compared two different versions of information reuse: (1) one in which only the optimal trajectory of the previous optimization is used to guide the branching process, and (2) one in which all trajectories of the previous optimization are kept and used to guide the branching process. The information reused for both versions of information reuse is illustrated in Figure 3.4. Figure 3.4(a) show the decision tree of the previous optimization. In this figure the states that are not crossed out denote the optimal path through the tree. Figure 3.4(b) shows the information retained if only the optimal path is retained for reuse, whereas Figure 3.4(c) shows the information retained if all paths are retained for future use.

  A drawback of this heuristic is that, depending on the chosen version of the heuristic, a large amount of memory must be retained. Also, the computational effort required to search the tree in such a manner must outweigh the computational effort of evaluating each decision.

3.3 Decision variables

The width of the tree to be searched is dependent on the number of decisions that can be made at each point in time. The decision variables for systems that adhere to cyclic operation such as TRANSYT, SCOOT, and SCATS, have to decide upon optimal cycle times, splits, and offsets. While a cyclic operation facilitates the analysis of traffic behavior around controlled intersections by ensuring the repetitiveness of signal timings over time, it



(a) previous (b) trajectory (c) decision tree

Figure 3.4: Information retained from previous optimizations

may not provide the necessary flexibility to respond to large unexpected changes in traffic demands or to efficiently accommodate transit priority requests.

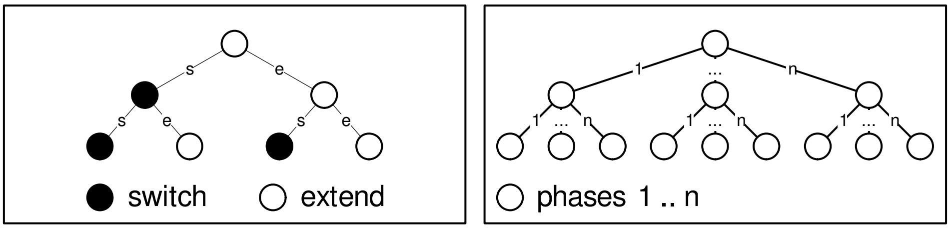
For a-cyclic systems, the terms cycle, splits, offsets have no meaning, because the system decides every time to switch or not to switch to the next phase. Consequently, the control is not to determine the optimal cycle, splits, or offset, but to solve the best control sequence for adapting a very short-term demand variation. For a-cyclic systems the decision in its simplest form comes down to deciding whether to extend the current phase or to switch to the next phase. This is the approach taken in OPAC, ALLONS-D, PRODYN, and SPPORT. Although this approach significantly reduces the number of options to consider, it does not allow arbitrary phase sequencing.

In its most elaborate form the choice available is that between phases. This approach allows the arbitrary sequencing of phases but comes at a cost in the width of the search tree. This is the approach chosen by UTOPIA-SPOT. Both of these approaches are shown in Figure 3.5.

A compromise between these two extremes is found in allowing phase skipping. When the skipping of phases is allowed, any phase sequence can be attained. This is shown in Figure 3.6. This is the approach taken by RHODES. The downside of this approach is that when the initial phase sequence is chosen wrongly the gain in width is counteracted with an increase in tree depth.

3.4 Prediction models

For pro-active traffic control it is important to predict vehicle arrivals, turning rates, and queues at intersections, in order to optimize phase timings that optimize a given measure of effectiveness (e.g., average delay).



(a) extending the phase (b) choosing the phase

Figure 3.5: Different approaches regarding decisions considered

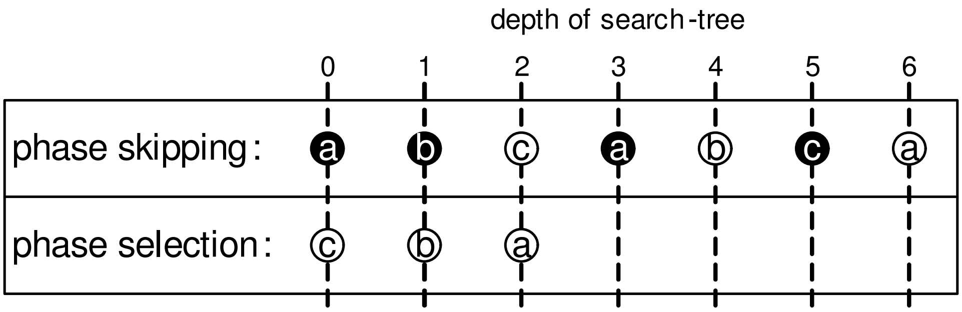


Figure 3.6: Phase skipping versus selection

3.4.1 Arrival models

In order to predict imminent vehicle arrivals, vehicle detectors are placed upstream of the controlled intersection. Some systems also take the arrivals into account that are expected to be released from an upstream intersection. Various models are used to predict the time needed by a vehicle to arrive at the controlled intersection. Section 3.4.1 describes how imminent arrivals are determined by the different traffic-adaptive systems. Subsequently, in Section 3.4.1 the approaches are described on the basis of which expected arrivals are determined.

Imminent arrivals

For PRODYN, vehicle detectors are placed for each link at 50 and 200 meters upstream from the stop line. The detectors at the 200 meter location are used to forecast future vehicle arrivals at the downstream intersection. The detectors at the 50 meter location are used for queue estimation, using a vertical queuing model. Additionally, a detector may be placed at the stop line. Arrivals to the downstream stop line are expected after a constant travel time has elapsed.

The ideal detector location for OPAC is about 8-12 seconds upstream of the stop line on each lane. In case of left turn or right turn pockets a count detector should be placed as far upstream as possible.

For SPPORT, SCOOT, and UTOPIA-SPOT, detectors are ideally placed just downstream of the upstream intersection. The detectors should be positioned in such a way that free flow conditions can be achieved over them, i.e., the mean speed of vehicles over the detector should be equal to the link mean speed within +/- 20%. This cannot be achieved if the detector is located too close to the upstream junction. Additional detectors located closer to the stop line or just downstream of a bus stop are used to provide updated traffic information. SCOOT uses a specified cruise time to predict when vehicular flows recorded in the profile are likely to reach a downstream stop line. A flow dispersion model is used to spread out platoons proportionally to the distance between signals. In SPPORT [33] arrival times are predicted by a model that uses a discrete-event microscopic simulation model explicitly designed to operate with the SPPORT model [24]. The simulator has the ability to model the effects that transit vehicles have on the general traffic when they stop in the right of way to board and discharge passengers [33].

Expected arrivals

The SCOOT system uses the detectors placed just downstream of the upstream intersection to generate a cyclic flow profile. Newly detected flows are weighted against flows from previous cycles to avoid unduly large fluctuations in the profiles. In all cases, the cyclic profiles are based on the cycle time of the upstream intersection. For links without detectors, the logic estimates a synthetic cyclic flow profile for the link from the cyclic flow profile of the upstream link and the state of the controlling signals. The synthetic cyclic flow profile consists of two parts: the main flow estimated from the discharge pattern from the main upstream feeder link, and a remainder calculated from the typical proportion of traffic entering the synthetic link from the other upstream links. Where there is no suitable upstream link to act as a proxy link, i.e., in the case of a side-road entry link, logic allows the user to input an estimated value for the flow based on street measurements [98].

UTOPIA-SPOT, PRODYN, and RHODES [52] use the output of the detectors on the approach of each upstream intersection, together with information on the traffic state and planned phase timings for the upstream signals, to predict future arrivals at the downstream intersection. This approach allows a longer prediction time horizon since the travel distance to the intersection is longer. A benefit of this approach is that it includes the effects of the upstream traffic signals in the intersection control optimization problem. In the original implementation of PRODYN, future arrivals were provided by an upper layer that simulated the dispersion of vehicles at each signal in the network. Later the original hierarchical approach was abandoned in favor of a purely decentralized approach, because the computational complexity of the original hierarchical approach could not be scaled to larger networks. Future discharge from the upstream signal is estimated in the purely decentralized approach by the upstream signal and communicated directly to the downstream signal.

3.4.2 Queuing models

Models explicitly capturing queues at intersections can be categorized into two groups: those based on vertical or point based queue representations and those based on horizontal or spatial queue representations. The former ignores the spatial aspects of queues, whereas the latter recognizes the spatial and temporal aspects of queues. Various methods have been proposed to capture the spatial and temporal aspects of queues, such as those based on kinematic wave theory [68, 95], shock wave analysis [126], and those based on the cumulative vehicle counts [66]. The formation of queues at a specific location will result in the propagation of queues in the upstream direction when the traffic signal is in a red phase. Models with a spatial queue representation have two important properties: (a) the number of vehicles that can be stored on a road segment is bounded by the jam density of the road (vehicles unable to enter the segment because of the standing queue would be stored in the upstream segment); and (b) in the queuing dissipation process, the head of the queue behaves like a moving bottleneck moving in the upstream direction while the tail of the queue may continue to grow depending on the arrival rate of vehicles joining the queue. The queuing process and the vehicle movements can be displayed on a time-space diagram as shown in Figure 3.7. In this figure the different traffic conditions that a vehicle can encounter when approaching and traversing a controlled intersection are denoted A–D. In zone A vehicles approach the intersection at free flow speed. Vehicles in zone C and D are queued and their corresponding trajectories are thus horizontal. Vehicles in zone E and B accelerate to free flow speed after having been queued or constrained by vehicle that have not yet been able to accelerate to free flow speed.

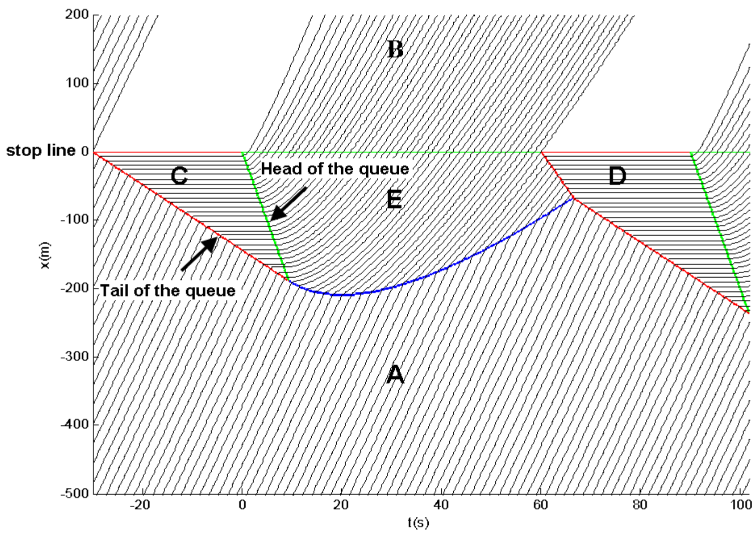


Figure 3.7: The queuing process at a controlled intersection

Most systems (e.g., RHODES, SPOT, OPAC, PRODYN) employ a vertical queuing model since they require the least computational effort. To avoid spill-back SPOT "translates" the state of the vertical queue into the state of the horizontal queue [94].

3.4.3 Turning rates

An assumption for all systems is that some estimates for turning rates at the intersection are given. These rates are not deterministic; they change over time. In order to update the turning rates the following information should be available: (1) a prior estimate whose uncertainty is modeled with a normal distribution with known mean and variance, (2) the turning percentages that are measured at any given time, and (3) the error distributions for these measurements.

Turning rates can be defined on the basis of this information using one of the following methods

- Information minimization/entropy maximization [51, 76, 117, 118]
- Bayesian [5, 72], and
- Maximum likelihood [5, 86]

In [5] it is argued that the Bayesian method is the most appropriate choice of model for the estimation of turning flows at intersections as the other two models involve a non-deterministic number of iterations based on error tolerance whereas the Bayesian method consists of exactly seven iterations. This method is also used by RHODES [79].

A prior estimate can be estimated from approach and departure volumes for each leg of the intersection on the basis of the method described in the Highway Capacity Manual [107]. In this method each approach to the intersection is considered an origin. Each departure leg is considered a destination. The problem then becomes one of estimating the origin-destination (O-D) table given the entering and exiting volume on each leg of the intersection.

3.5 Planning horizon

Traffic-adaptive systems employ a traffic model to evaluate alternative traffic signal timings over a planning horizon. The length of the planning horizon as well as how the horizon is split up into successive intervals differs between each adaptive system. Typically however the horizon has a fixed length (of typically 1 to 2 minutes) and is subdivided into fixed intervals. From their descriptions we can deduce that OPAC, PRODYN, SPPORT, and ALLONS-D all use or have used 5-second intervals. The approach where both the length of the planning horizon and the length of the time intervals in which it is subdivided are variable is not applied by any of the algorithms reviewed.

The planning horizon provides the real-time traffic-adaptive signal-timing control logic with the ability to plan future signal-timing decisions. If the planning horizon is short, perhaps several seconds, then the signal-timing decisions are restricted. For example, if the planning is made for a 10-second horizon, the signal-timing logic can only make timing decisions that extend or shorten the current phase. On the other hand, if the planning is done for a longer horizon, the signal-timing decisions can include decisions on phase sequencing and phase duration.

It may seem contradictory that a look-ahead planning algorithm could be more myopic (i.e., short-sighted) than traditional fixed-time or coordinated-actuated control. This is caused by the fact that the optimization algorithms are often unable to consider a full cycle in their planning horizon due to limitations in real-time computational abilities, whereas off-line methods plan for a full cycle of traffic, which may be in excess of 120 seconds.

In the context of a single, isolated intersection, the drawback of being unable to plan at least a full cycle in advance manifests in multi-phase configurations where the saturation flow rate is slower on one or more phases (i.e., left-turn phases) than the rate for other phases (i.e., through-movement phases). Presented with a finite planning horizon, a look-ahead search algorithm must decide whether to completely serve a left-turn queue dispersing at a slow rate, or truncate service of the left-turn phase to service large through-phase queues at a higher rate. In general, look-ahead search algorithms tend to neglect left-turn queues, often serving only the minimum green. Newell [85] reported similar drawbacks of the rolling horizon look-ahead search technique on intersections where saturation flow rates are not distinct, but traffic flow is asymmetric (e.g., heavy traffic in only one cross-street direction versus heavy traffic in both main street directions). However, in the context of longer planning horizons, residual queues on under-serviced phases accumulate delay long enough so that the decision to preempt their service is no longer the optimal short-term decision.

In the context of a single, isolated intersection, the drawback of being unable to plan at least a full cycle in advance manifests itself in the difficulty for adjacent signals to stabilize into compatible cycles, since the look-ahead search algorithm is unable to plan long enough to recognize the cyclic nature of platooned arrivals on each approach.

To counteract the suboptimal tendencies of rolling horizon with an insufficient horizon [85] some algorithms have introduced terminal costs in order to penalize residual queues at the end of the horizon [102]. The purpose of the terminal cost is to counteract a bias that could lead the signal optimization process to select signal switching decisions that yield a low cost in the near future but a high cost thereafter. SSPORT, SPOT, OPAC and PRODYN all incorporate terminal costs in their optimization.

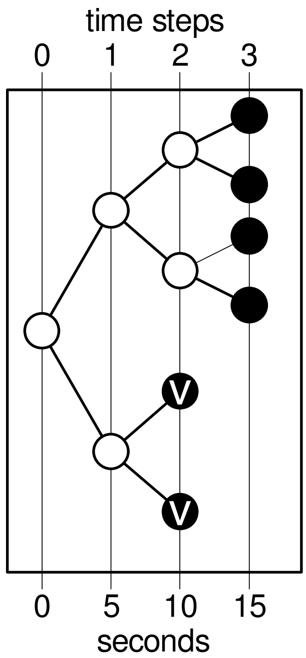
The ALLONS-D algorithm takes a different approach wherein the length of the horizon depends on the current traffic situation. The ALLONS-D algorithms enlarges the horizon until it finds a solution in which all projected arrivals are cleared. Although the idea of a horizon that shrinks or grows dependent on the traffic situation sounds attractive, it might not turn out this way in the case of the ALLONS-D algorithm. In saturated conditions - with many projected arrivals - the length of the horizon might become so large that the optimization method used by ALLONS-D might be unable to come up with an answer in time.

The RHODES system takes yet another approach. It uses the concept of historical delay which penalizes each individual vehicle proportionally to the average duration that the vehicle has been queued prior to the optimization horizon, until that vehicle is served. Historical delay thus contributes to the objective during all time intervals in the horizon.

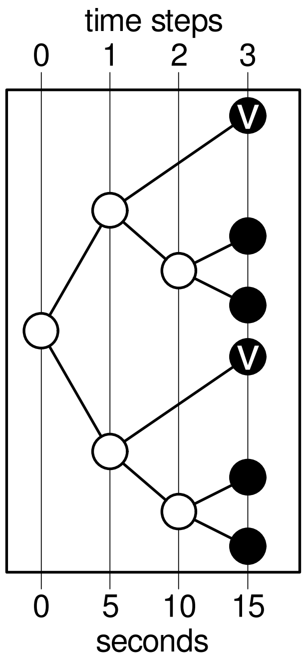
Figure 3.8 summarizes the different approaches. Figure 3.8(a) illustrates a decision space with a fixed interval and with a variable length horizon (black nodes marked "v" denote states at which the search was ended). Figure 3.8(b) illustrates a decision space with a variable interval (black nodes marked "v" in this case denote states that have been reached by implementing a decision with a larger interval). Finally, in Figure 3.8(c) a decision space is illustrated in which both the time interval and the length of the planning horizon is variable.

3.6 Update frequency

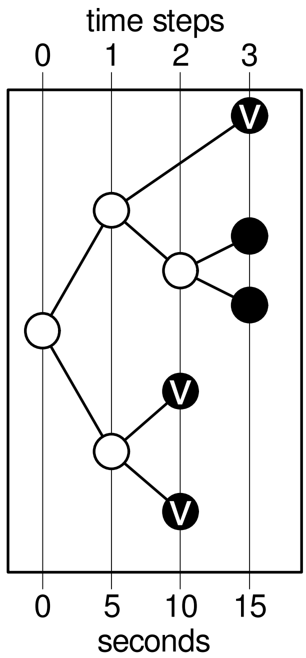
Traffic-adaptive systems rely on predicted arrivals. As the distance over which these arrivals are predicted increases, the reliability of these predictions often decreases. This is why a rolling horizon is often applied. The concept of a rolling horizon originated in operations research [7] and is used to determine a short term policy based on a longer term analysis. All adaptive systems reviewed that depend on arrival predictions employ the concept of a



(a) Variable horizon with a fixed interval



(b) Fixed horizon with a variable interval



(c) Variable horizon with a variable interval

Figure 3.8: Different approaches regarding the length of the planning horizon

rolling horizon. These algorithms implement only the first (few) decision(s) of the control plan after which a new optimization is performed.

The rolling horizon concept is visualized in Figure 3.9, where each horizontal bar denotes the calculated control plan for a decision horizon. The system commits to this control plan until the optimization process refreshes it. The amount of time that passes between each subsequent optimization is called the commitment period.

The commitment period is, for all adaptive systems reviewed, equal to the length of the intervals which subdivide the planning horizon. For most adaptive systems reviewed the length of this interval is typically fixed to 5 seconds. Waiting 5 seconds between decisions to switch or extend the current phase can however have a significant impact on delay.

Consider, for example, the case where a queue dissipates earlier than predicted. With a 5-second commitment period, an adaptive system may take up to 5 seconds to realize the error, resulting in the waste of green time. With a 1-second decision resolution, controllers can more quickly terminate phases as queues clear out, reallocating this time or capacity to phases that do have traffic to serve.

Note that, as all adaptive systems choose their commitment period equal to the length of the interval in which the planning horizon is subdivided, switching from a 5-second to a 1-second decision resolution increases the number of time steps in the planning horizon by a multiple of 5. This imposes too much of an increase in computational effort for many algorithms to solve in real-time. Thus, the typical trade-off is to also decrease the duration

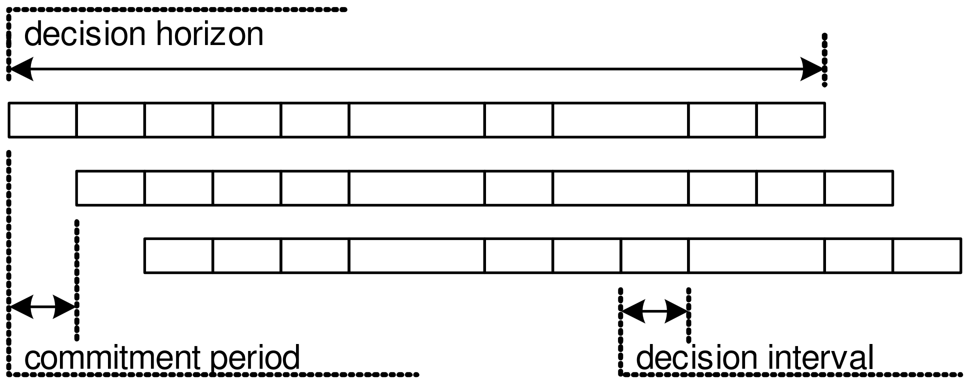


Figure 3.9: Rolling horizon

of the planning horizon.

3.7 Conclusions

There are many different ways to configure a traffic-adaptive system. Although the core of each of the traffic-adaptive systems reviewed is based on the idea of finding a short term policy on the basis of a long-term analysis, they differ with respect to their architecture, the search algorithm applied, the decision variables, the prediction model used, the length and resolution of the planning horizon, and the update frequency.

Unfortunately computational complexity still prevents the configuration of a traffic-adaptive system in which no compromises have to be made in order to end up with a workable system that is able (a) to come up with good signal timings and (b) to deliver them on time.

As the base performance of an adaptive system is at least as good as that of an actuated controller there are considerable advantages to the deployment of an adaptive system. However, in order to gain the full advantage of traffic-adaptive control, the system should be carefully tuned. Computational complexity, geometry of an intersection, and demand patterns should be considered.

Chapter 4

Local adaptive control

In this chapter a new algorithm is presented for the look-ahead adaptive control of an intersection. In Section 4.1 the choice for look-ahead traffic-adaptive control is motivated as well as the need to create a new algorithm. The new algorithm is presented in Section 4.2. In Section 4.3 the results from a comparison of the look-ahead adaptive controller with a traffic-actuated controller are presented. This chapter ends with Section 4.4 in which conclusions are drawn and recommendations are made for future research.

4.1 Motivation

A traffic-actuated controller decides to either extend the active green phase or to switch to the next phase based on whether vehicles are still present on the approaches of the active green phase. A traffic-actuated controller can be considered to suffer from tunnel vision as it does not consider traffic on the other approaches. Traffic-adaptive control differs from traffic-actuated control because it can evaluate a set of feasible control decisions and choose a decision that is optimal with respect to its current objectives. A *look-ahead* traffic-adaptive controller additionally is capable of determining the optimal control decision on the basis of a longer term analysis which often incorporates information from further upstream. This allows the look-ahead traffic-adaptive controller to make better decisions in the long run. "Regular" traffic-adaptive control can be considered to be short-sighted compared to look-ahead traffic-adaptive control. In Section 4.1.1 the advantages of look-ahead traffic-adaptive control compared to traffic-actuated control are illustrated. Subsequently, in Section 4.1.2 it is explained why a new algorithm for look-ahead traffic-adaptive control has to be created.

4.1.1 Advantages of look-ahead traffic-adaptive control

The state of the art in traffic control is currently formed by traffic-adaptive systems. A traffic-adaptive system is able to flexibly adapt the signal plan to the current traffic situation. Adaptive traffic control systems are more advanced than traffic-actuated systems as decisions are made on the basis of a traffic model. The system is therefore able to evaluate a set of possible control decisions and choose the decision that is optimal with respect to the current objectives. Figure 4.1 illustrates some of the benefits of look-ahead traffic-adaptive control for a single intersection using some simple examples.

Figure 4.1(a) depicts a situation where a standard traffic-actuated controller will extend the green for the single vehicle (2) approaching from the west that just triggered the detector. A standard traffic-actuated controller bases its decision to extend or switch just on the basis of the measured demand for the current green phase. As vehicle 1 has just cleared the stop line, the result of this decision is that two vehicles (3, 4) are forced to stop to allow a single vehicle (2) to continue. The advantage of traffic-adaptive controller compared to a traffic-actuated controller is that the former is able to consider its decision on the intersection-level (and not just for the current green phase) and thus prevents this inefficiency by terminating the current green phase and giving green to the two vehicles (3, 4) that approach the intersection from the north and south.

A traffic controller that is not able to look further than the standard detector configuration of a traffic-actuated controller is unaware of the two vehicles (5, 6) that approach the intersection of Figure 4.1(b) from the west. A short-sighted controller will decide to give green to the single vehicle (3) approaching the intersection from the north, which forces the two vehicles approaching from the west to stop. This inefficiency can be prevented when a traffic controller is able to incorporate information from further upstream in its decision-making. This is one example where it is advantageous to base the decision to switch or to extend the current green phase on the basis of a look-ahead traffic controller.

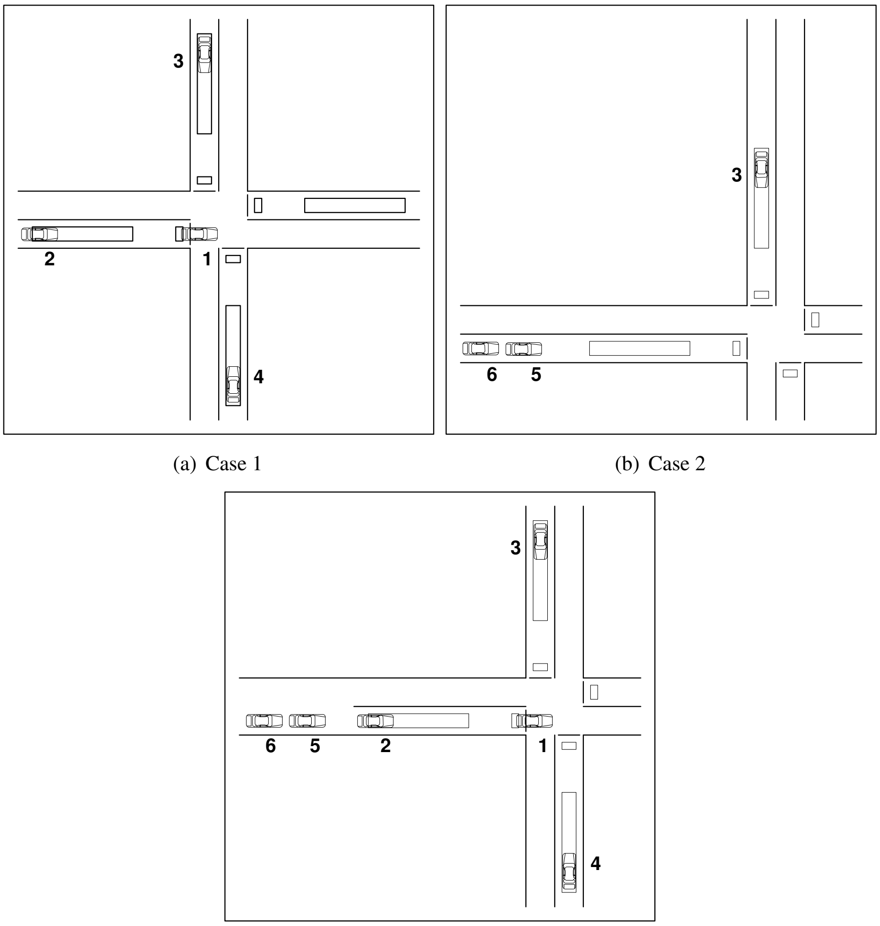
Finally, Figure 4.1(c) depicts a situation where a traffic-adaptive controller without look-ahead capabilities, will give preference to the vehicles (3, 4) that approach the intersection from the north and the south. The result of this decision in this case is that the three vehicles (2, 5, 6) approaching from the west are forced to stop. This example shows the advantage of look-ahead traffic-adaptive control. This type of control is able to consider the traffic state on all approaches of the intersection and has a wider field of view and is thus able to make better decisions.

4.1.2 The need for a new algorithm

Current approaches to look-ahead traffic-adaptive control use a stage-based approach to traffic control as opposed to the movement-based approach that is employed by state of the art in traffic-actuated control. The movement-based approach is more flexible than the stage-based approach as it allows green for signals in different phases to start sooner if demand for all conflicting movements in the current phase has cleared. Furthermore, a faster algorithm than is currently provided by the state of the art is needed in order to enable co-operation between nearby intersection controllers. The approach used to coordinate nearby intersections is presented in Chapter 5 and depends on intersections being able to iteratively optimize their performance on the basis of information received from nearby intersections in a negotiation process.

4.2 The algorithm

In this section a new algorithm for look-ahead traffic-adaptive control will be presented. This new algorithm integrates the best elements of the algorithms reviewed in Chapter 3 and includes a number of improvements. The new algorithm e.g., employs a movement-based approach to traffic control, which although common in traffic-actuated control, has not been adopted yet by the state of the art in look-ahead traffic-adaptive control, where a stage-based

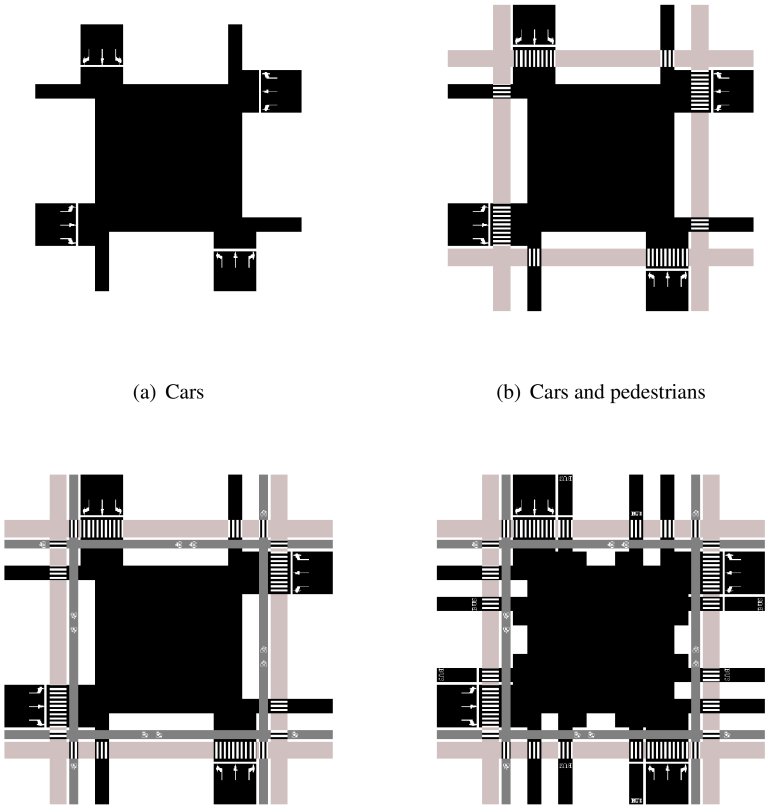


(a) Case 1 (b) Case 2

(c) Case 3

Figure 4.1: Cases used to illustrate the advantages of look-ahead traffic-adaptive control over traffic-actuated control

approach is still employed. This movement-based approach allows for a significant speed-up of the optimization process, which allows to search a larger search space in the same amount of time. Section 4.2.1 describes the differences between the movement-based approach and the stage-based approach. This section also describes how the movement-based approach is implemented in the developed look-ahead traffic-adaptive control algorithm. In Section 4.2.2 the algorithm used to search the resulting search tree is presented. The different performance functions that can be used in the optimization are presented in Section 4.2.3. The constraints applied throughout the optimization are described in Section 4.2.4. Section 4.2.5 concludes this section by describing how the size of the search space can be further reduced by manipulating the resolution of the optimization horizon.



(a) Cars

(b) Cars and pedestrians

(c) Cars, bicycles and pedestrians

(d) Cars, bicycles, pedestrians and public transport

Figure 4.2: Different intersection geometries [83]

4.2.1 Control decisions

In many European countries, in contrast to the U.S.A., controlled intersections often have separate infrastructure for bicycle and transit traffic movements. The main reason for this is to improve traffic safety by separating the weaker from the stronger road users in time and space, and to realize priority treatment of certain categories of road users. As a result, the geometry of an intersection can be rather complex as shown in Figure 4.2. In Figure 4.2(a) an intersection is shown that consist solely of infrastructure for cars. This intersection is extended in Figures 4.2(b), 4.2(c), and Figure 4.2(d) with dedicated infrastructure for pedestrians, bicycles and public transport respectively.

As the complexity of an intersection's geometry increases so does the control structure (i.e., the composition and the sequence of the green phases). The majority of the controlled intersections are controlled by a vehicle-actuated controller (e.g., in The Netherlands about

	Signal groups	Regular phases		Maximal phases	
Geometry	Number	Number	Search space	Number	Search space
Cars	12	111	111^N	17	17^N
+ pedestrians	20	2186	2186^N	112	112^N
+ bicycles	28	23362	23362^N	352	352^N
+ public transport	40	105722	105722^N	834	834^N

Table 4.1: Number of possible phases for different intersection geometries

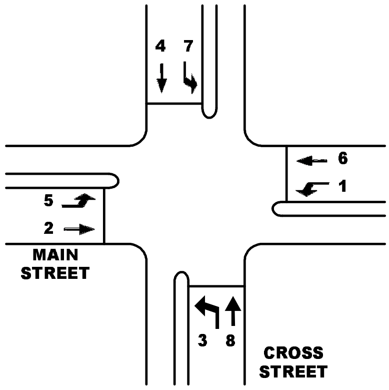
85%). Many techniques [19, 42, 56, 83, 103] exist to determine the best structure of a vehicle-actuated controller. Commonly the choice of a control structure is based on the critical conflict group only. Depending on the sequence of realization, the internal lost times will be longer or shorter. This enables the minimization of internal lost times by choosing a sequence of green phases with minimum clearance times. Each structure has its own clearance times and so its own minimum cycle times. The critical conflict group is the conflict group that, despite being sequenced efficiently, demands the longest cycle time. As the capacity of an intersection is determined by the critical conflict group this is often used as a basis for the creation of the control structure.

Although the look-ahead adaptive control algorithm is able to determine the phase-sequence on-line it is still important to provide the look-ahead adaptive control algorithm with a control structure to work with as it is not possible to evaluate all possibilities on-line in real time. Table 4.1 shows e.g., the size of the search space to be evaluated to find a truly optimal control structure for each of the intersections in Figure 4.2 for an N seconds planning horizon. Phases define the signal groups that are allowed to receive green simultaneously. In the table a difference is made between “regular” phases and “maximal” phases. “Maximal” phases are phases for which the signal groups are not contained in any other, larger phase. This difference is important to make, as the state of the art in traffic-actuated controllers uses a movement-based approach whereas look-ahead adaptive control still uses a stage-based approach. In order to gain the same level of flexibility with a stage-based approach as with a movement-based approach stages have to be incorporated that correspond to the non-maximal phases, which greatly increases the search space. The stage-based approach is less flexible than the movement-based approach as it does not allow movements to end earlier than the stage ends. As a result opportunities to give green to a subsequent movement earlier are lost. Furthermore, as clearance times in the stage-based approach are applied to the phase instead of to the individual movements, the stage-based approach can result in the application of clearance times that are larger than necessary. The algorithm proposed by this thesis therefore adopts a movement-based approach since this significantly reduces the size of the search space.

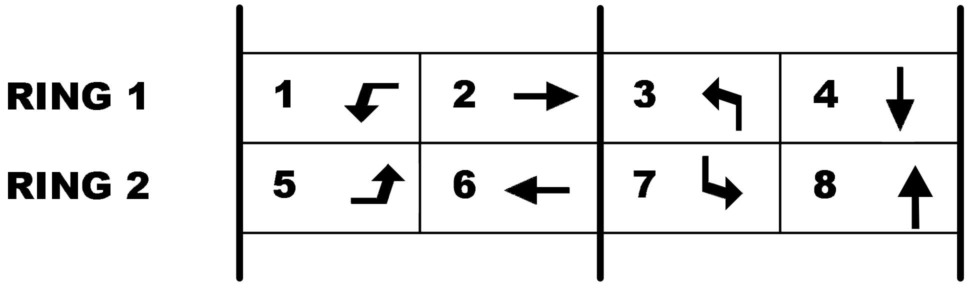
In the remainder of this section the American (NEMA) and Dutch (RWSC) approach to movement-based traffic-actuated control are introduced. A new movement-based look-ahead traffic-adaptive control is subsequently proposed.

The NEMA approach

The National Electrical Manufacturers Association (NEMA) has defined a method for organizing signal groups in a dual-ring structure as part of their Traffic Signals standard. NEMA



(a) Signal group naming according to NEMA



(b) Standard NEMA dual-ring, eight-phase controller configuration

Figure 4.3: NEMA controller configuration

is an electronics industry group in the United States, and its Traffic Control Systems Section has created a set of manufacturing guidelines for traffic control hardware that is widely adopted in America. The initial technical standard for Traffic Signals, commonly referred to as TS1, was published in 1976, and subsequent revisions include the publication of TS2 in 1992 [84]. These standards ensure that equipment is based on proven designs, and that hardware available from different manufacturers is compatible and interchangeable. The following figures illustrate the standard NEMA dual-ring, eight signal groups controller configuration for a four-legged intersection. Figure 4.3(a) depicts the intersection geometry, and movements are labeled with the number of the corresponding signal group that serves those movements. Figure 4.3(b) shows the ring structure of the standard dual-ring, eight signal groups NEMA configuration.

Note that eight signal groups are shown, each of which accommodates a through movement or a left-turning movement. A “barrier” separates the north-south signal groups from the east-west signal groups. Any signal group in the top ring (signal group 1, 2, 3 and 4 of Ring 1) may be displayed with any signal group in the bottom ring (phases 5, 6, 7, and 8 of Ring 2) on the same side of the barriers without introducing any traffic conflicts. For simplicity, the right turns are omitted and assumed to proceed with the through movements. Figure 4.4 is a flow diagram illustrating the alternative phase sequences allowed by the

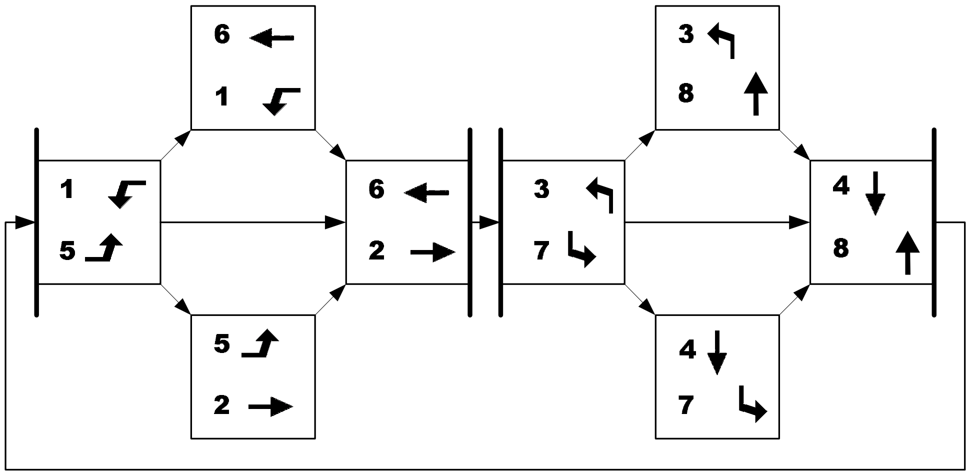


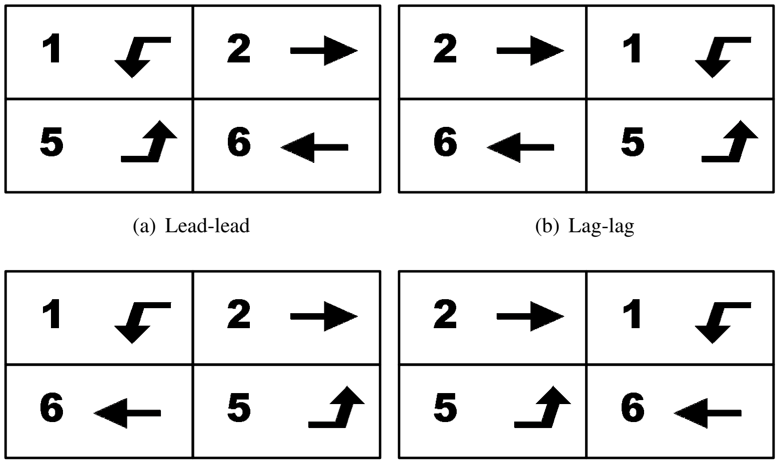
Figure 4.4: Control flow allowed by the NEMA dual-ring, eight-phase controller configuration

dual-ring, eight-phase NEMA configuration.

At any given time, one signal group is active on each ring of the controller. In the case of the standard NEMA dual-ring, eight signal groups configuration, there are always two active signal groups. Consider the controller starting in phase 1+5, serving left turn traffic from the east and west. In a stage-based approach or single-ring configuration, typically the left-turn phase would terminate, followed by a through movement phase, the 2+6 combination. This approach allows a total of four phases (which are maximal). However, in the case of a movement-based approach or multi-ring controller, each active signal group may time separately. Suppose while in phase 1+5, there has been a sufficient gap between actuations, so that signal group 5 gaps out and ring 2 switches to signal group 6, allowing for westbound (right-to-left across the page) through traffic to start service while ring 1 continues serving signal group 1, westbound left turning traffic. Rather than having to follow a fixed cyclic phase sequence, such as 1+5, 2+6, 3+7, 4+8, the dual-ring controller has the flexibility of choosing from a set of eight alternative phases instead of the four allowed by the stage-based approach. Using multiple rings and barriers is a standard way of specifying the allowable subset of phase sequences that are considered a safe progression of phases. Modern controllers are now capable of operating several rings, phases, and barriers.

The sequence of the signal groups is determined in advance and established in the ring structure. In dual-ring operation, there are four different sequences possible, which are shown in Figure 4.5. Signal group sequence names are linked to whether or not the left turn precedes, or leads, the opposing through movement. For each of the two major signal groups, there are four basic signal group sequences:

1. *Lead-lead*: The left-turn movements lead the opposing arterial movement,
2. *Lag-lag* : The left-turn movements lag the opposing arterial movement,
3. *Lead-lag* : One of the left turn movements leads the opposing arterial movement while the other left turn movement lags the opposing arterial movement,



(a) Lead-lead (b) Lag-lag

(c) Lead-lag (d) Lag-lead

Figure 4.5: Possible phase sequences for NEMA dual-ring, eight-phase controller configuration

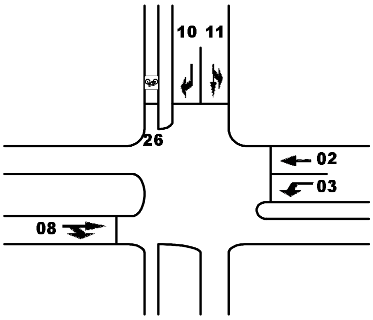
4. *Lag-lead* : The mirror image of the lead-lag phasing pattern.

The RWSC approach

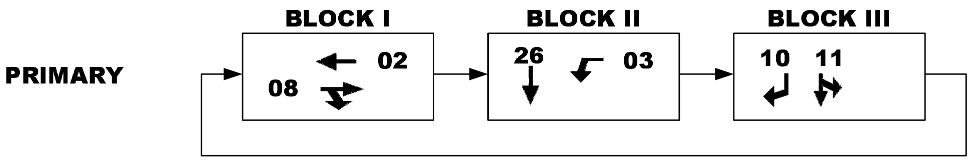
The Dutch Ministry of Transport has defined a method for organizing signal groups in a block structure as part of their Traffic Signals standard [27, 28]. This method is referred to as the RWSC-approach. A block is defined by this method as a collection of signal groups that can be green simultaneously. In this method the sequence of the signal groups is controlled on the higher level by the block procedure whereas the time a signal group remains green is determined on a lower level by the signal group procedure.

The preferred sequence in which signal groups get green is specified in the so-called block structure. The critical conflict group is determined (e.g., with VRIGEN [83]). For each signal group in the dominant conflict group a block is defined. This block is a set consisting of a signal group from the dominant conflict group and other signal groups that have no mutual conflicts (green phase combination). The sequence of the blocks is such that the time that is lost when switching between the signal groups of the dominant conflict group is minimal. For the intersection of Figure 4.6(a), the block structure may look as that of Figure 4.6(b) (the description of the block-procedure including the examples have been derived from [2, 27]).

The standard procedure of the blocks is as follows. If a block is in an active state, the signal groups of the block are allowed to become green. If all signal groups of a block have become green or if the decision has been taken to skip the green phase for the signal group the next block becomes active. The signal groups of an active block get “the turn to become green” as soon as all conflicting signal groups of the preceding block are in the state yellow, red, or parallel green. From this moment on, the signal group procedure takes over the further course of the signal group. As soon as each signal groups of the block has



(a) Intersection geometry



(b) Block structure for the intersection of Figure 4.6(a)

Figure 4.6: RWSC controller configuration

or has had the turn to become green, the next block becomes active. Input for this procedure is information from detectors and from the status of the signal groups.

During the control process the following situation may occur: suppose that all signal groups of the intersection are red and block I is active. Assume that there is no request for green for signal group 8 and that there is demand for signal groups 2 and 3. Signal group 2 and 8 are both allowed to turn green. As there is no demand for signal group 8, this signal group waives the right to turn green. Then Block II becomes active, and signal group 8 loses its right to become green. Because all conflicting signal groups of signal group 3 are red, this signal group can become green simultaneously with signal group 2 (even though they belong to different blocks). This shows that the block procedure offers the possibility for signal groups of different blocks to become green simultaneously.

Early switch to green Now suppose that the following situation occurs: Signal group 10 has a request for green, signal group 26 has no request, signal group 2 is red, block II is active and signal group 8 is green. The signal groups 3 and 26 are red; they can become green only after signal group 8 has become red. This gives an unacceptable control situation: signal groups 2 and 26 are red, there is traffic waiting for signal group 10, 8 is green and there should be the possibility to give green to signal group 10 simultaneously.

For such situations the block procedure has the possibility to deviate from the standard sequence in which signal groups come into the state right on green. A signal group s (e.g., signal group 10) can switch to green earlier under the following conditions:

- There is a request/demand for signal group s,
- All conflicting signal groups are in the state red, yellow, or parallel green, and

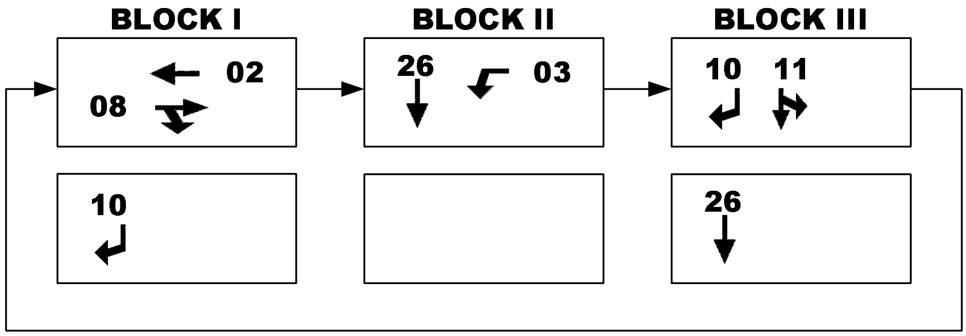


Figure 4.7: Alternative block structure for the intersection of Figure 4.6(a)

- There are no conflicting signal groups with a request for green that should get green before signal group *s* according to the sequence of the blocks.

In the situation described above, signal group 10 satisfies all conditions to get advanced green. By means of the procedure of advanced green the control program can deviate from the normal sequence of green phases. Still there are some possibilities that situations occur that will be incomprehensible or unacceptable for the road users.

Alternative green Given the block structure of Figure 4.6(b) there is a possibility that a situation occurs that is unacceptable for the road user. Suppose that signal groups 10 and 26 both have a request. Signal group 2 is red and signal group 8 is green. Block II is active. Signal groups 3 and 26 are red: they can become green only after 8 has become red. This situation is unacceptable for road users: 2 and 26 are red, 8 is green, and traffic is waiting for signal group 10. There is a possibility for 10 to switch to green, but the conditions to advance to green are not satisfied, because there is a request for 26. In such a situation a possibility can be created by defining a so-called alternative signal group in block I. This goes as follows: The block structure as given in Figure 4.6(b) is called the primary block structure. Next to this primary block structure, alternative blocks can be specified.

In the alternative block the signal groups can be included that may become green if not all signal groups of the primary block get green. In the example signal group 10 may become green in block I if signal group 8 is green, but signal group 2 is red. A signal group in the alternative block receives the right to become green if at least the following conditions are satisfied:

- The block the signal group is assigned to as an alternative is active
- Signal group is in the state waiting green
- There is a request for green
- All conflicting signal groups are in the state red, yellow, or parallel green
- All conflicting signal groups in the active primary block have no request for green or their green time has elapsed.

In the situation that has been described above, signal group 10 satisfies these conditions and has the possibility to receive the right to turn green. The consequence is that all conflicting signal groups in the primary block lose that right, i.e., they are skipped. The specification of an alternative realization of a signal group *s* is only useful if:

- The signal group is not a member of the next block
- There is at least one signal group in the primary block that has no conflict with the signal group *s*.

In the following section we will describe how the stream-based approach of traffic-actuated control can be incorporated in a look-ahead traffic adaptive control algorithm.

A new look-ahead approach

The look-ahead algorithm we will develop, like current traffic-actuated control approaches, but unlike look-ahead traffic-adaptive control approaches, uses a movement-based approach instead of a stage-based approach. It adopts the block procedure of the RWSC approach in the calculation of the optimal control trajectory over the optimization horizon. In Figure 4.8 the advantage is illustrated of employing a movement-based strategy over a stage-based strategy for the intersection of Figure 4.6(a). The figure shows a possible timing of green and red intervals for the signal groups given the block structure of Figure 4.6(b). Time in the figure progresses from left to right across the page. The period of time in which a signal group is green or red is denoted by a bar that is accordingly colored. The interval of time in which a block is active is depicted at the top side of the picture. The figure shows that as soon as signal group 8 of Block I has terminated, signal group 3 of Block II is allowed to switch to green. This is allowed since signal group 3 has no conflict with the signal groups of Block I that are still green (in this case this is signal group 2). Similarly signal group 10 of Block III is allowed to advance to green as soon as the green phase for signal group 26 has ended.

As the movement-based strategy allows movements to switch to green as soon as all conflicting movements have cleared the number of green combinations possible with just three blocks would have required six distinct stages if instead a stage-based approach had been applied. These six stages are depicted on the bottom of the picture. Optimizing over the horizon using blocks instead of stages significantly reduces the branching factor of the tree and thus significantly reduces the search space without making sacrifices with regard to the quality of the solution.

As the sequencing of the blocks is determined on-line, the alternative green of the RWSC approach is implicitly made possible if next to the timing of the signals also the sequence of the blocks is allowed to be the subject of optimization. This is illustrated in Figure 4.9, where signal group 10 of block III is allowed to switch to green as soon as the conflicting signal group 2 has terminated. As a result signal group 11 is also allowed to switch to green earlier than that would have been possible with a fixed block sequence. If this is undesired behavior and if the optimization of the block sequence is not allowed then the alternative green offered by the RWSC approach can still be made explicitly possible when desired.

In the block procedure, the signal of the next block switches to green as soon as the conflicting signals of the previous block have cleared. As it is not known in advance which block will be the next block, as this will be determined later on in the optimization, it is

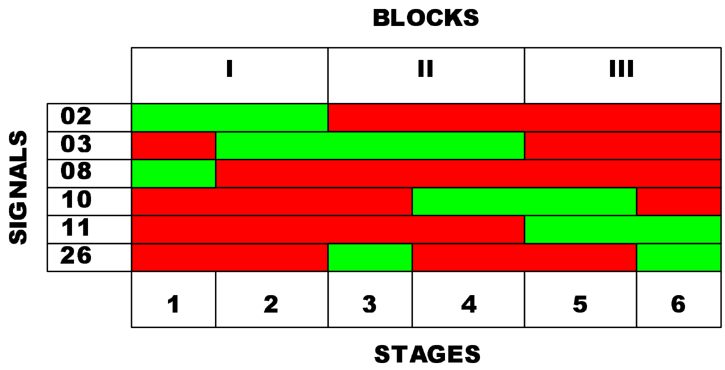


Figure 4.8: Number of blocks versus the number of stages for the intersection of Figure 4.6(a)

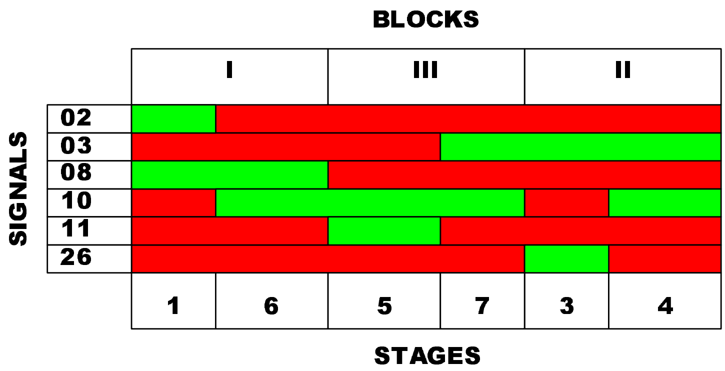
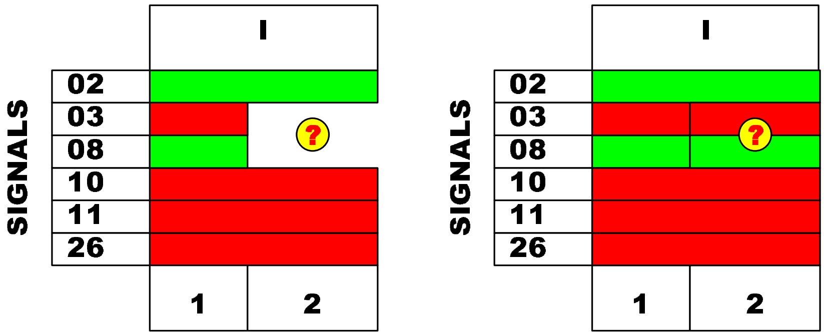


Figure 4.9: Alternative block sequence

not possible to determine the performance (i.e., resulting delay, number of stops, etc.) of the block up to the end time of the block for all signals as some signals will have cleared before that time. To calculate the performance of the block in a look-ahead adaptive control algorithm two options are available and these are both depicted in Figure 4.10.

Both subfigures of Figure 4.10 show a situation where signal group 8 of Block I has terminated. As soon as signal group 8 has terminated (and the necessary time to clear the intersection has passed) signal group 3 can switch to green as this signal group has no conflicts with the other signal groups of Block I that are still green (i.e., signal group 2). However, as in this stage of the optimization process it is not known which block will follow Block I, it cannot be assumed that signal group 3 will indeed be allowed to switch to green or that it will remain red. As the state of the signal group between the moment signal group 3 can possibly start and the moment the block ends is unknown the performance of signal group 3 cannot be determined either.

Two options are available to handle this. The first and most logical one is illustrated in Figure 4.10(a) where the performance of each signal is assumed to be zero after the signal has cleared. However, this has an unwanted side-effect, as it is impossible in that case to



(a) Block performance - Assume unknown (b) Block performance - Assume green

Figure 4.10: Possible options for determining the performance of a block

improve upon the performance of a block. Any decision taken after the signal has cleared will only further diminish the performance and will not contribute to it. This is why the look-ahead algorithm uses the option of Figure 4.10(b) in which signals that have cleared are assumed to remain green until the end of the block. In this case none of the conflicting signals that are not part of the block can go to advanced green, and this must be compensated for while calculating the performance of the next block. This will be described in Section 4.2.3

4.2.2 Search algorithm

Consider the decision tree of Figure 4.11 where the center of the figure (the root of the tree) denotes the current state k_0, labeled now. Four paths lead away from this state, each corresponding to the result of a control decision u_0 that can be taken at this state. The control decision made results in a new state from which again four control decisions can be made. Each decision takes us further into the future and deeper into the tree formed by the states resulting from each possible decision. Let k_{j+1} denote the successor state of k_j following the implementation of control u_j at time step k for time interval $[k\Delta t, (k+1)\Delta t)$. As the decision space has a tree-like structure, the search for the optimal sequence of decisions $u^* = \{u_k^*\}_{k=1}^K$ corresponds to building the tree.

Since search space size grows exponentially with problem size, it is not possible to explore all assignments except for the smallest problems. The only way out is to not look at the whole search space. Efficiency in searching the decision space is considered by the degree to which the entire tree will not have to be built to find an optimal path. In [102] several well-known algorithms are assessed based on computational speed and on the quality of the results (in terms of vehicle delay). The search algorithm presented in this section is based on a dynamic programming formulation, similar to the one found in [101]. However, the algorithm described below uses blocks instead of stages, creating a more efficient algorithm. Although the usage of blocks instead of stages increases the algorithm's complexity, the computational requirements are much less than when stages would have been used. To improve further upon the performance of the algorithm it is extended with techniques from the branch-and-bound algorithm. The algorithm is described below.

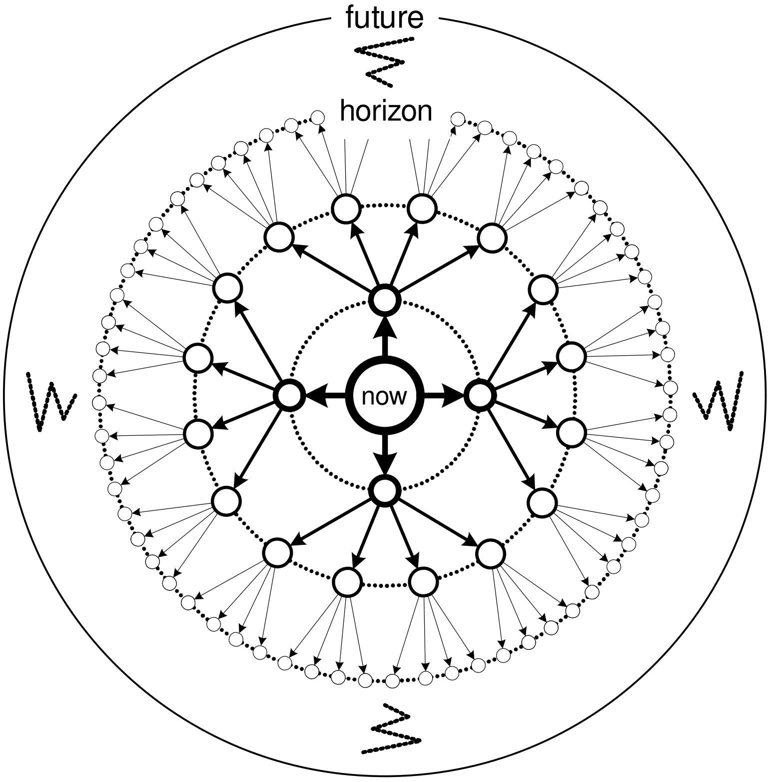


Figure 4.11: Decision tree

Dynamic programming formulation

Dynamic programming is a method for solving problems exhibiting the properties of overlapping subproblems and optimal substructure. A problem is said to have overlapping subproblems if the problem can be broken down into subproblems that can be reused several times whereas a problem is said to have an optimal substructure if its optimal solution can be constructed efficiently from optimal solutions to its subproblems. The reader is referred to [35] for a more thorough treatment of dynamic programming.

In our formulation the problem is to determine the optimal sequence and duration of the blocks over an optimization horizon. In our formulation each decision stage represents the total time allocated to a block. Each decision stage is divided into states. A state encompasses the information required to go from one decision stage to the next. In our formulation the state denotes the total time allocated to the blocks upto and including the current decision stage. Starting with an initial block, the algorithm treats each block as a decision stage, and optimizes over as many cycles as necessary to obtain an optimum.

The following notation is introduced:

$S \equiv$ Set of signals. Individual signals are indexed by s.

B	$\equiv$	Set of blocks. The cardinality of this set will be denoted $\lvert B \rvert$. Each block is itself a set of signals that are allowed to turn green in the block. Individual blocks are indexed by i. B_b denotes the currently active block.
$g_s^{\max}$	$\equiv$	Maximum green time for signal s.
$g_i^{\text{block,max}}$	$\equiv$	Maximum green time for block i.
j	$\equiv$	Index for decision stages of the DP.
k	$\equiv$	State variable denoting the total number of allocated time steps.
K	$\equiv$	Length of the optimization horizon in discrete time steps.
u	$\equiv$	Control variable denoting the number of time steps allocated to the block.
$U(j,k)$	$\equiv$	Set of feasible control decisions for a given decision stage j and state k.
$F(j,k,u)$	$\equiv$	Performance of control u, for a given decision stage j and state k.
$V(j,k)$	$\equiv$	Cumulative performance for a given decision stage j and state k.

The first decision stage in our formulation concerns the number of additional time steps to allocate for the block that is currently active. The first decision stages therefore decides about the time to allocate to block B_b, the second decision stage decides about the time to allocate to block $(b+1) \mod (\lvert B \rvert)$ to decision stage 2, etc. The exact number of decision stages used is a by-product of the computations and does not have to be specified in advance. Given a value for the state variable k, the control variable u can assume values from the following discrete set:

$$U(j,k) = \left\{ i \middle| i = 0, \ldots, \min \left\{ k, g^{\text{block,max}}_{(b+j-1) \mod (\lvert B \rvert)} \right\} \right\} \tag{4.1}$$

where $g_i^{\text{block,max}}$ is the maximum of the maximum green time $g_s^{\max}$ for each of the signals in block i and is calculated as follows:

$$g_i^{\text{block,max}} = \max_{s \in B_i} \{ g_s^{\max} \} \tag{4.2}$$

By allowing u to assume a value of 0 blocks can be skipped and any desired block sequence can be generated. Figure 4.12 shows how consecutive decision stages are related. It shows that the previous state k_{j-1} can be determined on the basis of the current state k_j and chosen control decision u_j as follows:

$$k_{j-1} = k_j - u_j \tag{4.3}$$

The forward recursion of the algorithm is now presented. In the following k_{j-1} is calculated as a function of k_j and u_j via (4.3). The algorithm starts with decision stage $j = 1$, and proceeds recursively to $j = 2, 3, \ldots$ At each decision stage, the method calculates the best control decision $u^*_{j,k}$ for each possible value of the state variable k. The performance function (to be minimized) is assumed to take the following form:

$$F(1,k_1,u_1) + F(2,k_2,u_2) \ldots + \ldots \tag{4.4}$$

Given that the performance functions of interest are quantities such as delay, number of stops, and queue lengths, there is no loss of generality in assuming that $F(j,k,u) \geq 0$

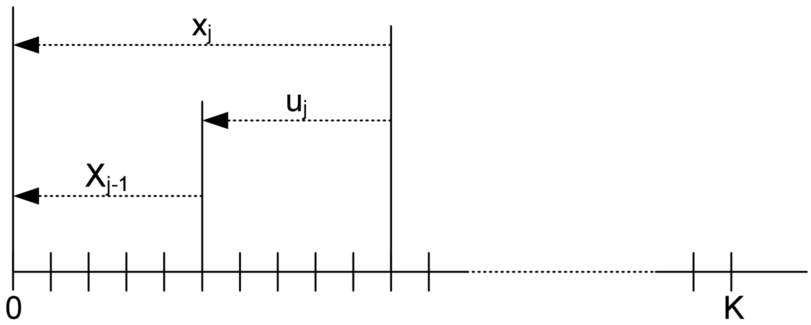


Figure 4.12: The relation between stages is determined by states and chosen control decisions

for all j and all feasible values of k and u. The calculation of the performance functions is discussed in greater detail in the Section 4.2.3.

The optimal performance, denoted by $V(j,k)$ is a function of the immediate performance $F(j,k,u)$ of implementing the optimal control decision u^*

$$V(j,k) = \min_{u \in U(j,k)} \{F(j,k,u) + V(j-1,k-u)\} \tag{4.5}$$

The corresponding forward recursion algorithm is provided in Algorithm 4.1. The recursion ends if there is nothing to be gained from evaluating a new decision stage and at that time the optimal solution can be retrieved. This is the case if all of the previous $|B|$ stages have not improved the performance. Since a later decision stage allows more phase changes for the same value of the state variable, it follows that $v_{j-1,K} \geq v_{j,K}$. Furthermore, note that if $v_{j-1,K} = v_{j,K}$, it follows that there is nothing to be gained by allowing the specific block change associated with decision stage j. This reasoning is applied to the $|B|-1$ stages preceding decision stage j.

Algorithm 4.1 Forward recursion

1: initialize $v_0 \leftarrow 0$, $k \leftarrow 1$
2: **for** $k = 0 \ldots K$ **do**
3: $\quad v_{j,k} \leftarrow \min_{u \in U(j,k)} \left\{F(j,k,u) + v_{j-1,k-u}\right\}$
4: $\quad$ record $u^*_{j,k}$, an optimal solution to the above problem
5: **end for**
6: **if** $(j < |B|)$ **then**
7: $\quad j \leftarrow j+1$, and repeat from step 2
8: **else** {check if done}
9: $\quad$ **for** $i = 1 \ldots |B|-1$ **do**
10: $\quad\quad$ **if** $v_{j-i,K} \neq v_{j,K}$ **then**
11: $\quad\quad\quad j \leftarrow j+1$, break for loop and repeat from step 2
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: **end if**

The optimal solution can subsequently be retrieved by determining the optimal trajectory of states and the associated optimal controls using Algorithm 4.2. Let J denote the last decision stage for which the value function has been calculated in the forward recursion. Then, we may retrieve an optimal policy by tracing back through the table that has recorded the optimal control decisions, $u^*_{j,k}$ for $j = J - (|B| - 1), J - (|B| - 2), \ldots, 1$. Note that since the forward recursion ends only if $v_{j,K} = v_{j-1,K}$ for $j = J - |B|, \ldots, J$, the controls satisfy $u^*_{j,K} = 0$ for these j, and consequently, it is sufficient to retrieve controls starting with decision stage $j = J - (|B| - 1)$.

Algorithm 4.2 Retrieval of optimal policy

```
k* ← K
for j = J − (|B| − 1), J − (|B| − 2) ... 1 do
    read u*_{j,k*} from table computed in forward recursion
    if j > 1 then
        k* ← k* − u*_{j,k*}
    end if
end for
```

The formulation of the algorithm thus far relies purely on Dynamic Programming. To further improve upon the efficiency of the algorithm the elementary mechanics of the branch-and-bound algorithm are included.

Bounding

Use of the branch-and-bound technique accelerates the evaluation of state values by restricting evaluation to only states along control trajectories with a potential to obtain a lower total cost than the best trajectory found so far in the search.

The following equation is a reformulated version of (4.5) with the inclusion of a lower bound value, LB, used to eliminate control trajectories from further consideration when their corresponding cost already exceed the upper bound, UB, established by a lower cost trajectory found previous in the search:

$$V(j,k) = \begin{cases} \min_{u \in U(j,k)} \left\{ F(j,k,u) + v_{j-1,k-u} \right\} & \text{if } v_{j-1,k-u} + \mathrm{LB} < \mathrm{UB} \\ 0 & \text{if } j = 1 \wedge k = 0 \\ \infty & \text{otherwise} \end{cases} \tag{4.6}$$

The search for the lowest cost control trajectory proceeds as previously described with the additional initialization of the lower and upper bounds, $\mathrm{LB} = 0, \mathrm{UB} = \infty$, and subsequent update of the upper bound whenever a full trajectory has been evaluated as follows:

$$\mathrm{UB} = \begin{cases} \min \{\mathrm{UB}, V(j,k)\} & \text{if } k = K \\ \mathrm{UB} & \text{otherwise} \end{cases} \tag{4.7}$$

In order to obtain a tight upper bound early on in the optimization process, the algorithm initializes the upper bound with the current performance of the then optimal policy determined in the previous update interval.

The lower bound is obtained by calculating the minimum amount of delay imposed for each signal group due to constraints with respect to clearance intervals and minimum green times. The decision to end the block imposes a minimum delay on any vehicle remaining in queues serviced by that block. This delay is a lower bound on the future delay and can be added to the previous delay experienced along the current trajectory in order to prune the decision and its corresponding subtree.

The lower bound (LB) can be calculated as follows. First, the vehicles are stopped for the duration of the all-red clearance interval between the signals of the current block and the next block. Then, these vehicles must wait through at least the minimum green duration and the clearance interval for each subsequent block until the controller returns to the original block to serve these vehicles. In addition, a lower bound on delay contributions from every approach may be taken into account. Note that predicted arrivals may also be added to waiting queues during each time step to account for increasing queue lengths and the corresponding increase in delay per time step while waiting for the desired blocks. When the desired block begins serving a set of vehicles, the block must be green for at least the minimum green duration. During the minimum green interval, the departure rate can be used to estimate the decreasing queue length and the corresponding lower delay per time step. After the minimum green interval has elapsed, the controller may elect to extend that block or switch. In the calculation of the lower bound delay contribution from this queue it is assumed that the block will be extended, continuing to disperse the queue. However, the lower bound on the future delay for vehicles served by the subsequent block will assume that the controller switches immediately after the minimum green duration to the next block. In this manner, a lower bound on the delay may be calculated, incorporating a contribution of the delay from vehicles awaiting service for each block in the block structure.

4.2.3 Performance functions

The following notation is introduced:

λ_s^{start}	$\equiv$	Green start lag for signal s.
λ_s^{end}	$\equiv$	Green end lag for signal s.
y_s	$\equiv$	Yellow time for signal s.
$r_{r,s}$	$\equiv$	All-red time needed to safely switch from signal r to signal s.
$\mathbf{q}_s^{\text{in}}$	$\equiv$	Ordered multi-set[1] of estimated arrival times for signal s.
$\mathbf{q}_s^{\text{out}}$	$\equiv$	Ordered multi-set of estimated departure times for signal s.
$\widehat{\mathbf{q}}_{s,j,k}^{\text{in}}$	$\equiv$	Ordered multi-set of estimated arrival times of arrivals that are queued for signal s for a given decision stage j and state k.
$\widehat{\mathbf{q}}_{s,j,k}^{\text{out}}$	$\equiv$	Ordered multi-set of estimated departure times for departures of signal s for a given decision stage j and state k.
$\widehat{\mathbf{q}}_{s,j,k,\lvert\mathbf{q}_s^{\text{out}}\rvert}^{\text{out}}$	$\equiv$	Estimated departure time of the last vehicle $\left\lvert\widehat{\mathbf{q}}_{s,j,k}^{\text{out}}\right\rvert$ departing from signal s.
m	$\equiv$	Index for arrivals to the intersection.
$A(j,k,u,s)$	$\equiv$	The demand for signal s in the control interval determined by k and u .
$A(j,k,u,s)_1$	$\equiv$	Estimated arrival time of the first vehicle arriving at signal s, given state k and control decision u.

[1] A multi-set (or bag) is a generalization of a set in which the entries are not required to be distinct. An ordered multi-set is a multi-set in which the order of the items is important. In our case the entries are sorted by their estimated arrival time

$R(j,k,u,s,m)$	$\equiv$	The time after which the m^{th} vehicle will be able to depart signal s, given state k and control decision u.
$Q^{\text{in}}(j,k,u,s)$	$\equiv$	The (queued) arrivals remaining at signal s after having implemented control decision u to reach state k.
$Q^{\text{out}}(j,k,u,s)$	$\equiv$	The departures from the signal s after having implemented control decision u to reach state k.
$\text{GS}(j,k,u,s)$	$\equiv$	The time green can start for signal s, given state k and control decision u.
$\text{GE}(j,k,u,s)$	$\equiv$	The time green can end for signal s, given state k and control decision u.
$\text{GD}(j,k,u,s)$	$\equiv$	The green duration for signal s, given state k and control decision u.
$g^{\text{end}}_{j,k,s}$	$\equiv$	The time green can end for signal s, given state k and control decision u.

It will be assumed that the following information is available when calculating the first decision stage:

- The estimated arrival times for each vehicle approaching a signal ($\mathbf{q}^{\text{in}}_s$) for all $s \in S$
- The time of the last vehicle served by a signal ($\widehat{\mathbf{q}}^{\text{out}}_{s,0,0,|\mathbf{q}^{\text{out}}_s|}$) for all $s \in B_b$
- The time green started for a signal ($g^{\text{start}}_{0,0,s}$) for all $s \in B_b$

The control decision u^* that results in the best performance determines the information that is retained in the calculation of the subsequent stages. The demand for a signal s is determined on the basis of any remaining queues ($\widehat{\mathbf{q}}^{\text{in}}_{s,j-1,k-u}$) after having implemented the optimal control decision of the previous decision stage $j-1$, and the arrivals during the interval $[k-u,k)$ specified by the control decision (u):

$$A(j,k,u,s) = \widehat{\mathbf{q}}^{\text{in}}_{s,j-1,k-u} \cup \Big\{a|(a \in \mathbf{q}^{\text{in}}_s) \wedge (a \geq k-u) \wedge (a < k)\Big\} \tag{4.8}$$

where $\widehat{\mathbf{q}}^{\text{in}}_{s,0,0}$ is determined as follows:

$$\widehat{\mathbf{q}}^{\text{in}}_{s,0,0} = \Big\{a|(a \in \mathbf{q}^{\text{in}}_s) \wedge (a < 0)\Big\} \tag{4.9}$$

The state of a signal after the application of a control decision is determined by this demand ($A(j,k,u,s)$) and the time at which a vehicle is are able to depart ($R(j,k,u,s,m)$):

$$Q^{\text{in}}(j,k,u,s) = \{a_m|a_m \in A(j,k,u,s) \wedge R(j,k,u,s,m) < k\} \tag{4.10}$$

The departures for a signal are determined on the basis of the vehicles that have already left ($\widehat{\mathbf{q}}^{\text{out}}_{s,j-1,k-u}$) and the vehicles that depart during the control decision:

$$Q^{\text{out}}(j,k,u,s) = \widehat{\mathbf{q}}^{\text{out}}_{s,j-1,k-u} \cup \{a_m|a_m \in A(j,k,u,s) \wedge R(j,k,u,s,m) \geq k\} \tag{4.11}$$

The time at which a vehicle is able to depart depends on the saturation flow (q^{sat}_s) of the signal and:

- on the time when the green phase effectively starts ($\mathrm{GS}(j,k,u,s) + \lambda_s^{\mathrm{start}}$) *or*
- on the time of the last departure for the signal ($\widehat{\mathbf{q}}^{\mathrm{out}}_{s,j,k,|\mathbf{q}_s^{\mathrm{out}}|}$) if the green phase for the signal is continued.

It is determined as follows:

$$R(j,k,u,s,m) = \sum_{1}^{m-1} \frac{1}{q_s^{\mathrm{sat}}} + \begin{cases} \mathrm{GS}(j,k,u,s) + \lambda_s^{\mathrm{start}} & \text{if } s \notin B_{(b+j-2) \mod (|B|)} \\ \widehat{\mathbf{q}}^{\mathrm{out}}_{s,j,k,|\mathbf{q}_s^{\mathrm{out}}|} + \frac{1}{q_s^{\mathrm{sat}}} & \text{if } s \in B_{(b+j-2) \mod (|B|)} \end{cases} \tag{4.12}$$

The saturation flow q_s^{sat} dictates the minimum inter-departure time between consecutive vehicle departures. The time of the most recent departure from a signal $\widehat{\mathbf{q}}^{\mathrm{out}}_{s,j,k,|\mathbf{q}_s^{\mathrm{out}}|}$ ensures that the minimum inter-departure time is respected for a signal that is green for two consecutive blocks.

To calculate how the state of intersection is affected after the application of a chosen control decision it is important to know when the green signal for a signal group starts. Green starts for signal group s if it has demand ($|A(j,k,u,s)| \neq 0$) and if it is active in the block ($s \in B_{(b+j-1) \mod (|B|)}$). In order not to waste the available green time, green starts no sooner than necessary to allow the first arrival ($A(j,k,u,s)_1$) to pass without delay. Of course, the green signal is allowed to start only after any conflicting signal groups have cleared. The algorithm used to determine the time green starts is given in Algorithm 4.3.

Algorithm 4.3 Calculate $\mathrm{GS}(j,k,u,s)$ the time green can start for signal s

if $|A(j,k,u,s)| \neq 0$ **then** {there is demand}
    $g^{\mathrm{start}}_{j,k,s} \leftarrow A(j,k,u,s)_1$ {green starts no sooner than necessary}
    for all $r \in B_{(b+j-2) \mod (|B|)}$ **do** {signals that are green in the previous block}
        $g^{\mathrm{start}}_{j,k,s} \leftarrow \max\{g^{\mathrm{start}}_{j,k,s}, g^{\mathrm{end}}_{j-1,k-u,r} + r_{r,s}\}$ {green starts no sooner than conflicting signals have cleared}
    end for
else {there is no demand}
    $g^{\mathrm{start}}_{j,k,s} \leftarrow k$ {as there is no demand, green is skipped}
end if
return $g^{\mathrm{start}}_{j,k,s}$

The time green can start for a signal group depends on the time that conflicting signal groups have cleared. Note that green can start before the start time of the block if the conflicting signal groups have cleared before that time. The time conflicting signals have cleared depends on the time the green phase for the signal group has ended.

The time green can end for a signal is determined after having evaluated all $u \in U(j,k)$. It is determined for the best performing control decisions using Algorithm 4.4. The time green ends for a signal depends on:

- whether it is active in the block ($s \in B$),
- whether it served any vehicles ($\widehat{\mathbf{q}}^{\mathrm{out}}_{s,j,k,|\mathbf{q}_s^{\mathrm{out}}|} \geq g^{\mathrm{start}}_{j,k,s}$),
- the time it has started ($g^{\mathrm{start}}_{j,k,s}$),

- the minimum green time ($g_s^{\min} + y_s$), and
- the time the last vehicle was served ($\widehat{\mathbf{q}}^{\text{out}}_{s,j,k,|\mathbf{q}_s^{\text{out}}|}$).

As violations of the maximum green time can be determined before the decision is evaluated, there is no need to check for it here. The algorithm used to determine the time green ends is given in .

Algorithm 4.4 Calculate $\mathrm{GE}(j,k,u,s)$ the time green can end for signal s

---

Require: The departures ($\widehat{\mathbf{q}}^{\text{out}}_{s,j,k}$) that result after having implemented the optimal control decision have been determined
1: **if** $\widehat{\mathbf{q}}^{\text{out}}_{s,j,k,|\mathbf{q}_s^{\text{out}}|} \geq g^{\text{start}}_{j,k,s}$ **then** {demand has been served}
2: $\quad g^{\text{end}}_{j,k,s} \leftarrow \max\left\{g^{\text{start}}_{j,k,s} + g_s^{\min} + y_s, \widehat{\mathbf{q}}^{\text{out}}_{s,j,k,|\mathbf{q}_s^{\text{out}}|}\right\}$ {determine end of green phase}
3: **else** {no demand has been served}
4: $\quad g^{\text{end}}_{j,k,s} \leftarrow k - u$ {as no demand has been served, the green phase was skipped}
5: **end if**
6: **return** $g^{\text{end}}_{j,k,s}$

The result of $Q^{\text{in}}(j,k,u,s)$, $Q^{\text{out}}(j,k,u,s)$, and $\mathrm{GE}(j,k,u,s)$ for the best performing control decision are recorded as $\widehat{\mathbf{q}}^{\text{in}}_{s,j,k}$, $\widehat{\mathbf{q}}^{\text{out}}_{s,j,k}$, and $g^{\text{end}}_{j,k,s}$ respectively for subsequent use in the calculation of the next decision stage.

We now proceed to the calculation of several performance indexes.

Number of stops

The following notation is introduced:

$\mathrm{S}(j,k,u,s)$	$\equiv$	The number of vehicles that have to stop while approaching signal s as a result from applying control decision u, given state k.
$\mathrm{SR}(j,k,u,s)$	$\equiv$	The number of vehicles that have to stop while approaching signal s during the red period $[k-u,k)$.
$\mathrm{SG}(j,k,u,s)$	$\equiv$	The number of vehicles that have to stop while approaching signal s during the green period $\left[g^{\text{start}}_{j,k,s}, k\right)$.
$\mathrm{SGG}(j,k,u,s)$	$\equiv$	The actual number of vehicles that have to stop while approaching signal s during the green period $\left[g^{\text{start}}_{j,k,s}, k\right)$.
$\mathrm{SGR}(j,k,u,s)$	$\equiv$	The number of vehicles that were wrongfully assumed to be stopped in the calculation of the previous block.
w_s	$\equiv$	Weighting factor that can be applied to favor a signal s.

The total number of stops for an intersection that results from applying control decision (u) at state (k) can be determined by summing the total number of stops for each signal:

$$F(j,k,u) = \sum_{s \in S} \mathrm{S}(j,k,u,s) \tag{4.13}$$

where $\mathrm{S}(j,k,u,s)$ denotes the number of stops per signal group and is determined differently for signals that are red ($s \notin B_{(b+j-1) \mod (|B|)}$) and signals that are green ($s \in$

$B_{(b+j-1) \mod (|B|)}$:

$$\mathrm{S}(j,k,u,s) = w_s \cdot \begin{cases} \mathrm{SR}(j,k,u,s) & \text{if } s \notin B_{(b+j-1) \mod (|B|)} \\ \mathrm{SG}(j,k,u,s) & \text{if } s \in B_{(b+j-1) \mod (|B|)} \end{cases} \tag{4.14}$$

In calculating number of stops for a signal that is red ($\mathrm{SR}(j,k,u,s)$), we count those vehicles that arrive in the time-interval $[k-u,k)$:

$$\mathrm{SR}(j,k,u,s) = |\{a | a \in A(j,k,u,s) \wedge a >= k-u \wedge a < k\}| \tag{4.15}$$

To determine the number of stops for a signal that is allowed to turn green ($\mathrm{SGG}(j,k,u,s)$), we count those vehicles that arrive in the time-interval $[\mathrm{GS}(j,k,u,s),k)$ and are unable to leave:

$$\mathrm{SGG}(j,k,u,s) = \left|\left\{a | a \in A(j,k,u,s) \wedge a \geq \mathrm{GS}(j,k,u,s) \wedge R(x_j,u_j,s,m) \geq k\right\}\right| \tag{4.16}$$

As green for a signal can start earlier than the start of the block (i.e. if the conflicting signals of the previous block have cleared earlier than the previous block ends), we have to compensate for any stops that were unjustly attributed to the previous block. The number of stops ($\mathrm{SGR}(j,k,u,s)$) that was unjustly attributed to the previous block can be calculated as follows:

$$\mathrm{SGR}(j,k,u,s) = |\{a | a \in A(j,k,u,s) \wedge a \geq \mathrm{GS}(j,k,u,s) \wedge a < k-u\}| \tag{4.17}$$

This compensation term ($\mathrm{SGR}(j,k,u,s)$) is subtracted from the actual number of stops (SGG) for the signal to determine the additional number of stops caused by the green signal ($\mathrm{SR}(j,k,u,s)$):

$$\mathrm{SG}(j,k,u,s) = \mathrm{SGG}(j,k,u,s) - \mathrm{SGR}(j,k,u,s) \tag{4.18}$$

Delay

The following notation is introduced:

$\mathrm{D}(j,k,u,s)$	$\equiv$	The amount of delay experienced at signal s as a result of applying control decision u, given decision stage j and state k.
$\mathrm{DR}(j,k,u,s)$	$\equiv$	The amount of delay experienced at signal s that is red as a result of applying control decision u, given decision stage j and state k.
$\mathrm{DG}(j,k,u,s)$	$\equiv$	The amount of delay experienced at the signal s that is green as a result of applying control decision u, given decision stage j and state k.
$\mathrm{DGG}(j,k,u,s)$	$\equiv$	The amount of delay experienced at the signal s that is green as a result of applying control decision u, given decision stage j and state k.
$\mathrm{DGR}(j,k,u,s)$	$\equiv$	The amount of delay wrongfully attributed to the signal s in the previous decision stage for a given control decision u, decision stage j and state k.
w_s	$\equiv$	Weighting factor that can be applied to favor a signal s.

Each vehicle at an intersection can contribute to delay when the light is red, green, or yellow. Any vehicle stopped at a red light contributes an amount of delay that is equal to the duration of the red phase. In addition, delay can be accumulated by vehicles at a green light. E.g., for long (standing) queues, only a small portion of the front of the queue will be able to clear the intersection when the light turns green. Delay is incurred even to vehicles that are close enough to the intersection to pass through. The latter type of delay accounts for the minimum amount of time required for the vehicle immediately in front to respond and move through the intersection.

The total delay $F(j,k,u)$ consists of several components. It is convenient to classify the vehicles according to whether they arrive in the time interval $\left[g^{\text{start}}_{j,k,s}, k\right\rangle$ or whether they are in the queue at the beginning of the time interval:

The total delay for an intersection that results from applying control decision (u) at state (k) can be determined by summing the total delay for each signal:

$$F(j,k,u) = \sum_{s\in S} \mathrm{D}(j,k,u,s) \tag{4.19}$$

where $\mathrm{D}(j,k,u,s)$ denotes the total delay per signal group and is determined differently for signals that are red ($s \notin B_{(b+j-1) \mod (|B|)}$) and signals that are green ($s \in B_{(b+j-1) \mod (|B|)}$):

$$\mathrm{D}(j,k,u,s) = w_s \cdot \begin{cases} \mathrm{DR}(j,k,u,s) & \text{if } s \notin B_{(b+j-1) \mod (|B|)} \\ \mathrm{DG}(j,k,u,s) & \text{if } s \in B_{(b+j-1) \mod (|B|)} \end{cases} \tag{4.20}$$

In calculating the delay for a signal that is red ($\mathrm{DR}(j,k,u,s)$), we determine the delay for each vehicle that is queued at the end of the block ($A(j,k,u,s)$):

$$\mathrm{DR}(j,k,u,s) = \sum_{a\in A(j,k,u,s)} \min\{u, k-a\}) \tag{4.21}$$

To determine the delay for a signal that is allowed to turn green ($\mathrm{SGG}(j,k,u,s)$), we determine the delay for each vehicle. The delay for a vehicle is calculated differently depending on whether the vehicle:

- is able to leave ($k \leq R(j,k,u,s,m)$), and
- was queued at the start of the green phase ($a_m \leq g^{\text{start}}_{j,k,s}$).

The delay for a signal that is allowed to turn green ($\mathrm{DGG}(j,k,u,s)$) is calculated as a summation of the delay for each vehicle as follows:

$$\mathrm{DGG}(j,k,u,s) = \sum_{a_m\in A(j,k,u,s)} \begin{cases} k - g^{\text{start}}_{j,k,s} & \text{if } a_m \leq g^{\text{start}}_{j,k,s} \wedge k \leq R(j,k,u,s,m) \\ R(j,k,u,s,m) - g^{\text{start}}_{j,k,s} & \text{if } a_m \leq g^{\text{start}}_{j,k,s} \wedge k > R(j,k,u,s,m) \\ k - a_m & \text{if } a_m > g^{\text{start}}_{j,k,s} \wedge k \leq R(j,k,u,s,m) \\ R(j,k,u,s,m) - a_m & \text{if } a_m > g^{\text{start}}_{j,k,s} \wedge k > R(j,k,u,s,m) \\ 0 & \text{otherwise} \end{cases} \tag{4.22}$$

As green for a signal can start earlier than the start of the block (i.e. if the conflicting signals of the previous block have cleared earlier than the previous block ends), we have to compensate for any delay that was unjustly attributed to the previous block. The delay ($\mathrm{DGR}(j,k,u,s)$) that was unjustly attributed to the previous block can be calculated as follows:

$$\mathrm{DGR}(j,k,u,s) = \sum_{a \in A(j,k,u,s)} \begin{cases} (k-u) - \max\{\mathrm{GS}(j,k,u,s),a\} & \text{if } s \notin B_{(b+j-2) \mod (|B|)} \\ (k-u) - \max\{\mathrm{GE}(j,k,u,s),a\} & \text{if } s \in B_{(b+j-2) \mod (|B|)} \end{cases} \tag{4.23}$$

where $k-u$ denotes the end of the previous block

This compensation term ($\mathrm{DGR}(j,k,u,s)$) is subtracted from the actual delay (*DGG*) for the signal to determine the additional delay caused by the green signal ($\mathrm{DG}(j,k,u,s)$):

$$\mathrm{DG}(j,k,u,s) = \mathrm{DGG}(j,k,u,s) - \mathrm{DGR}(j,k,u,s) \tag{4.24}$$

4.2.4 Constraints

While optimizing the performance of an intersection certain constraints should be respected. The algorithm ensures that constraints are respected with respect to:

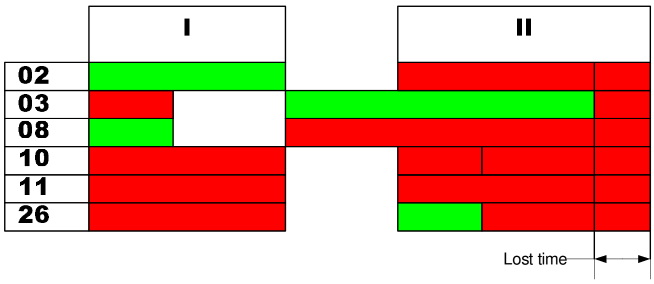
- Minimum and maximum green and red times,
- Protection of dilemma, option, and comfort zones,
- Prior commitments,
- Block skipping and termination in the presence of demand,
- Maximum allowable queue lengths.

The remainder of this section explains how all of these constraints are handled by the algorithm.

Minimum and maximum green and red times Blocks that violate minimum and maximum green time constraints are pruned (together with their corresponding subtree) as soon as the constraint violation can be determined. In case of a minimum green time violation this can be determined before the performance of the block is calculated based on the start time of green for each signal and the time the block to be evaluated ends. This is depicted in Figure 4.13(a). As the length of a block does not determine the length of the signals the block is comprised of, the block size itself cannot violate a maximum green constraint. However, the block size can be chosen inefficiently, e.g., when the end of the block ends after the moment the last green signal of the block has switched to red. These inefficient block sizes are also pruned (together with their corresponding subtrees) from the search space. This is depicted in Figure 4.13(b).

Thus far it has been assumed that when a signal group is allowed to advance to green that it does so at the earliest possible moment (i.e., after the time needed to clear the intersection has passed). For a traffic-actuated controller this is the only option available as it is not able to look ahead in time. A look-ahead controller however is capable to determine a short-term decision on the basis of a long-time horizon. As such, for a look-ahead controller, other options are available. Two of these options are illustrated in Figure 4.14, which each result in a different performance. Assume in both cases that the maximum green time of the signal groups is limited to allow a maximum of three vehicles to pass per green phase.

(a) First option for determining block performance



(b) Second option for determining block performance

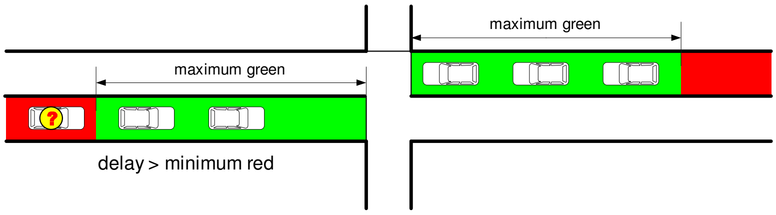
Figure 4.13: Possible options for determining the performance of a block

Figure 4.14(a) illustrates the option where the signal groups switch to green as soon as the first vehicle is predicted to reach the stop line. This mimics the standard behavior of a traffic-actuated controller. The result of this behavior is that the last vehicle predicted to arrive will be delayed for at least the minimum red time of the signal.

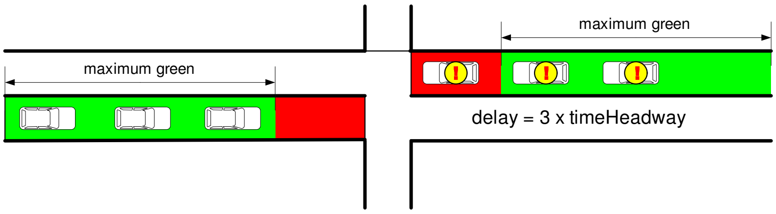
In Figure 4.14(b) the switch to green is delayed such that the end of green aligns with the end of the block. In this case the result is that all vehicles are able to pass the stop line. The result of this behavior is that the three vehicles on the western approach are each delayed for a time that is approximately equal to the time headway with which vehicles can pass the stop line. Both options can be implemented in a look-ahead algorithm. However, the clustering of signals that is necessary to reduce the search space, does not allow to determine the optimal start time of green in real-time.

As a compromise the algorithm switches the signal to green as soon as the first vehicle is predicted to arrive at the stop line, but extends the green phase beyond the maximum green time if this is allowed by future conditions. This is similar to the traffic-actuated control tactic of parallel and waiting green. The difference is however, that in look-ahead adaptive control it can be determined how long the signal can be extended without impeding other signals. This is depicted in Figure 4.15. This is not the case for traffic-actuated control and this is one of the main reasons why parallel green is not enabled by default in the RWSC-structure [28].

The minimum red time for a signal group is incorporated in the calculation of the time



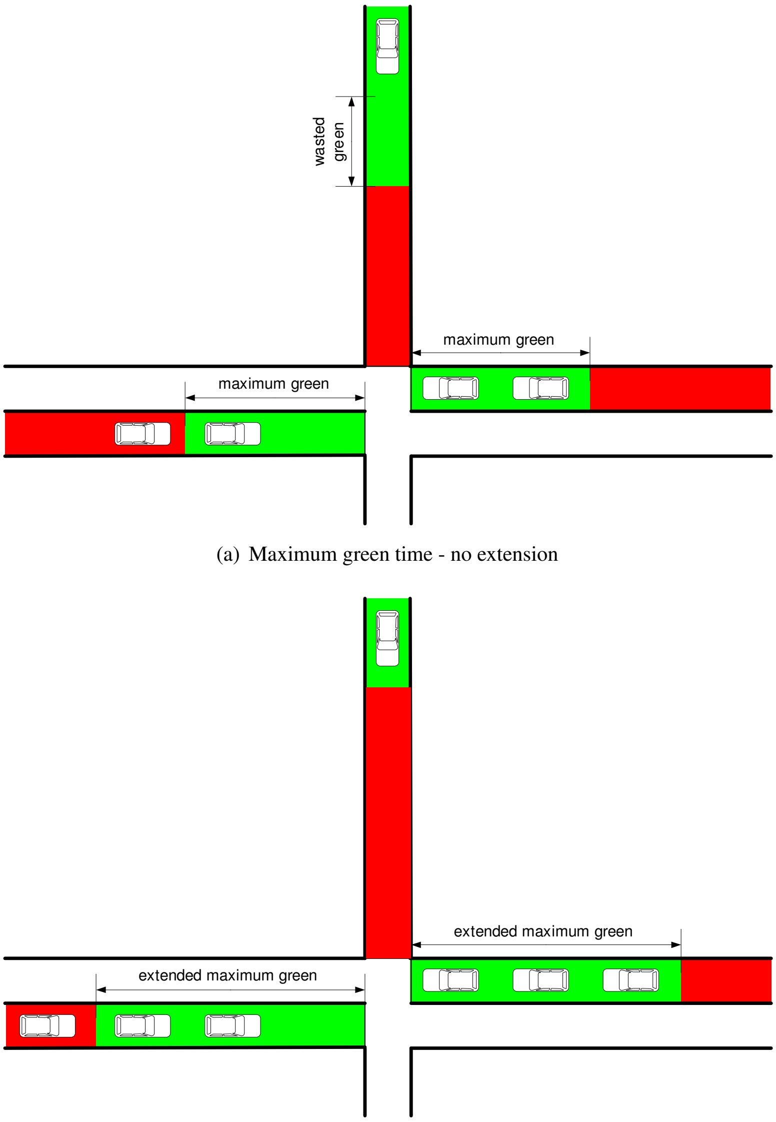
(a) Maximum green time - option 1



(b) Maximum green time - option 2

Figure 4.14: Possible options for determining the green interval

that the green phase can start for that signal group. When the green phase is unable to start before the block ends the block is pruned. So no explicit pruning rule is implemented to prune a block that violates the minimum red time constraint as the block is already pruned on the basis of a minimum green time violation.



(a) Maximum green time - no extension

(b) Maximum green time - extension

Figure 4.15: Extended green time

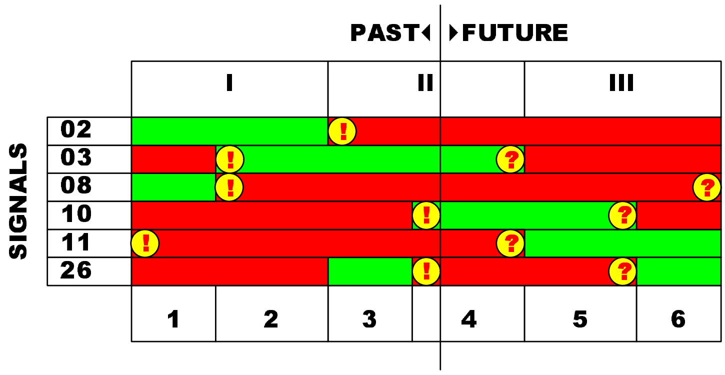


Figure 4.16: Prior commitments

Prior commitments Each time the horizon is "rolled" the control policy is freshly optimized over the optimization horizon. However, the optimization does not start with a clean slate. Past optimizations may have led to prior commitments, which must be respected in order to ensure that the constraints on minimum and maximum green times are respected. This is illustrated in Figure 4.16. In order to determine whether green times can end (denoted by the exclamation marks) the start of green for these signals must be retained (denoted by the question marks). Also, in the calculation of the performance function, it is important to know the departure time of the last vehicle that passed the stop line, in order to respect the minimum headway between the last departing vehicle and the next vehicle to be released.

Dilemma, option, and comfort zone protection The dilemma zone is the zone in which vehicles that are faced with yellow are both unable to pass and to come to a full stop on the start of red. Usually this zone is protected by appropriately adapting the yellow time or the maximum speed with which vehicles are allowed to cross the intersection. In The Netherlands rules and regulations aim to ensure that vehicles will not be trapped in this zone. However, this is only the case when it can be assumed that vehicles obey the legal speed limits. If not, there is still a chance that drivers are trapped in the dilemma zone, causing an unsafe situation for vehicles entering the intersection on conflicting signal groups. In order to mitigate this situation the all-red time for conflicting signal groups should be extended as it is almost certain that the vehicle will enter the intersection during red. For vehicles for which the speed and position can be accurately determined (i.e., through prediction or measurement) the algorithm determines whether the vehicle is stuck in the dilemma zone. If it is the all-red time for conflicting signals is extended. Another possibility would be to disallow the termination of the green phase. The latter is however unacceptable as this would reward the speeding vehicle for its actions.

The option zone, i.e., the zone in which a vehicle faced with yellow can both pass and come to a full stop on the start of red, can be protected to prevent that a rear-end collision occurs between a following vehicle that decides to continue and a leading vehicle that has decided to come to a stop. In The Netherlands a detector is often placed to determine whether more than two vehicles are present in this zone. The location of this detector is determined for a given maximum speed and yellow-time duration. The protection of this zone can be handled as a soft or a hard constraint in the optimization algorithm, by either

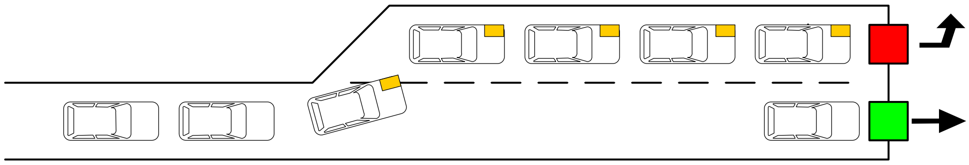


Figure 4.17: Demand for left turn pocket blocks through traffic

applying a penalty each time this situation occurs or by pruning the search space. The application of a penalty is performed by adding a user-defined cost in cases where the option zone is violated.

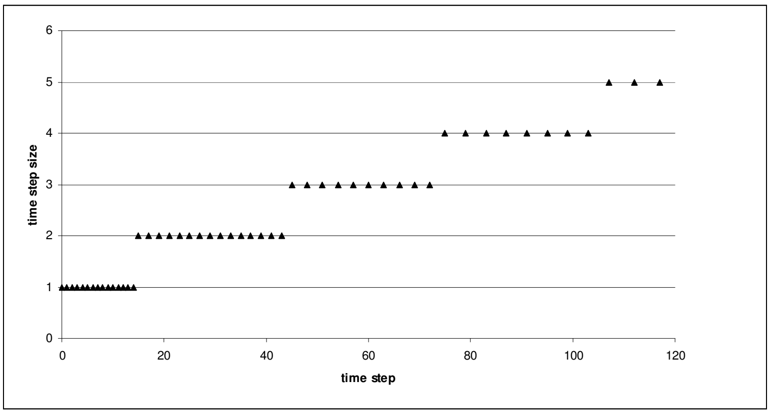
The comfort zone is similar to the option zone, but uses different values for the maximum possible deceleration of a vehicle. The comfort zone uses the maximum possible deceleration that is still perceived as comfortable instead of the minimum deceleration that is legally required. This zone can also be "protected" by the algorithm. However, as the focus is less on safety and more on comfort, this zone will usually be protected by a soft constraint and violations of the comfort zone will be penalized significantly less than a violation of the option zone.

Block skipping in the presence of demand Although arbitrary block sequencing can lead to significant savings in delay, arbitrary block sequencing is often disallowed for safety reasons. The algorithm therefore allows to constrain block skipping for certain blocks to only those cases when the signal groups that make up the block are without demand when the previous block ends. Similarly, the termination of a block for which one of the signal groups is still saturated, can be disallowed by the algorithm.

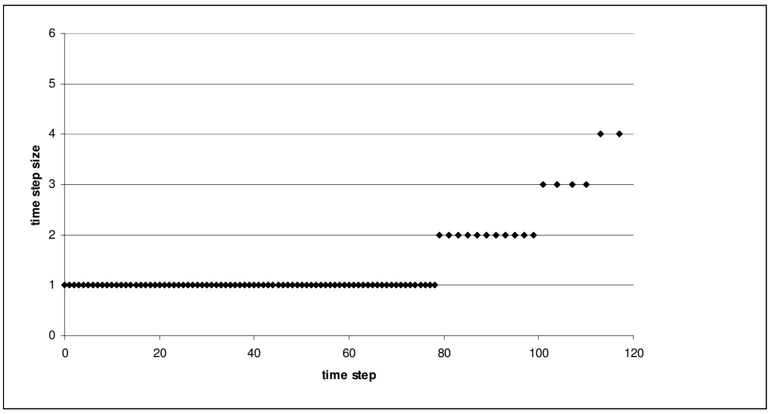
Maximum queue length Many intersection do not have separate approach lanes for each signal group. Instead the approach link flares some distance before the stop line to create a so called pocket for turning traffic. In order to prevent that turning traffic blocks the movement of traffic for the other signal groups a constraint can be put on the maximum queue length allowed for the approach lane. This is depicted in Figure 4.17

4.2.5 Horizon

In order to be able to cope with longer planning horizons the algorithm is capable of using a planning horizon with a variable resolution. This facility was implemented to further speed-up the optimization process (e.g., to enable even longer planning horizons). The algorithm facilitates two types of variable resolution horizons: one in which the resolution decreases linearly and one in which the resolution decreases exponentially. Both are illustrated in Figure 4.18. The figures show how the resolution of a 120-second horizon decreases over time from a 1 to a 5 seconds decision interval by plotting the moments for which the optimization algorithm evaluates the possible control decisions and the time period for which the optimization algorithm subsequently commits to these decisions (the size of the time step). The variable resolution horizons use a higher resolution horizon for the near future than for the more distant future. The number of decision points that remain after the application of a variable resolution horizon and the corresponding reduction of the search space



(a) linearly increasing step size



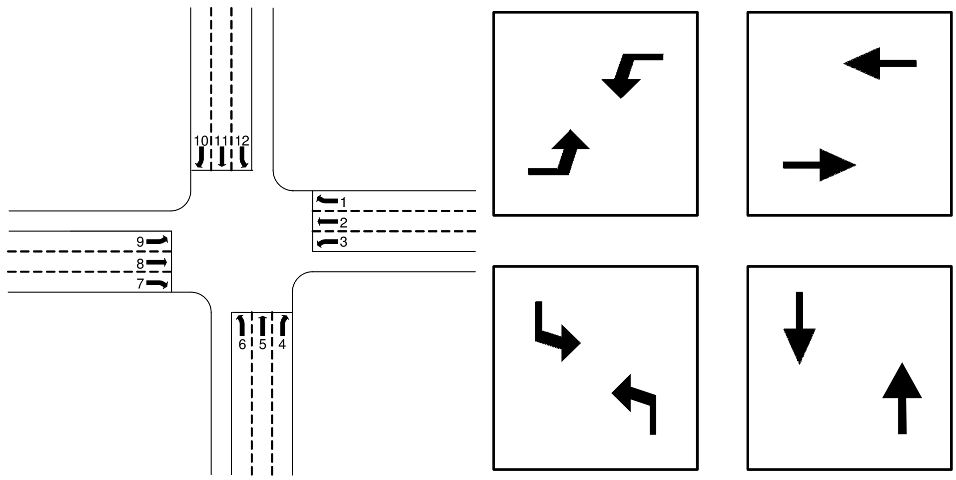
(b) exponentially increasing step size

Figure 4.18: Variable time step sizes for the optimization horizon

	Number	Search Space
Fixed	120	$\|B\|^{120}$
Linear	52	$\|B\|^{52}$
Exponential	97	$\|B\|^{97}$

Table 4.4: Search space reduction as a result of a horizon with a variable resolution

is illustrated in Table 4.4. As the estimated arrival time for arrivals expected from further upstream is more susceptible to change, the impact of the lower resolution at the end of the horizon on the quality of the signal plan is limited.



(a) Geometry (b) Blocks

Figure 4.19: Features of the intersection used in the simulation

4.3 Analysis and results

In order to test the performance of the developed algorithm the traffic management testbed described in Appendix A and published in [113–116] was used. This test-bed enabled us to interface the adaptive control algorithm of Section 4.2 with the Paramics simulation model developed by Quadstone [93] and the AIMSUN simulation model developed by TSS [108].

4.3.1 Scenario

The simulations were performed for the 4-arm intersection depicted in Figure 4.19(a). Each of the twelve possible movements on the intersection has a separate, single, approach lane. The total demand for the intersection is set to 4400 vehicles per hour, which is distributed over the movements in proportion to the saturation flow rate of each movement. Maximum green times for each stage and block were subsequently determined according to Webster's method [121]. The blocks are shown in Figure 4.19(b). For simplicity, the right turns are omitted and assumed to proceed with the through movements.

4.3.2 Results

The results have been obtained from a number of one-hour simulations each with a different random seed and are displayed in Figures 4.20 and 4.21. The figures show the average delay per vehicle as it evolves over a one hour period. As each simulation starts with an empty network some time is needed before vehicles start to arrive at the intersection and the intersection becomes fully stressed. This is the so called warm-up period of a simulation. We can be sure the warm-up period has ended when the curve of the average vehicle delay stabilizes into a line that is more-or-less horizontal.

In Figure 4.20 the average delay of a vehicle is shown when the intersection is operating under *stage-based* (a) traffic-actuated control and (b) look-ahead traffic-adaptive control.

The average delay encountered by a vehicle in the traffic-actuated controlled case stabilizes at about 25 seconds per vehicle (see Figure 4.20(a)) whereas the average delay encountered by a vehicle in the traffic-adaptive controlled case stabilizes at about 18 seconds per vehicle (see Figure 4.20(b)). This proves that planning for future arrivals can substantially improve the performance of an intersection.

In Figure 4.21 the average delay encountered by a vehicle is plotted for a *movement-based* look-ahead traffic-adaptive controller. This allows for a total of eight phases (i.e., the number of signal group configurations supported by a typical 2-ring NEMA traffic-actuated controller) instead of the four phases allowed for by stage-based control. The average delay has reduced even further to about 16 seconds per vehicle (see Figure 4.21(a)). Note that this reduction in delay can be obtained without significantly increasing the number of computations as the movement-based look-ahead traffic-adaptive control algorithm developed in this thesis allows to evaluate more phases without increasing the size of the search space.

When however computational complexity demands a longer update-interval (e.g., 5 seconds) the average delay per vehicle rises again to about 18 seconds per vehicle (see Figure 4.21(b)). This clearly shows the impact of the trade-off made regarding the choice of the decision space, planning horizon, update frequency and delay model. The better the optimization method employed is, the less comprises have to be made.

4.4 Conclusions

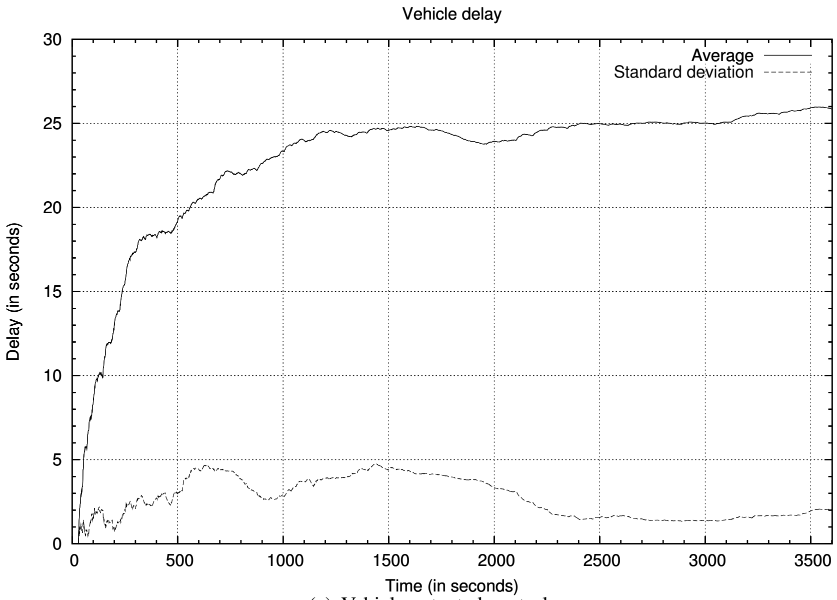
In this chapter a new algorithm has been presented for the look-ahead adaptive control of an intersection. Decisions made by a look-ahead traffic-adaptive control algorithm are taken on the basis of a longer term analysis. This longer term analysis implies the search for an optimal sequence of control actions in a search tree that is formed by the selection of control actions over a time horizon. To be able to efficiently search this search tree for longer optimization horizons the algorithm presented incorporates the following innovations with respect to the state of the art:

- The algorithm integrates the currently best known dynamic programming optimization approach to look-ahead traffic-adaptive control with a branch-and-bound type optimization.
- The algorithm applies the more flexible movement-based approach to traffic signal control as opposed to the stage-based approach employed by the current state of the art in look-ahead adaptive control. This enables the algorithm to analyze a larger number of possible signal timings without further increasing the size of the search space.

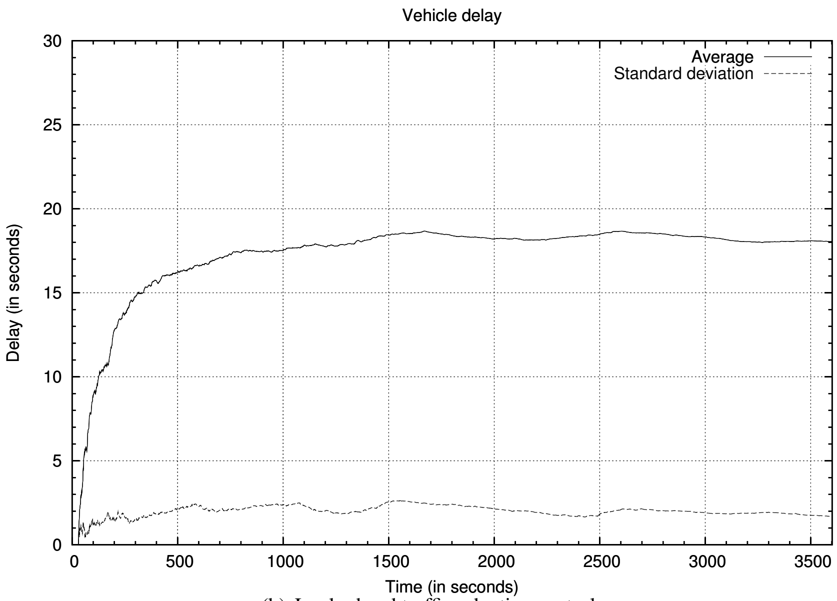
The computation time gained through use of the developed algorithm compared to the state of the art in look-ahead traffic-adaptive search algorithms can be used to either further:

- extend the planning horizon of the traffic-adaptive controller,
- increase the update frequency of the algorithm,
- increase the resolution of the planning horizon, or
- use more accurate models to determine the cost of applying a control decision.

A topic for further research is to ascertain which (combination) of these possibilities contributes most to the performance of the intersection. In this thesis the computation time gained is used to enable controllers to iteratively coordinate their decisions with nearby controllers. This coordination mechanism will be presented in the Chapter 5.

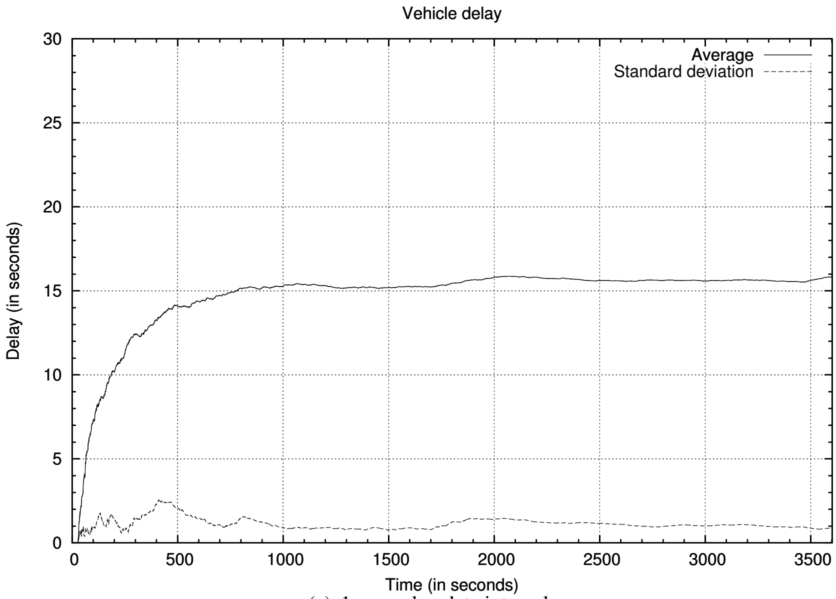


(a) Vehicle-actuated control

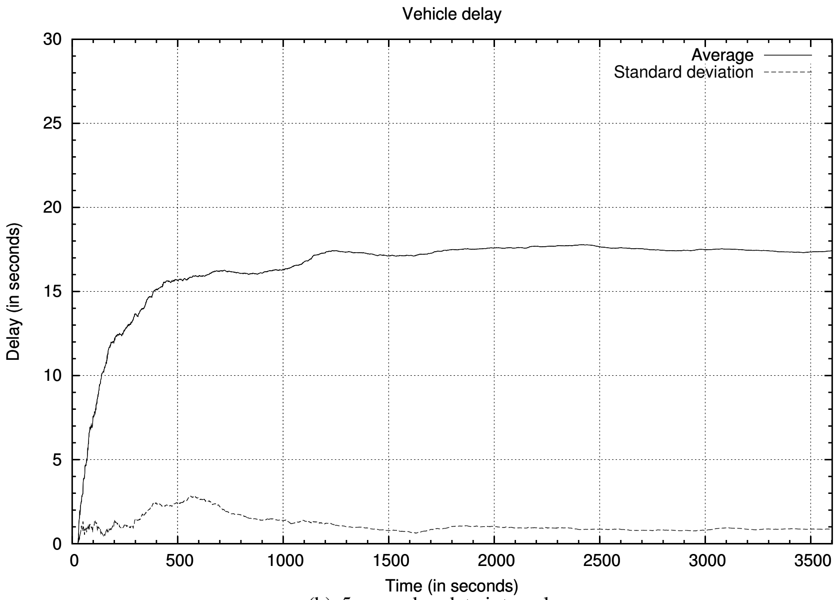


(b) Look-ahead traffic-adaptive control

Figure 4.20: Performance for an intersection with 4 phases



(a) 1-second update interval



(b) 5-second update interval

Figure 4.21: Performance for an intersection with 8-phase traffic-adaptive control

Chapter 5

Area adaptive control

The aim of this chapter is to tie together the individual traffic control measures into a cooperative system that ensures that the control actions of the individual control instruments are coordinated. Section 5.1 further motivates why the coordination between control instruments is becoming more important. In Section 5.2, the approach chosen to enable the cooperation between control instruments is introduced. As control instruments operate on different levels, the procedures used to coordinate traffic control instruments are also different. The procedure used to coordinate control measures on a microscopic level (i.e., on the level of individual vehicle arrivals) is described in Section 5.3. Subsequently, in Section 5.4 the procedure used to coordinate control measures on the macroscopic level (i.e., on the level of flows) is described. Section 5.5 concludes this chapter and summarizes the most important results of this chapter.

5.1 Motivation

Since the early eighties, the Dutch Ministry of Transport has installed electronic systems to improve traffic flow and safety on the motorway network, including an extensive "Motorway Signaling System" [58, 60] (for autonomous speed control and lane closure commands), ramp metering systems and Dynamic Route Information Panels. Most traffic control instruments that are installed along the world's freeways have been tuned individually to attain a rigid local objective. As more and more instruments are deployed, chances are that instruments will be deployed in each other's region of influence, resulting in interference whenever the actions of the individual instruments are not coordinated. Meanwhile different developments are being conducted that will further increase the number of traffic control instruments available to the traffic operator.

In order for a traffic operator to focus on the management aspects of traffic control and to ensure the timely coordination of the traffic control instruments it is desirable that certain coordination tasks are dealt with automatically whenever possible. By allowing the individual instruments to coordinate their actions based on the information they receive from sensors and each other, traffic control instruments can be coordinated more often and more accurate than can be done by a traffic operator.

In the literature many examples exist where the answer to the dynamic traffic control problem is sought in the form of a traffic control center that monitors the traffic network and performs a global, or area-wide, optimization to set up new parameters for its local controllers. Much of this work has focused on centralized, and typically predictive, control [20]. Although this approach is very appealing it just is not always possible to do this efficiently and effectively, which is largely due to the amount of data involved and the computational complexity of the problem.

One way to more effectively handle contingencies is to make the traffic control instruments more intelligent and have them deal with the intricacies of configuring the traffic control instruments to the situation at hand. The traffic operator can then focus on the direction of traffic over the network, since a part of the problem is dealt with by the traffic control instruments.

5.2 Approach

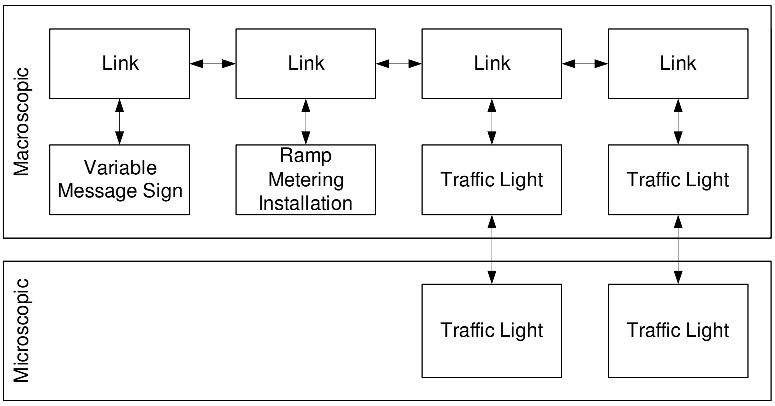
The approach chosen in this thesis is to handle the coordination of the actions of the individual traffic control instruments distributedly. In [23, 38, 39, 57, 61, 67, 70, 112] it is argued that multi-agent systems can aid in the distribution of the problem and facilitate the coordination of the activities of the traffic control instruments when needed. The term agent is used to denote an intelligent actor, which observe and act upon an environment through sensors and actuators, whereas a multi-agent system is a system composed of several agents, collectively capable of reaching goals that are difficult to achieve by an individual agent or monolithic system.

Agent technology can make an important contribution to traffic management since the abstract concepts and ideas as used in the domain of agents and multi agent systems so easily translate to the traffic management domain. Most multi-agent systems are explicitly designed so that cooperation between the individual agents is obvious. This expertise with the principles of coordination in a heterogeneous environment of agents in general is directly beneficial to the development of these principles for traffic management in particular.

The approach chosen by this thesis is to represent each infrastructure element in the network by an agent. Each agent is made responsible for the performance of the link in the network it represents. An infrastructure-centric approach, as opposed to a controller-centric approach, is chosen because the former abstracts more easily to the network control objectives as used by a traffic manager. The agents provide the controllers with the necessary information in order to maximize the link's performance. This information is different for each type of controller.

The multi-agent system that is thus formed, forms a complete "shadow" network of the real network. Although each agent has a large degree of autonomy it is made susceptible to the interests of its neighbors. It therefore coordinates its actions with its neighboring agents. This coordination is obtained through direct communication and negotiation. As variable message signs and ramp metering installations can only be coordinated on a macroscopic level, these controllers are not provided with any microscopic information. This is different for traffic signals as traffic signals can be coordinated on the microscopic level. Figure 5.1 summarizes the architecture that has been chosen for the multi-agent system.

In Section 5.2.1 the multi-agent concept developed in this thesis is illustrated for a small network. In this section it is illustrated how agents cooperate to solve a local problem in the network by taking appropriate action elsewhere in the network. In Section 5.2.2 the procedure to perform the coordination among agents is introduced.



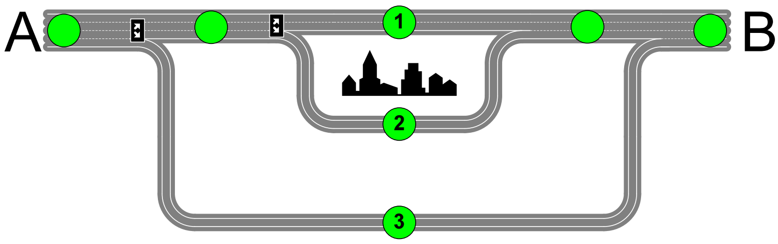


Figure 5.2: Network used to illustrate the approach

5.2.1 Illustrative examples

The multi-agent traffic control concept developed in this thesis will be illustrated throughout this chapter using the network depicted in Figure 5.2. This network is comprised of three separate routes (marked 1, 2, and 3 respectively). Each of these routes will get the road user from location A, which is located on the left side of the page, to location B, which is located on the right side. Under free-flow conditions the topmost route gets road user from A to B the fastest. However, road users can also choose to make use of the alternative routes 2 and 3 if they prefer. The route that the road user will ultimately choose can be influenced through means of the variable message signs that are located at each choice point in the network.

The topmost route (route 1) represents a freeway corridor and is used later on in this chapter to illustrate how the multi-agent traffic control concept can be used to coordinate the ramp metering installations that control access to the freeway. The second route (route 2) represents an urban corridor controlled by traffic signals and is used later on in this chapter to illustrate how the multi-agent traffic control concept can be used to coordinate the traffic signals on an urban corridor. The network itself is used to ultimately illustrate how the multi-agent traffic control concept is able to coordinate all traffic control instruments in the network (ramp metering installations, traffic signals, and variable message signs).

Using the network depicted in Figure 5.2, the benefits of coordinated control can be easily illustrated. Consider the case that an incident occurs on the preferred route (route 1). Because of this incident the preferred route is no longer able to accommodate the traffic demand. Without intervention the queue that will form upstream of the incident location will grow rapidly as more and more vehicles will join the queue. In order to prevent that vehicles join the queue needlessly, e.g., because an alternative route would have been available to them, the agent that represents route 1 informs the upstream agent regarding the remaining capacity on the route, which is illustrated in Figure 5.3.

The agent upstream of route 1 represents the link at which traffic can choose from multiple alternative routes. This link agent is also informed by the agent representing the downstream alternative (in this case route 2), regarding any spare capacity available that can be used to to accommodate the surplus of traffic. This is illustrated in Figure 5.4. If spare capacity is available the node-agent can inform approaching traffic about the available alternative. This way traffic that at first wanted to make use of the preferred route (route 1), can be partly redirected to the alternative route (route 2).

When the available spare capacity is still insufficient to accommodate the surplus of traffic and route 2 as such is able to take care of only part of the problem, a solution must be found further upstream for the remaining part. This is again done by informing upstream agents regarding the remaining downstream capacity (illustrated in Figure 5.5).

As soon as this information reaches the agent that represents the link at which traffic can again choose from alternative routes a solution is again sought at the downstream alternative (in this case route 3). This is illustrated in Figure 5.6. If the spare capacity on route 3 is sufficient to take care of the remaining surplus of traffic, then a further escalation of the problem can be prevented. Only traffic that has already made the choice for route 1 will contribute to the further growth of the queue. If it is not known in advance when and where a capacity reduction will occur, the formation of queues cannot be prevented.

Although the concept of multi-agent traffic control is easily illustrated the coordination of traffic control instruments is not a simple task. Fortunately the agents have one common goal and are thus cooperative. To achieve coordination between the agents we developed an iterative coordination procedure, which will be described in the next section.

5.2.2 Coordination procedure

In this section the procedure that is used to coordinate the agents is introduced. This coordination procedure is subsequently further elaborated in Section 5.3 and Section 5.4 for coordination on the microscopic and macroscopic level respectively. The coordination procedure assumes that each of the agents functions fully autonomously and tries to maximize its own, local, performance.

At each time step each agent determines its current state. It does so on the basis of the information it gathers from local detectors and from information received from upstream and downstream agents. At each time step each agent tries to optimize its performance by regulating its inflow and outflow. Requests are made to downstream agents by upstream agents concerning their intended outflow. When such a request is received the downstream agent determines the impact of the intended inflow on its performance. This impact, expressed in the form of a cost is communicated to the upstream agent that made the request. The upstream agents can than decide whether the downstream costs outweigh their own costs, and make a decision regarding the outflow they want to realize. If downstream costs are such that they do not outweigh the own costs, the performance of the agent can only be

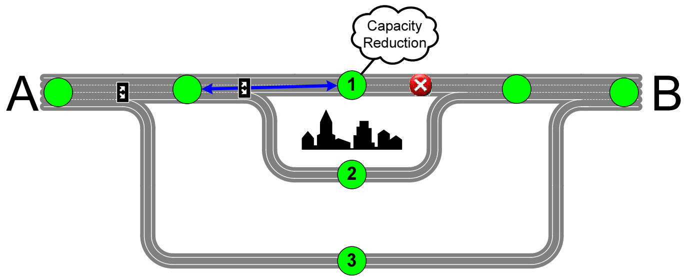


Figure 5.3: A problem . . .

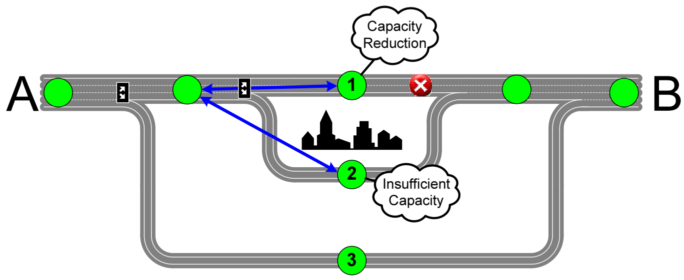


Figure 5.4: An alternative is sought . . .

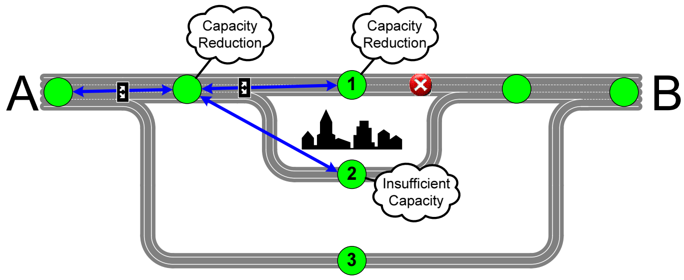


Figure 5.5: The first trial fails . . .

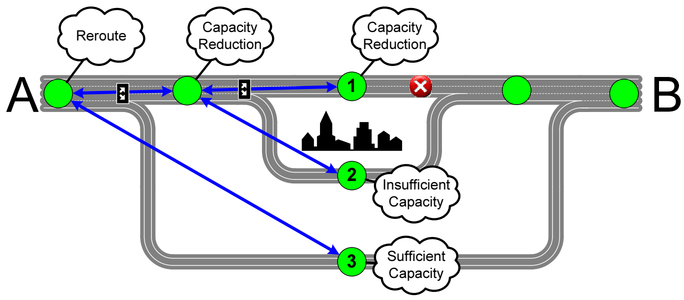


Figure 5.6: The second trial succeeds.

regulated by manipulating the inflow. This is done by incorporating the downstream outflow costs into the inflow cost for upstream agents.

As not all links are equipped with traffic control instruments, agents more often than not have no other means of control than to increase the cost for inflowing traffic for upstream links. Problems can therefore often not be dealt with close to the source of the problem, but can only be dealt with by actuators elsewhere in the network. As contributions to the solution of the problem are found elsewhere in the network the time delay between the moment the contribution is made and the moment this is felt at the problem location becomes important. When the source of a problem has been resolved, it will usually take some time before the consequences of a problem have been resolved as well. In general, it is therefore not a problem that the effects of the deployed measures persist a while longer although the actual cause of the problem has been resolved.

The procedure described is used to coordinate the actions of the traffic control instruments in the network on both a macroscopic and the microscopic level and is summarized using the following high-level procedure:

Algorithm 5.1 High-level coordination procedure

```
1: loop
2:   while NOT (equilibrium or cycling) do
3:     update current state
4:     determine optimal control settings incorporating (a) downstream cost and (b) ex-
       pected upstream inflow
5:     for all downstream agents do
6:       send intended outflow
7:       receive downstream cost of planned outflow
8:     end for
9:     for all upstream agents do
10:      receive planned inflow
11:      determine cost of upstream inflow given local and downstream cost
12:      send cost of upstream inflow
13:    end for
14:  end while
15: end loop
```

In steps 3–4 of the procedure the agent tries to optimize its local performance. Coordination with neighboring agents takes place in steps 5–8 for downstream agents and in steps 9–10 for upstream agents. The coordination procedure is repeated until the agent no longer wishes to update its control settings. The next section further elaborates this procedure for the coordination of traffic signals on a microscopic level.

5.3 Microscopic coordination

In this section the algorithm developed to perform coordination on the microscopic level is introduced. This section starts out in Section 5.3.1 by motivating the need for coordination on a microscopic level and an overview of previous work done on this subject. Subsequently the algorithm developed to perform coordination on a a microscopic level is introduced in

Section 5.3.2. The working of the algorithm is subsequently illustrated in Section 5.3.3. Section 5.3.4 describes the results from the analysis of the algorithm. This section ends with some discussion in Section 5.3.5.

5.3.1 Motivation

The idea of coordination between traffic control instruments is very old [82]. In urban areas, with a large number of signalized intersections, traffic engineers often face the task of coordinating multiple subsequent intersections in order to allow platoons of vehicles to move through several signalized intersections without stopping. The movement of a platoon of vehicles through several signalized intersections is referred to as progression. To achieve progression, a timing relationship is developed between successive intersections such that vehicles, traveling at a predetermined speed, can pass through the green indications at successive signals. This limits the flexibility of the system. The system thus created can be compared to the coordinated equivalent of the fixed-time control of a single intersection.

The efficiency of a progression scheme is largely dependent on the platoon ratio in the traffic stream. Platoon ratio is the fraction of arterial traffic that travels from the first intersection through the last intersection in the system. Signal performance will also depend on the amount of traffic on minor movements in two ways: (1) The traffic volume on the cross streets affects the percentage of traffic turning into the main street, and therefore affects the platoon ratio; and (2) a low volume on the minor movements allows the controller to spend more time in green on coordinated movements.

Given the dynamics in traffic volume and platoon ratios a progression scheme is needed that is able to flexibly adapt to changes in volumes and platoon ratios.

5.3.2 Coordination procedure

To come up with a progression scheme that is able to flexibly adapt to changes in volumes and platoon ratio a traffic signal control agent has to be aware of (a) the effect of the actions of agents downstream and upstream on its own performance and (b) the effects of its own actions on the performance of the agents downstream and upstream. The look-ahead traffic-adaptive control algorithm presented in Chapter 4 already allows a traffic signal control agent to incorporate the effect of arrivals that are expected from upstream intersections in its decision making. This can be considered as a first step in creating a cooperative traffic signal control agent. The next step is to make the traffic signal control agent aware of the its effects on other, nearby controllers. The algorithm described in Chapter 4 minimizes a performance function on all local approaches. As such it is not capable to ascertain whether the delay it prevents locally is annulled at the downstream intersection.

In order for the algorithm to be able to ascertain whether delay prevented upstream is annulled at the downstream intersection we extended the algorithm of Chapter 4 to also incorporate the expected downstream performance in its decision making. In order to incorporate the downstream performance information must be shared between intersections regarding downstream conditions. These conditions include the downstream signal plan, residual queues, pending arrivals, and the saturation flows at the downstream intersection. This information is necessary in order to determine the delay a released vehicle will encounter as it approaches the downstream intersection. In order to incorporate the downstream objective the intersection needs the same information for the downstream intersection as is needed in order to optimize the local intersection. The difference is that the signal plan for the downstream intersection can be considered as given, whereas the own signal plan is the result of the optimization. As at first no information from neighboring intersections is available the signal plan is iteratively updated to incorporate the changes at neighboring intersections as they become available.

We will now illustrate the microscopic-coordination procedure and its benefits using some simple examples.

5.3.3 Illustrative examples

In this section the microscopic coordination procedure is illustrated using two simple examples. We start in the next section by illustrating the procedure step by step for an urban corridor. This example is subsequently extended to a grid network.

Illustrative example for a corridor

The benefits and the dynamics of the microscopic coordination procedure can be understood by considering the following example. Figure 5.7 shows a corridor consisting of two intersections. The traffic demands consist of six vehicles, one eastbound approaching the western intersection (1), one westbound approaching the eastern intersection (2) and four northbound approaching the western (3,4) and eastern (5,6) intersections respectively. Black vehicles are used to denote the vehicles that travel from the west to the east or vice versa whereas the white vehicles denote the vehicles that travel from the south to the north page. Potential conflicts are therefore always between vehicles with a different color and never between vehicles of the same color. Notice that vehicles 1 and 6 and vehicles 2 and 4 are at equal distance (d) from respectively the western and the eastern intersection. When two vehicles are predicted to arrive at an intersection and request opposing signal phases, a conflict is said to occur. Two of these conflicts are depicted in the time-space diagram of Figure 5.8. This diagram shows the distance to the downstream intersection(s) of vehicles 1, 3, and 6 as it evolves over time. The dashed lines mark the location of the intersections, whereas the other lines mark the location of vehicles 1,3, and 6 as time progresses. The diagram shows that vehicle 1 has potential conflicts with vehicle 3 at the first downstream intersection and with vehicle 6 at the next downstream intersection. In the time-space diagram these events are marked with a star shape. In total there are four potential conflicts as Figure 5.8 only depicts the conflicts of vehicle 1 with vehicles 3 and 6. The potential conflicts that are not depicted are those of vehicle 2 with vehicles 5 and 4.

The intersection can deal with these potential conflicts in one of two ways, one suboptimal (as shown in Figure 5.9(a)) and one optimal (as shown in Figure 5.9(b)). In the suboptimal case preference is given to the eastbound vehicle (1) over the northbound vehicle (3), whereas in the optimal case preference is given to the northbound vehicle (3) over the eastbound vehicle (1). The vehicles that are not given preference to are stopped. These events are marked in the time-space diagrams by a stop-symbol. From a local perspective both options can be considered equal as from a local perspective not all potential conflicts in the network are visible. As the western intersection is unaware of traffic approaching the eastern intersection it is unaware of the impact of its decision to first release the eastbound vehicle (1). If it releases this vehicle first, it will come in conflict with the second northbound vehicle (6) on the eastern intersection. The coordination procedure can prevent

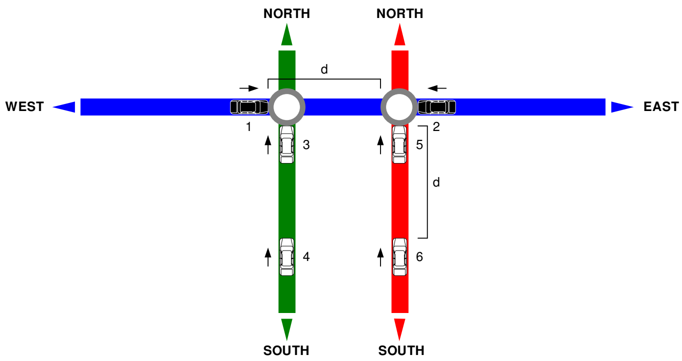


Figure 5.7: Geometry of the corridor

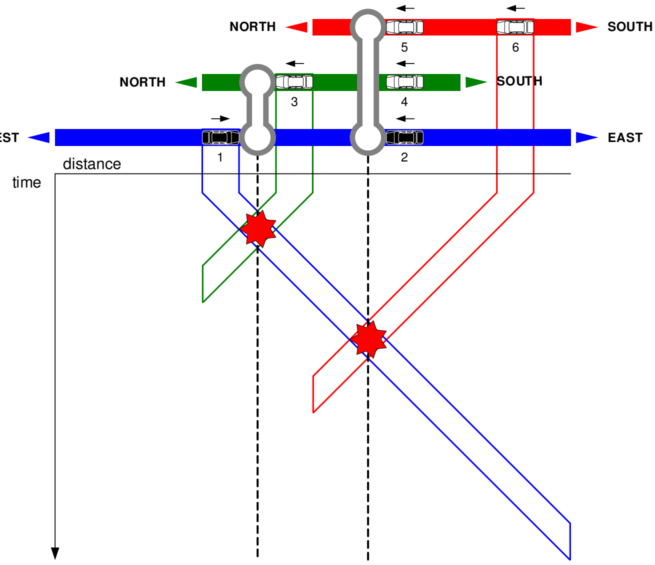
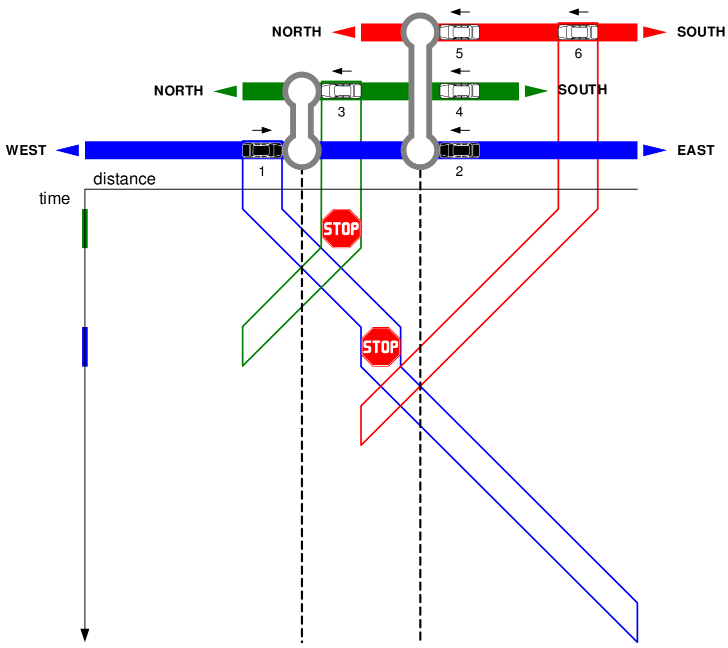
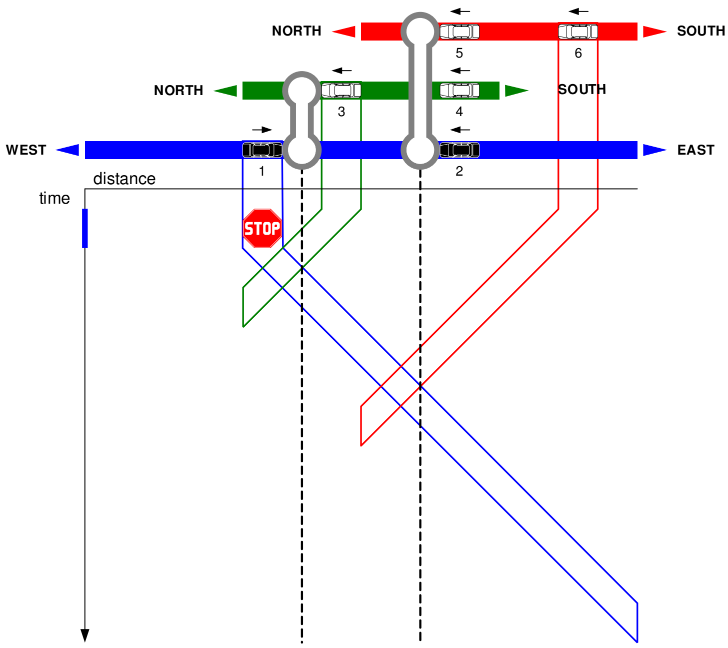


Figure 5.8: Conflicts on the corridor

suboptimal behavior at the local intersection. The process which results from applying the coordination procedure is described next.



(a) Traffic is controlled suboptimally



(b) Traffic is controlled optimally

Figure 5.9: Different ways in which traffic could be handled

Figures 5.10–5.13 illustrate the process that each intersection goes through to come up with an initial signal plan. Vehicles are represented by circles here and the numbers within represent the units of delay the vehicle has encountered as a consequence of the decisions made by the intersection controller. In the first iteration (depicted in Figure 5.10) both controllers are unaware of what the situation is at nearby intersections. As the process of the western intersection mirrors that of the eastern intersection in the remainder only the former will be described in detail. The intersection starts out to *observe* the traffic state on its approaches. The intersection controller determines that there are two vehicles in conflict. These are the two vehicles that are already waiting at the stop line and that request an opposing phase. As the intersection is unaware of the state of neighboring intersection it foresees no conflict for the vehicles approaching the intersection from the south. The choice which vehicle to give the right of way is arbitrary as each choice will bring about the same amount of delay. In this example the intersection *decides* to give the right of way to the black vehicle, resulting in delay for the white vehicle waiting for the stop line (the number in the circle denotes the amount of delay encountered by the vehicle). This iteration ends not by directly *acting* on the choices made, but by informing nearby intersections about the actions intended. If the signal plans resulting from this iteration would be implemented this would result into four stops (although both intersections combined are aware of only two).

The second iteration starts when information regarding nearby intersection's intentions is received. In this iteration the intersection *observes* that its choice for giving the right of way to the black vehicle is suboptimal as the result of this choice is that the vehicle will be stopped at the next intersection. This choice results in two stops (four in total for the two intersections), whereas the decision to give the right of way to the white vehicle leads to only one stop (two in total). It therefore *decides* to give way to the white vehicle. The delay incurred to the black vehicle enables it to pass freely through the next junction. The intersection also *observes* that there is a new conflict between the vehicle that originates from the nearby intersection and the second white vehicle that approaches the intersection. The choice which of these vehicles to give the right of way is arbitrary as each choice will bring about the same amount of delay. In this example the intersection *decides* to give way to the vehicle inbound from the nearby intersection. The second iteration again ends by not directly *acting* on the choices made but instead by informing nearby intersections about its new intentions. If the signal plans resulting from this iteration would be implemented this would result into four stops (of which both intersections are now fully aware).

The third iteration starts as soon as updated information regarding nearby intersection's intentions comes available. This time both intersection stick by their *decision* to give right of way to the white vehicle. However, they also realize that there is no longer a conflict between the vehicle that originates from the nearby intersection and the vehicles approaching the intersection from the south. The vehicles approaching both intersection from the south therefore do not have to be delayed. As changes were again made the third iteration ends by informing nearby intersections about their new intentions. If the signal plans resulting from this iteration would be implemented, this would result into only two stops (of which both intersections are fully aware).

The fourth and final iteration starts as soon as updated information regarding the intentions of nearby intersections becomes available. This time the new information, for both intersections, does not lead to any changes in intentions. The intentions can therefore be acted upon as the process can be said to have converged.

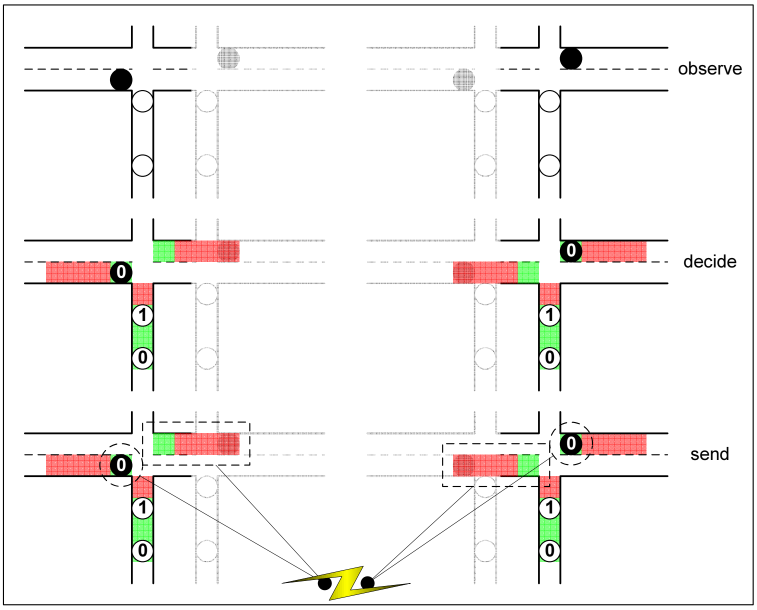


Figure 5.10: Iteration 1

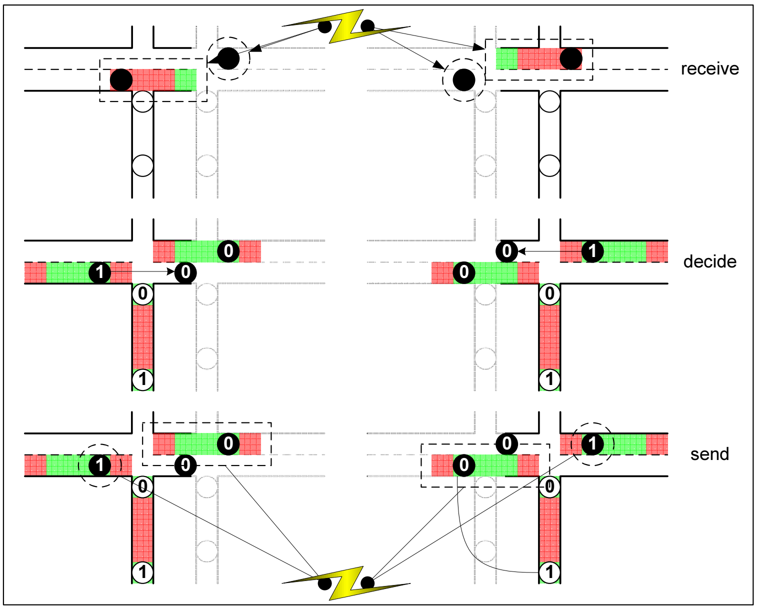


Figure 5.11: Iteration 2

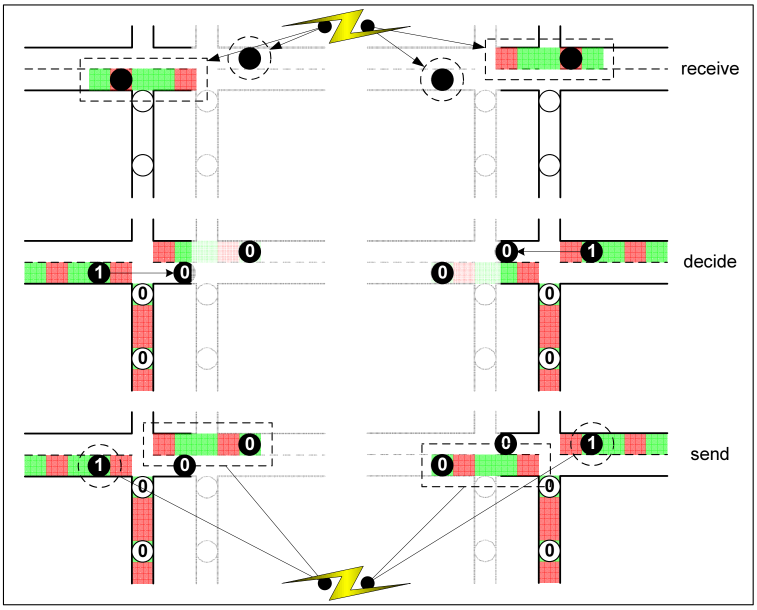


Figure 5.12: Iteration 3

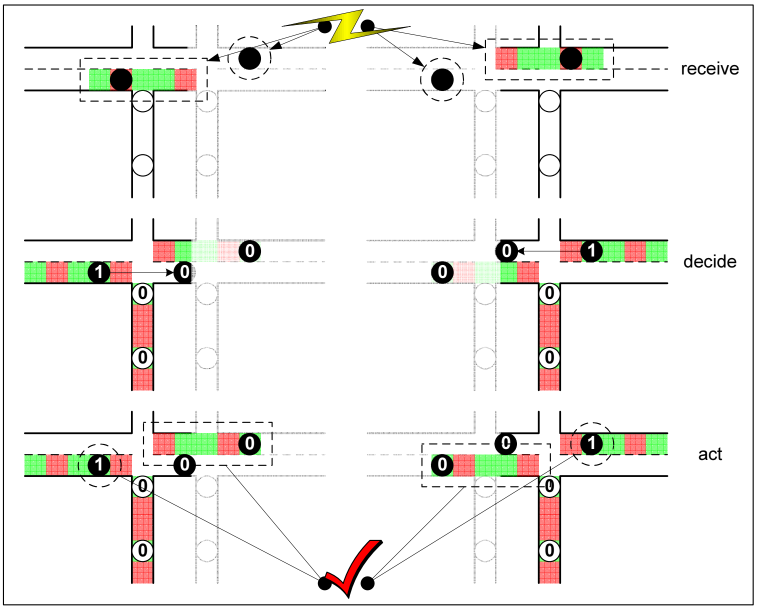


Figure 5.13: Iteration 4

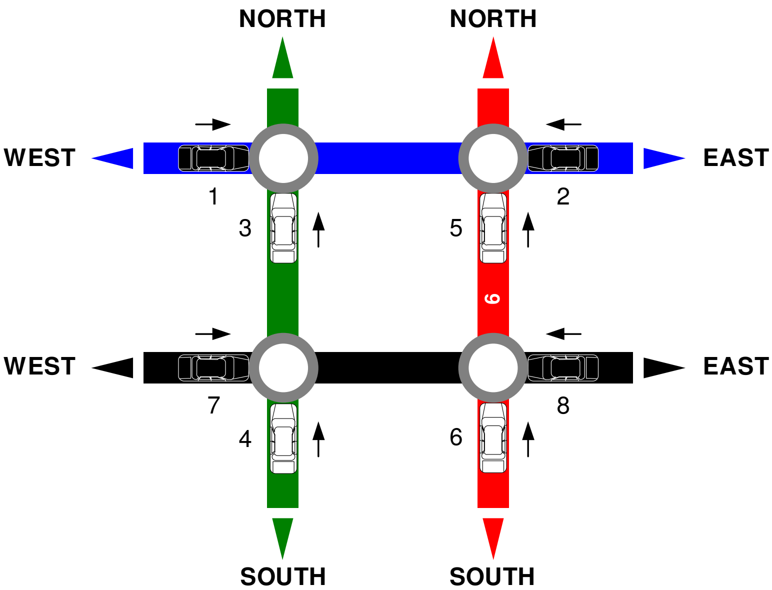


Figure 5.14: Geometry of the network

Illustrative example for a grid network

The example of Figure 5.7 referred to a corridor. The process can however easily be extended to a network as will be shown in the example of Figure 5.14. Figure 5.14 shows a small network consisting of four intersections. The traffic demand consist of six vehicles, one southbound for each of the approaching the northern intersection (1,7), one northbound for each of the southern intersections (2,8) and one westbound on the approach of every intersection (3,4,5,6). When two vehicles are predicted to arrive at an intersection and request opposing signal phases, a conflict is said to occur. Two of these conflicts are depicted in Figure 5.15. These are the potential conflicts for the southbound vehicle of the northwestern intersection (1). In total there are four potential conflicts.

In Figure 5.16 the decision space for the network is shown for the current time step. In total there are 2^4 possible options each resulting in a different number of stopped vehicles. No matter what choice is made, there will always be a minimum of four stops as there is a conflict between vehicles waiting at the stop line for each intersection. The made choice however also determines whether the released vehicles will come into conflict further downstream in the network with other vehicles. In the worst case there are six stops as each vehicle released upstream comes in conflict at the downstream intersection.

In order to avoid conflicts further downstream in the network, intersections on the opposite corners of the network should not simultaneously release vehicles toward the same downstream intersection. These potential conflicts can only occur at the two western intersections in the example. Figure 5.17 clearly illustrates this. In Figure 5.17(a) an example of suboptimal behavior is shown where the intersections on the opposite corners of the network all simultaneously release vehicles toward the same downstream intersections. In Figure 5.17(b) an example of optimal behavior is shown where all intersections on opposite

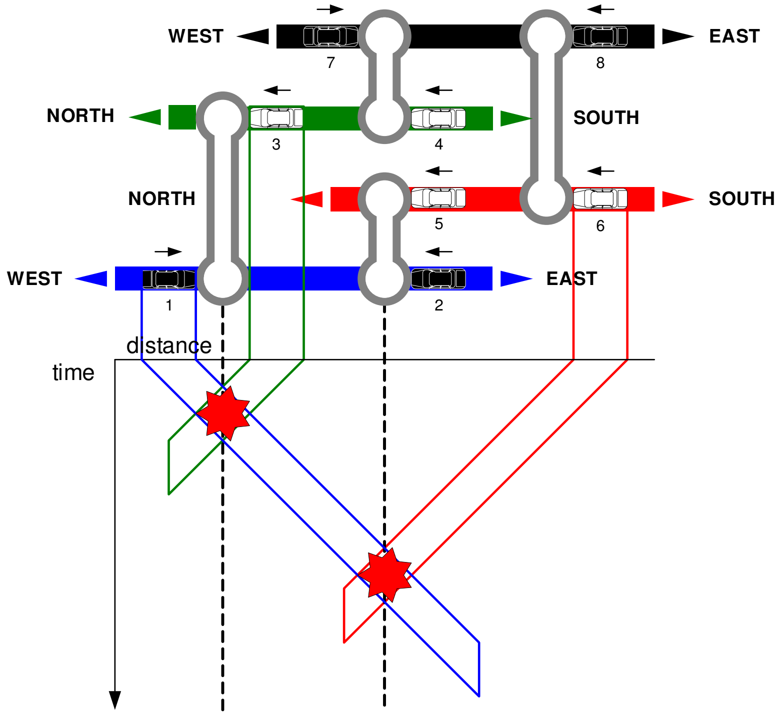
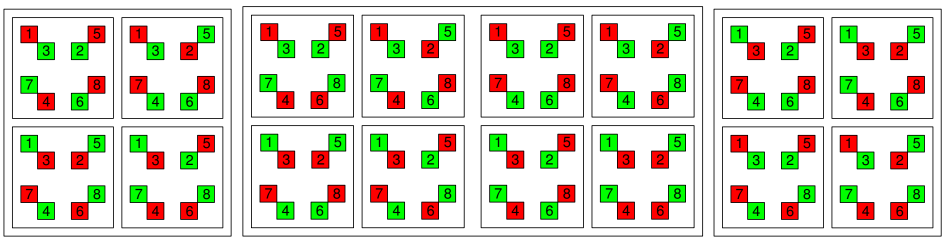


Figure 5.15: Conflicts on the network

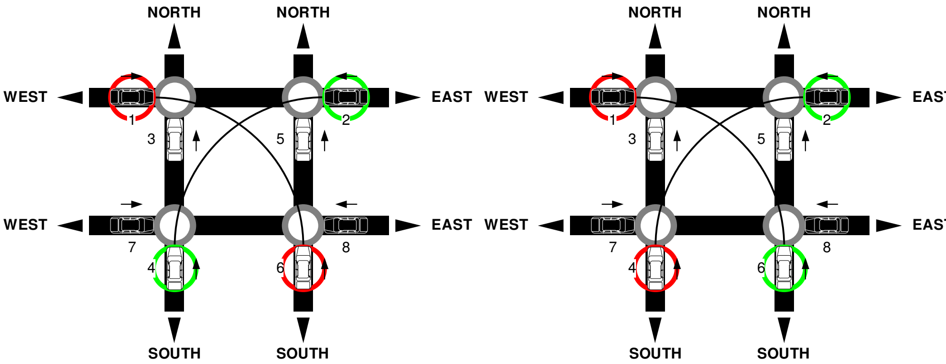


(a) 4 stops (b) 5 stops (c) 6 stops

Figure 5.16: Possible control decisions of the intersections and the number of stops that result

corners of the network alternate the release of vehicles to the downstream intersection to avoid conflicts at that intersection.

As downstream intersections are aware of any expected arrivals from upstream intersections these intersections have an increased cost as a consequence of the simultaneous arrival of the vehicles expected to be released from the upstream intersection. This cost is communicated to the upstream intersections. The upstream intersection incorporates the downstream cost in the next optimization iteration and makes adjustments so that the potential downstream conflict does no longer occur.



(a) Suboptimal (b) Optimal

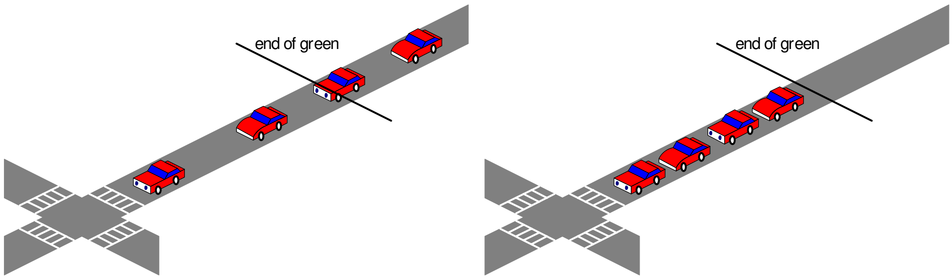
Figure 5.17: Potential downstream conflicts

5.3.4 Analysis and results

Simulations show that as a consequence of the applied iterative coordination procedure so called green waves emerged and dissolved as volumes and platoon ratio changed. As such the developed iterative coordination procedure improves upon traditional signal coordination methods as it is capable to adapt to changes in traffic demand patterns.

One would expect that the time difference between the moment the first vehicle is released from the upstream intersection and the moment it arrives at the downstream intersection is at least the time needed to cover the distance between both intersections in order for the first vehicle of the platoon to arrive at the moment the downstream signal switches to green. Compared to traditional signal coordination platoons are however released earlier. As it turns out, the moment of release of the platoon at the upstream intersection is determined on the basis of the last vehicle of the platoon to arrive at the downstream intersection before the signal at the downstream intersection would switch to red again. The intersection controller is capable to take into account that platoons of vehicles released from an intersection disperse over time and distance. By aiming for the last vehicle in the platoon and stopping the first vehicles of an approaching platoon, the dispersed platoon is compressed so that the number of vehicles that is able to pass the downstream intersection during the green time is maximized (see Figure 5.18).

When total traffic demand reached the capacity level of the network, simulations show that under stable conditions (i.e., little variations in volumes and platoon ratio) the iterative coordination procedure performed worse than traditional signal coordination methods. Careful examination showed the cause of the increased network travel times. Upstream intersections were seen to spare their downstream neighbors. Because the upstream intersections start to buffer traffic as soon as the presumed capacity of the downstream intersection is reached, the downstream intersections are never able to show their true capacity. The true capacity of the network therefore never surfaces, as the intersections hold each other back in showing their true potential. The cause was found in the setting of the saturation flows which were underestimated. The solution can be found in applying different weights to delay encountered upstream and downstream. A higher weight for delay experienced upstream stimulates the intersection controller to clear the own intersection and at the same



(a) Inefficient usage of green (b) Efficient usage of green

Figure 5.18: The efficiency of green usage is determined by the arrival distribution of incoming vehicles

time take the downstream objective into account. The weights are skewed such that the interest of the intersection are put in front, while it still takes the objectives of the downstream intersections into account.

Analysis also showed that when saturation flows were overestimated the last vehicles of the platoon of vehicles released from an upstream intersection were often unable to clear the downstream intersection as saturation flows in practice were lower than in reality. The overestimation of the traffic flows resulted in an overestimation of the number of vehicles able to clear the intersection, leading to the formation of residual queues at the downstream intersection. As the vehicles that are unable to clear the intersection arrive just after the end of green, they had to wait a full signal cycle which significantly impacted the performance of the network. This is why initial saturation flows are initialized conservatively in order to prevent that the performance of the network under stable conditions deteriorates compared to traditional signal coordination.

5.3.5 Discussion

The advantage of the kind of coordination provided by the microscopic coordination procedure over traditional forms of coordination, is that it is adaptive. As traffic volumes and platoon ratios vary over time, there are times that it is not advantageous to provide a fixed progression for vehicles on a corridor. The distributed coordination procedure is able to adapt to different traffic volumes and platoon ratios, and is able to create and to dissolve progression between consecutive intersections on the fly.

Although the application of the microscopic coordination procedure in hypothetical test networks shows very promising results, the question remains to be answered how effective this procedure will be in practice. A necessary requirement to be fulfilled in order for coordination on the microscopic level to be effective is that the arrival times of vehicles for vehicles released from an upstream intersection are predictable. As commitments between intersections are made over longer distances it is more difficult for intersections to keep these commitments as disruptions in the predicted arrival patterns can have a significant impact on the performance of the intersection.

The rules of thumb that are applied in deciding whether a pair of intersections should be coordinated (based on distance and travel times between intersections) are also valid for the coordination layer. If intersections are spaced too far from each other it may be ineffective to coordinate these intersections.

5.4 Macroscopic coordination

In this section the algorithm developed to perform coordination on the macroscopic level is introduced and illustrated by means of part of the network that was introduced in Section 5.2. This section starts out in Section 5.4.1 by motivating the need for coordination on a macroscopic level. Subsequently the algorithm developed to perform coordination on a macroscopic level is introduced in Section 5.4.2. The dynamics and the benefits of the algorithm are illustrated in Section 5.4.3. Section 5.4.4 describes the results from the analysis of the algorithm. The section ends with some discussion in Section 5.4.5.

5.4.1 Motivation

Many traffic-adaptive systems [1, 9, 18, 73] allow the protection of vulnerable areas in the network by holding traffic further upstream. For further information regarding these systems the reader is referred to Chapter 2. This functionality is called "gating". The form of gating applied in the state of the art in urban traffic control (with the exception of TUC, which claims inherent gating behavior) however relies on the operator to determine a set of fixed rules used to restrict the inflow of traffic in a certain area whenever traffic counts in the protected area exceed a certain level. The "gating" applied by the macroscopic coordination procedure defined in this section works dynamically. The agents representing the infrastructure elements of the network continuously monitor whether their part of the network is either under-utilized or over-utilized and try to direct traffic to or steer traffic away from these parts of the network. They do so by coordinating their planned outflow with their downstream neighbors.

5.4.2 Coordination procedure

In order to prevent that traffic breaks down on vulnerable parts of the network the inflow of traffic toward these parts of the network should be constrained. The coordination procedure developed employs two types of constraints: (1) hard constraints enforcing that the volume of traffic entering the vulnerable area does not exceed the volume the infrastructure is able to handle, and (2) soft constraints used to tempt agents further upstream in the network to steer traffic away from the vulnerable area, such as to alleviate the stress put on downstream agents that have already started gating. The following notation is introduced:

I, O	$\equiv$	Set of upstream I and downstream O agents indexed by i and o respectively.
L	$\equiv$	The length of the link expressed in meters.
ρ^{critical}	$\equiv$	Desired density on the link.
N	$\equiv$	Number of vehicles present on the link.
N^{desired}	$\equiv$	Desired number of vehicles on the link.
$q^{\text{out,desired}}$	$\equiv$	Desired total outflow from the link.
$q_o^{\text{out,desired}}$	$\equiv$	Desired total outflow from the link toward the link represented by agent o.
β_o	$\equiv$	The turning rate (the fraction of the total flow that leaves the link via the link represented by agent o).
q^{in}	$\equiv$	Realized inflow to the link.
q^{out}	$\equiv$	Realized outflow from the link.
q_i^{in}	$\equiv$	Realized inflow to the link that originates from the upstream link represented by agent i.
$q_i^{\text{in,max}}$	$\equiv$	Maximum attainable inflow from upstream link represented by agent i.
q_o^{out}	$\equiv$	Realized outflow from the link to the downstream link represented by agent o.
$q_i^{\text{in,desired}}$	$\equiv$	Desired inflow to the link by the upstream link represented by agent i.
$q^{\text{in,desired}}$	$\equiv$	Desired total inflow to the link.
$q_o^{\text{out,granted}}$	$\equiv$	Total outflow granted by downstream agent o.
$q^{\text{out,granted}}$	$\equiv$	Total outflow granted.
$q_i^{\text{in,granted}}$	$\equiv$	Inflow granted to agent i.
$q^{\text{in,granted}}$	$\equiv$	Total inflow granted.
c^{in}	$\equiv$	Cost for vehicles entering the link.
c^{out}	$\equiv$	Cost for vehicles exiting the link.
c_o^{out}	$\equiv$	Cost for vehicles exiting the link to the downstream link represented by agent o.

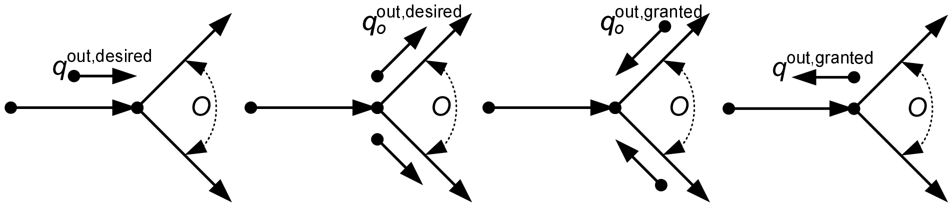
In order to ensure maximal throughput on a link the density of the link should be close to, but should not exceed the critical density. Given the critical density (ρ^{critical}) and the length of the link (L) the maximum number of vehicles that the link wishes to accommodate (N^{desired}) can be determined as follows:

$$N^{\text{desired}} = L\rho^{\text{critical}} \tag{5.1}$$

The number of vehicles that is already present on the link (N) can be approximated as suggested in [123, 124] and is summarized in Appendix B. In order to maintain the desired number of vehicles on the link the agent coordinates its outflow and its inflow with its downstream agents and upstream agents respectively. We will now proceed to describe the procedures followed by an agent to coordinate its outflow and inflow.

Outflow coordination

In order to maintain the desired number of vehicles on the link the agent will first try to increase its outflow to release any excess of vehicles that might arise given the upstream inflow (q^{in}) and the current surplus of vehicles on the link ($\max\left\{0, N - N^{\text{desired}}\right\}$). The total



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4

Figure 5.19: Steps to coordinate the outflow

outflow desired (step 1 of Figure 5.19) then becomes:

$$q^{\text{out,desired}} = q^{\text{in}} + \max\left\{0, N - N^{\text{desired}}\right\} \tag{5.2}$$

The total outflow desired is distributed over the downstream exits for each $o \in O$ as follows:

$$q_o^{\text{out,desired}} = \beta_o q^{\text{out,desired}} \tag{5.3}$$

where β_o is the turning rate (the fraction of the total flow of the link that leaves the link via o).

After the total desired outflow has been determined it is sent to the downstream agents as an inflow request (step 2 of Figure 5.19). As soon as the granted outflow ($q^{\text{out,granted}}$) and the cost of outflow (c_o^{out}) has been received (step 3 of Figure 5.19; see also Section 5.4.2 for how these quantities are determined), the outflow to realize is calculated as follows (step 4 of Figure 5.19):

$$q_o^{\text{out}} = \min\left\{q_o^{\text{out,granted}}, (1 - c_o^{\text{out}}) q_o^{\text{out,desired}}\right\} \tag{5.4}$$

Inflow coordination

As inflow requests are received by the downstream agents (step 1 of Figure 5.20) they are aggregated (step 2 of Figure 5.20) to determine the total inflow requested by the upstream agents:

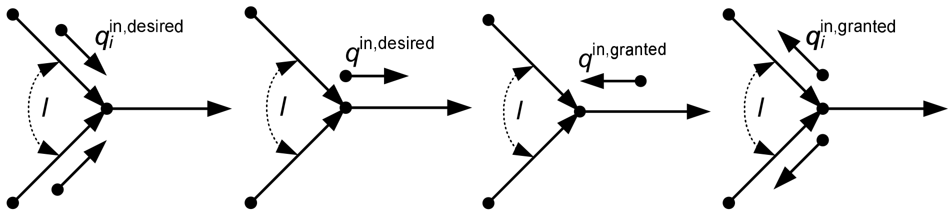
$$q^{\text{in,desired}} = \sum_{i \in I} q_i^{\text{in,desired}} \tag{5.5}$$

The inflow that can be granted depends on the outflow that has been granted to the agent and any excess of demand that is present on the link (step 3 of Figure 5.20):

$$q^{\text{in,granted}} = q^{\text{out,granted}} - \max\left\{0, N - N^{\text{desired}}\right\} \tag{5.6}$$

If the total inflow requested does not exceed the total inflow granted ($q^{\text{in,desired}} \leq q^{\text{in,granted}}$) then the inflow granted is divided over the entries proportionally to the size of the requested inflow as follows (step 4 of Figure 5.20):

$$q_i^{\text{in,granted}} = \min\left\{q_i^{\text{in,max}}, \frac{q_i^{\text{in,desired}}}{q^{\text{in,desired}}} q^{\text{in,granted}}\right\} \tag{5.7}$$



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4

Figure 5.20: Steps to coordinate the inflow

If the total inflow requested does exceed the total inflow granted ($q^{\text{in,desired}} > q^{\text{in,granted}}$) then the total inflow granted is first proportionally distributed over the entries that were given the highest priority. Whereas the total inflow granted is allowed to exceed the total inflow requested in case there is inflow capacity to spare, the inflow granted is maximized to the inflow requested if there is no inflow capacity to spare. The actual inflow granted can be calculated as follows:

$$q_i^{\text{in,granted}} = \min\left\{ q_i^{\text{in,desired}}, q_i^{\text{in,max}}, \frac{q_i^{\text{in,desired}}}{q^{\text{in,desired}}} q^{\text{in,granted}} \right\} \tag{5.8}$$

As soon as the inflow that will be granted to the highest priority entries has been calculated the inflow that remains to be granted (if any) can be distributed over the lower prioritized entries.

In order to have downstream costs trickle upstream in the network the cost is not only determined by the impact the intended outflow has on the local performance of the link (expressed by a local cost) but also by the cost involved for its own intended outflow (expressed by outflow-costs). Compare this to the production of goods, where costs are not only made in the manufacturing of an end product from raw materials but also in the purchase of these raw materials. As the cost of outflow is incorporated in the price that is set for the inflow of traffic, downstream costs are automatically passed on in the network. This allows traffic to choose a cheaper alternative further upstream in the network. If cost would be determined just on the basis of the impact the intended inflow would have on the quality of the link, costs would never trickle upstream in the network.

The inflow cost is determined as follows:

$$c^{\text{in}} = c^{\text{out}} + \max\left\{ 0, \frac{q^{\text{in,desired}} - q^{\text{in,granted}}}{q^{\text{in,granted}}} \right\} \tag{5.9}$$

The macroscopic coordination procedure and its benefits can be illustrated using some simple examples.

5.4.3 Illustrative examples

In this section the macroscopic coordination procedure is illustrated using three simple examples:

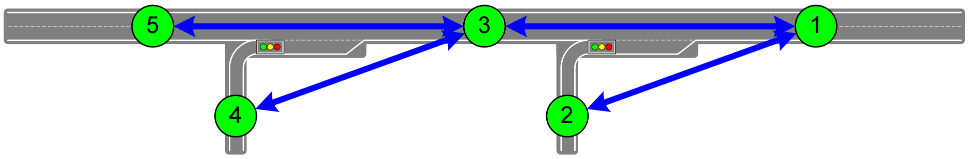


Figure 5.21: Geometry of the freeway corridor

- The first example illustrates the coordination procedure for a freeway network. This example shows how the flow of traffic toward a downstream bottleneck can be regulated by two ramp metering installations which are coordinated by means of the developed coordination procedure.
- The second example illustrates the coordination procedure for an urban corridor controlled by traffic signals. In this example the macroscopic coordination procedure ensures that any excess of traffic is buffered at the entrance of the corridor.
- The third and final example illustrates the coordination procedure for a network, where the macroscopic coordination procedure is used to inform traffic regarding the most cost-effective route alternative.

Illustrative example for a freeway corridor

To alleviate traffic congestion due to recurrent congestion, ramp metering has been applied throughout the world. Ramp metering aims to limit the number of vehicles entering the freeway so that freeway flow in the ramp metering installation's area of influence can be maintained at or exceed the desired quality level. Excess demand is forced to wait at the entrance ramp. The intention of ramp metering is, therefore, to maintain uninterrupted, non-congested flow on the freeway as long as possible by transferring delay from the freeway to the entrance ramp.

In Figure 5.21 the freeway corridor is represented that corresponds to route 1 of the network illustrated in Section 5.2. In this representation traffic travels from the left side to the right side of the picture. Traffic enters the freeway represented in the figure from the freeway upstream (left side of the picture) and the two on-ramps. Each on-ramp is metered by a ramp metering installation. Each link in the network is represented by an agent (numbered 1 to 5). The freeway can be considered a vulnerable area of the infrastructure and as such the on-ramps to the network are locally metered or "gated".

Now assume that the actions of the ramp metering installations are not coordinated and that each individual ramp metering installation is merely concerned with the flow of traffic local to their on-ramp. Each ramp metering installation then just acts upon the information it receives from the measurement loops directly upstream or downstream of the on-ramp (depending on the chosen implementation). Whenever traffic demand on link 1 approaches the capacity of the road, the ramp metering installation at on-ramp 2 will delay traffic on the on-ramp. As traffic demand from the link represented by agent 3 does not dwindle the queue on on-ramp 2 will grow uncontrollably.

Coordinated ramp metering refers to the application of ramp control to a series of ramps where the interdependency of ramp operations is taken into account. The primary objective of integrated ramp control is to prevent or to reduce the occurrence of congestion on a longer

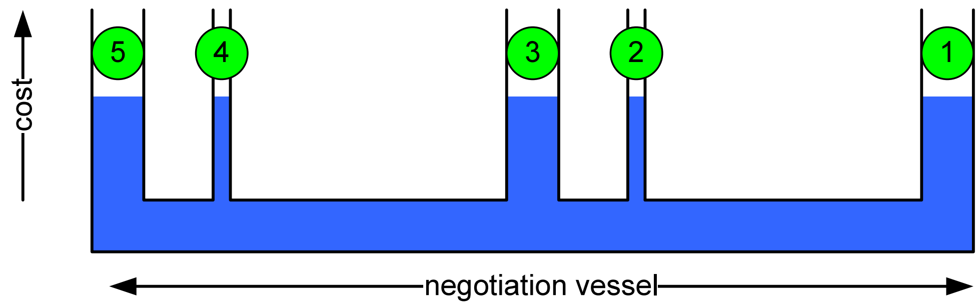


Figure 5.22: Communicating cost vessels

stretch of freeway. Therefore, the control of each ramp is based on the demand-capacity considerations for the whole stretch rather than on the demand-capacity constraint at each individual ramp [59, 62–64, 77, 88]. Instead of an integrated approach the coordination approach of this thesis maintains the autonomy of the individual ramp metering installations and instead coordinates the actions of the individual ramp metering installations using the agents representing each link. The multi-agent approach developed in this chapter uses a distributed approach instead of an integrated approach to coordination.

Now assume that the actions of the ramp metering installations are coordinated using the coordination procedure described in Section 5.4.2. In this case agent 1 receives the inflow requests from the upstream agents 2 and 3. The link represented by agent 1 however is not able to handle this upstream demand. As the freeway is considered more important than the on-ramps the inflow is first granted to agent 3 and subsequently to agent 2. As agent 2 is granted less inflow than requested it will start to buffer traffic on the on-ramp using the ramp metering installation. As the inflow requested by agents 2 and 3 exceeds that what agent 1 could grant, agent 1 associates a cost with the inflow.

This cost is also conveyed to agents 2 and 3. As all inflow requested by agent 3 has been granted agent 3 is also capable to grant the inflow requested by agents 4 and 5. However, in this case, although all the inflow requested by agent 4 and 5 is granted, there is a cost associated with the inflow. Agent 4 therefore decides to buffer traffic on the on-ramp using the ramp metering installation as it is cheaper to buffer traffic on the on-ramp compared to having it enter the freeway.

As agent 4 buffers traffic on the on-ramp the inflow of traffic toward agent 3 is reduced and therefore the outflow to agent 1 requested by agent 3 is also reduced. This allows agent 2 to clear part of the buffered traffic on the on-ramp. As agent 2 is able to clear its on-ramp the outflow requested by agent 2 is reduced, which results in a lower cost for the inflow of traffic toward agent 1 for both agents 2 and 3. This reduced cost is again relayed by agent 3 to agents 4 and 5 and as a result agent 4 is able to release some of the traffic built up on the on-ramp. This process continues until the cost of the traffic that has to be buffered in order to respect the hard constraint put on the inflow by agent 1 is equally shared among agent 2 and agent 4. This example illustrates how the coordination procedure effectively creates a system of communicating vessels (see Figure 5.22 for an illustration) which effectively allows different “gating” agents in the network to share the load of protecting vulnerable areas of the infrastructure.

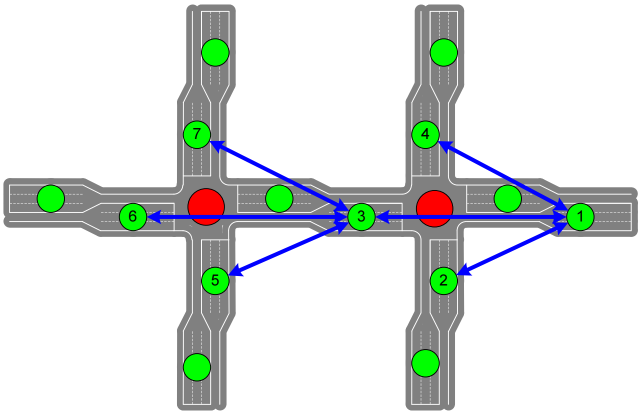


Figure 5.23: Geometry of the urban corridor

Illustrative example for an urban corridor

In Figure 5.23 the urban corridor is represented that corresponds to route 2 of the network illustrated in Section 5.2. This example will focus on traffic that travels from the left side to the right side of the picture. Relevant agents for the example are labeled 1–7.

Figures 5.24–5.27 are used to illustrate the consecutive steps taken by the agents to reduce the volume of the flow originating from the far left of the corridor as a result of a capacity drop on the far right of the corridor. Assume that the state of the network starts out as illustrated in Figure 5.24. In this figure each vertical bar denotes the outflow volume requested by the upstream agents as well as the inflow granted to the upstream agents. Figure 5.24 shows that downstream agents 1 and 3 have just barely been able to fulfill the outflow requested by their upstream agents 2–4 and 5–7 respectively, which indicates that the downstream capacities are fully utilized. Capacity levels are however sufficient to accommodate the demand.

In Figure 5.25 the outflow capacity of agent 1 has been reduced (i.e., as a consequence of the increased inflow of traffic from the minor directions on the downstream junction). Agent 1 is therefore no longer able to fulfill the outflow demand from its upstream agents 2–4.

Consequently agent 3 is also no longer able to accommodate the outflow requested by its upstream agents 5–7. This is illustrated in Figure 5.26. Whereas the downstream capacity for agents 5–7 first sufficed to accommodate the total outflow requested by these agents, agent 3 is now no longer able to process the requested demand.

In Figure 5.27 agent 3 no longer grants the outflow requested by agents 5–7 and as as consequence a queue starts building up at the link represented by agent 6.

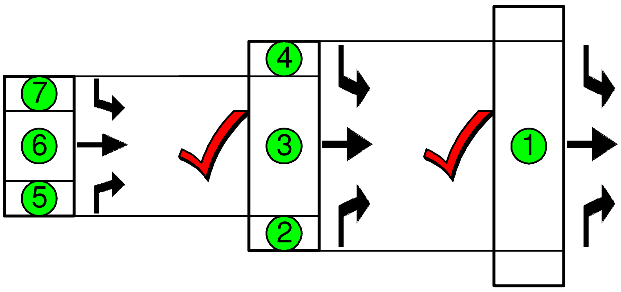


Figure 5.24: Capacity levels are sufficient

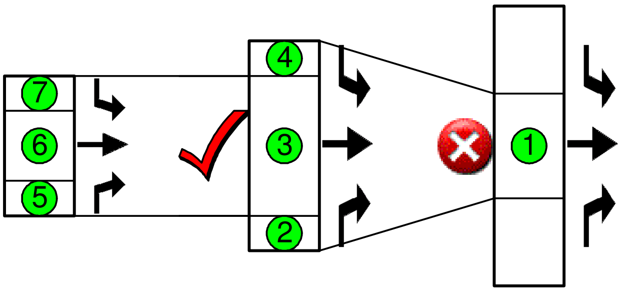


Figure 5.25: Capacity reduction at downstream junction

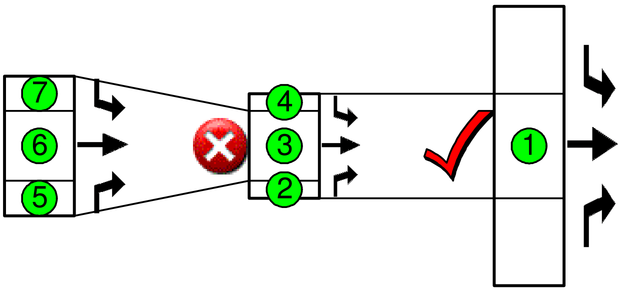


Figure 5.26: Inflow to agent 1 is reduced

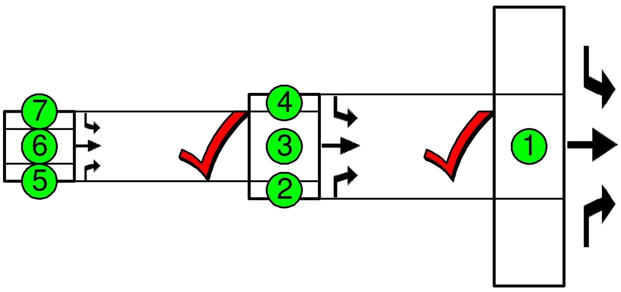


Figure 5.27: Inflow to agent 3 is reduced

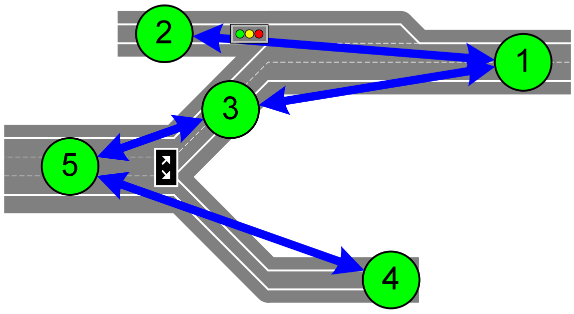


Figure 5.28: Geometry of the network

The macroscopic coordination procedure ensures that the amount of traffic released toward the urban area is buffered outside the network if demand levels exceed the level of demand the network can handle. As traffic is buffered at the gates of the urban area (formed by agent 6) the build-up of queues of idling vehicles in the vulnerable urban area is prevented.

Illustrative example for a network

Variable message signs (VMS) are programmable traffic control devices that display messages composed of letters, symbols or both. They are used to provide information about changing conditions in order to improve operations, to reduce accidents, and to inform travelers. They may advise or urge drivers to change travel speed, to change lanes, to divert to a different route, or simply to be aware of a change in current or future traffic conditions.

A VMS system can directly benefit (a) routing choice, saving vehicle traveled miles and hours, and (b) congestion reduction. The objectives of VMS depend on the application. For this example we will focus on the application of a VMS for route guidance. VMS applied for route guidance are also known as DRIPs.

In Figure 5.28, a small freeway network is represented. The example given below focuses on the traffic that travels from the left side to the right side of the picture. Relevant agents for the example are labeled 1–5.

The link represented by agent 5 branches off into the links represented by agents 3 and 4. Road users are informed just before this branching point about the traffic conditions on both branches by means of a VMS. A ramp metering installation meters the traffic that wants to enter the freeway link represented by agent 2.

In case a queue starts forming caused by e.g., an incident on the link represented by agent 1 a local ramp metering installation will only start to meter when the end of the queue approaches the junction of the links represented by agents 1, 2, and 3. This will somewhat slow down the growth of the queue on link 3. The road user will be informed by the DRIP at the junction of link 3, 4, and 5 about the worsening traffic conditions at link 3. This will slow down the growth of the queue even more. When the end of the queue reaches the junction of links 3, 4, and 5, traffic that never planned to take a route that includes link 3 in the first place will be faced with the consequences of the incident. The queue will then start to grow rapidly since road users wanting to follow link 4 will be blocked passage by road users wanting to follow link 3.

The situation illustrated above could have been further delayed or could have been prevented altogether when the actions of the ramp metering installation and the DRIP would have been coordinated.

As soon as the density of traffic on link 1 near the ramp metering installation approaches a level that entails that traffic entering link 1 from link 2 will have to be significantly delayed, the ramp metering installation could call upon the DRIP to start rerouting traffic via link 4. By combining the efforts of both the ramp metering installation and the DRIP the growth of the queue can be delayed significantly. By delaying the growth of the queue it will take longer for the end of the queue to reach the junction of arteries 3, 4, and 5. The queue might even dissolve before it reaches the junction. This way, road users that want to follow link 4 can do so as long as possible without being blocked passage.

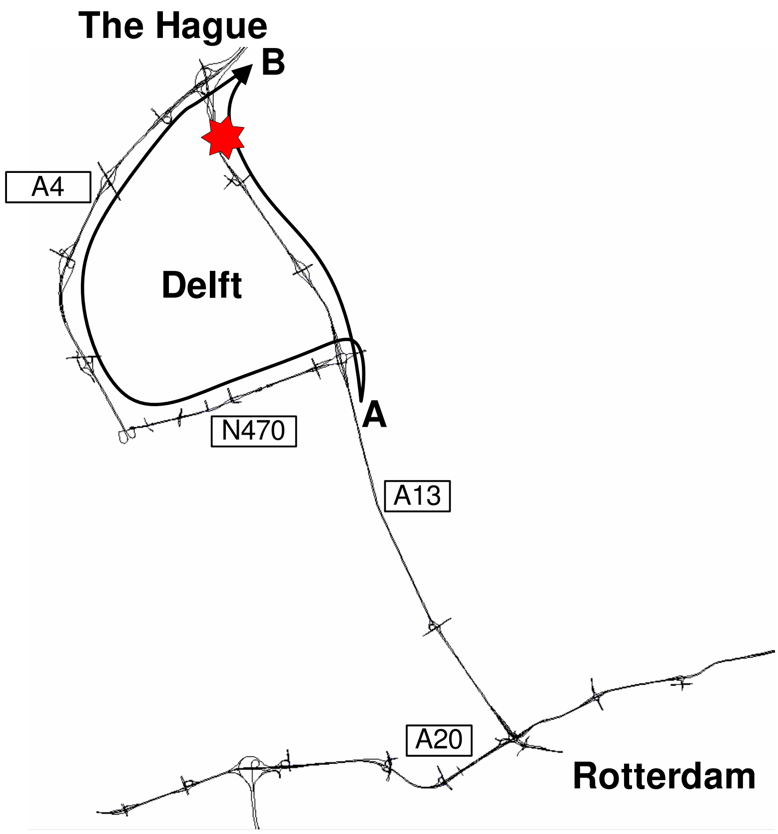
The route guidance installation is made aware of the downstream traffic state by having downstream costs trickle upstream in the network.

Since there is a large time lag between the moment the route guidance installation takes action and the moment the effects of these actions are visible at link 1, the ramp metering can be seen to help out as long as is needed. Traffic on the on-ramp remains buffered as long as the capacity reduction persists. Only when the capacity reduction is cleared the queue on the on-ramp is allowed to dissolve. If cost would be determined just on the basis of the impact the intended inflow would have on the quality of the link, costs would never trickle upstream in the network. If costs would be determined on the basis of the local traffic state alone then in the network of the scenario the route guidance installation would either become aware of the downstream traffic state very late or not at all.

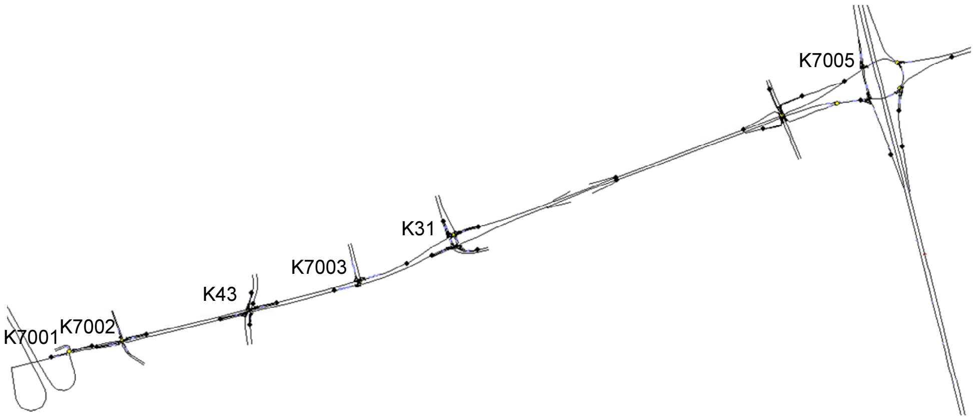
5.4.4 Analysis and results

The macroscopic coordination procedure was tested using a simulation of the Delft area. Figure 5.29 shows the geometry of the studied network. In this network the N470 (Kruithuisweg), a provincial road, forms a connection between the A4 and the A13, which are two highways. In case of an accident on one of the two highways, the other highway can be offered as an alternative by rerouting traffic via the N470. For the simulation it was chosen to create an accident during the morning rush hour. As during rush hour there is already a lot of congestion for traffic heading south on the A13 and for traffic heading east on the N470 it was chosen to locate the incident on the end of the A13 for traffic heading north. As the N470 for traffic heading west is free of congestion, the N470 can be offered as an alternative. As access to and traffic on the N470 is controlled by a number of traffic-actuated intersections this is however not a trivial task. Figure 5.29(b) shows the geometry of the N470. The N470 is controlled by 5 traffic controllers (named K7001, K7002, K42, K7003, K31, and K7005). Four of these traffic controllers control regular intersections and one (K7005) controls a larger interchange.

The network was modeled using the AIMSUN-NG traffic simulation package. Each of the intersections in the network is controlled by the same control program used to control the real intersections. This was made possible by means of a special interface developed by DHV and Path2Mobility. This interface, the CAI-interface, allows us to interface the controllers of most control program suppliers to the AIMSUN-NG model. The model was created by DHV, and calibrated visually, i.e., checks were made whether congestion manifested itself on the same sites and whether it appeared and disappeared at about the same time. The purpose for which the original model was developed did not demand for more de-



(a) Geometry of the entire network



(b) Geometry of the N470

Figure 5.29: Geometry of the studied network

tailed calibration at the screen line level. In this case a more detailed calibration was called for. For this purpose traffic counts were provided by the province of Zuid-Holland for the intersections under their administration and compared with the traffic counts provided by the simulation.

On the basis of the calibrated network, three scenarios were subject to analysis:

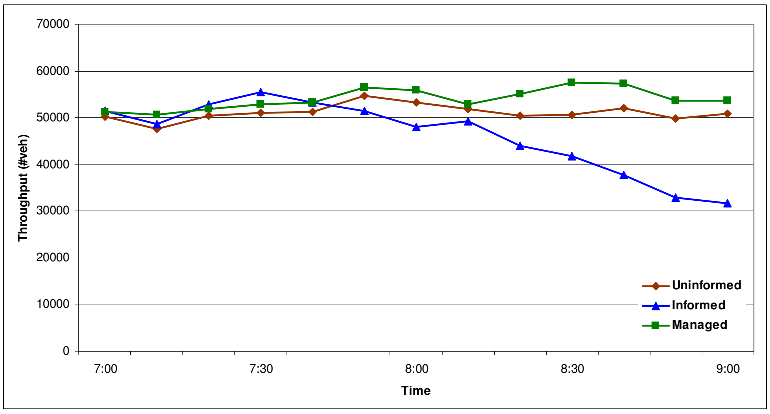
1. *Uninformed* In this scenario road users are unaware of the traffic conditions downstream and do not adapt their route to changing traffic conditions.
2. *Informed* This scenario builds upon the uninformed scenario, but in this case road users are made aware of the traffic conditions in the network and are able to adapt their route while en route.
3. *Managed* This scenario builds upon the informed scenario, but in this case the network is controlled by agents that are capable to adapt the network to the changing traffic conditions. Each link in the network is represented in the traffic management test bed by an agent. Each agent is made responsible for the performance of its link.

In Figure 5.30 the total throughput of the network is shown for the different scenarios. As can be seen in the figure the managed scenario clearly outperforms the other scenarios. At first sight it might be surprising to see that throughput deteriorates in the scenario in which the road users are informed about the current conditions in the network. Information is seen as the enabler on the basis of which traffic can manage itself and a so called user-optimum can be reached in the network. This is true, but only when we can assume perfect information and fully rational behavior. Unfortunately it is not true that additional information automatically leads to a better traffic performance. In this case, total traffic performance is better when the road user is left in the dark regarding traffic conditions elsewhere in the network. This is because the road user is only provided with current and not with future traffic conditions.

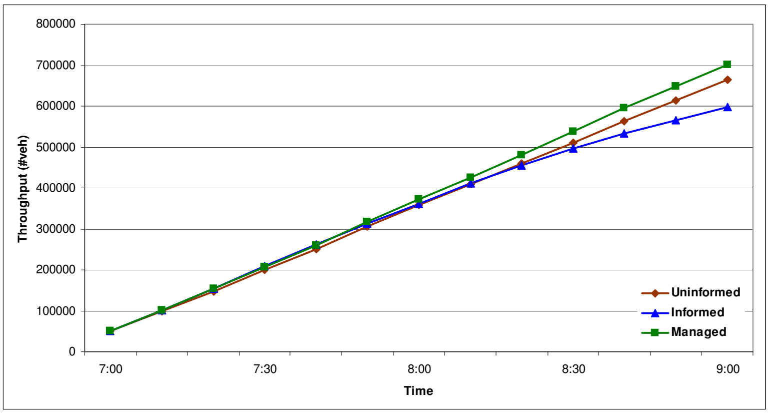
In the simulations traffic heading to the North (in the direction of The Hague) reroutes to the N470 and A4 (the indirect route) as soon as conditions on the A13 (the direct route) are such that the indirect route becomes a viable alternative. The capacity of the various intersection following the first intersection that traffic passes when entering the N470 is such that traffic is able to travel the N470 more or less unhindered. Problems arise when traffic reaches the last intersection on the N470 as this intersection is uncapable to handle the extra traffic. As soon as traffic on the A13 becomes aware of the the delay resulting from the bottleneck at the end of the N470 traffic no longer reroutes over the N470. However, this information becomes available too late for traffic that already entered the N470. The N470 can thus be compared to a funnel. Traffic is lured into the funnel by the promise of a better travel time, but eventually becomes stuck at the end. The delay experienced by traffic in the funnel is such that overall the traffic situation has turned from bad to worse.

In the managed case the traffic controller communicates its spare capacity downstream so that downstream controllers are made aware of the downstream capacity and can reduce the inflow of traffic to prevent downstream problems. Based on this information the intersection controller on the offramp from the A13 to the N470 reduces the traffic inflow on the N470 to a level that can be accommodated by the N470 as a whole, thus preventing future problems.

In Figure 5.31 the travel times are shown that are realized by traffic taking the indirect and the direct route for each scenario. As can be seen in Figure 5.31(a) traffic in the



(a) Throughput



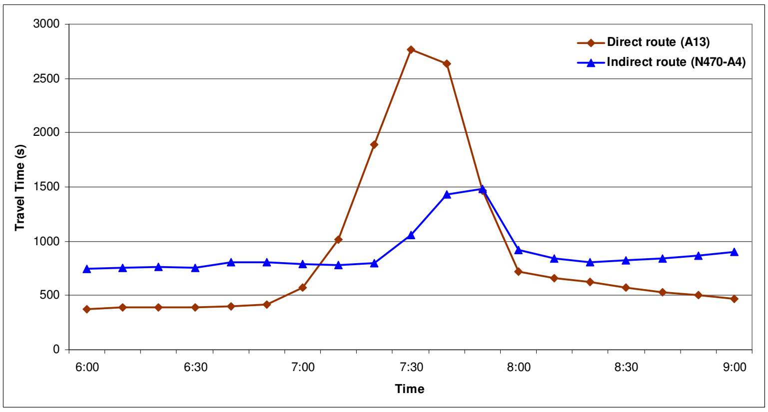
(b) Cumulative throughput

Figure 5.30: Total throughput for the different scenario's

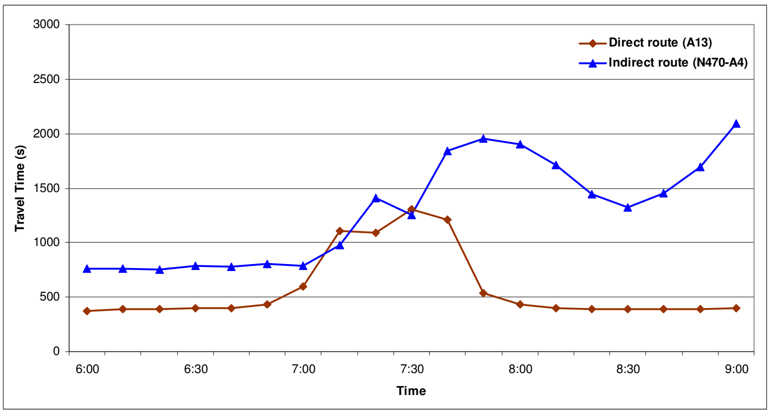
uninformed scenario is unaware of the alternative offered by the indirect route. Although the alternative offers a lower travel time it is not used resulting in an underutilization of the available infrastructure. In Figure 5.31(b) traffic is informed about the alternative. However, after some time the alternative becomes oversaturated because of the downstream bottleneck and travel times explode. The problem in this case goes from bad to worse. In Figure 5.31(c) however the alternative route is put to good use. Only a part of the traffic wishing to make use of the alternative is allowed because traffic is metered at the entrance of the N470. The N470 in this case forms a real alternative to the direct route. The travel times on both routes are comparable and the network is properly utilized.

5.4.5 Discussion

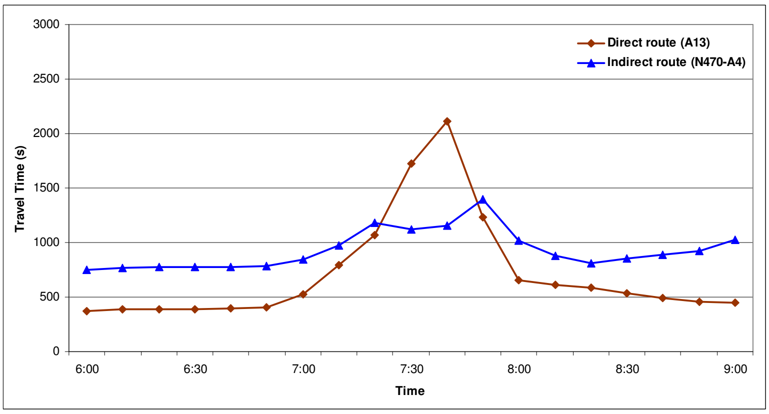
The advantage of the form of gating provided by the macroscopic coordination procedure over traditional forms of gating, is that it makes each link in the network aware of the cost



(a) Uninformed Traffic



(b) Informed Traffic



(c) Managed Traffic

Figure 5.31: Route travel times for the different scenarios

involved for the network of each vehicle released to each of its downstream links. This differs from traditional gating techniques as in these techniques the links are only made aware of the cost involved of releasing a vehicle on a specific set of links. The macroscopic coordination procedure was very effective in isolating the potential bottleneck in the network used in our analysis long before it would have surfaced in the travel times realized over this alternative. The supervision layer allows the traffic operator to explicitly take the spare capacity of the network into account when deciding whether he wants to offer a certain alternative.

The analysis of the supervision layer was done in the Delft region. This region offers only two alternatives. Under regular conditions, with some, but not to severe congestion, the alternative offered in the analysis cannot be considered a valid alternative as it is not attractive enough. A recommendation for further research is to test the macroscopic coordination procedure in a network with multiple, equally attractive alternatives, and see how the coordination procedure behaves in non saturated conditions. Furthermore, it is recommended to further analyze the coordination procedure with different incident locations and different demand patterns (e.g., evening rush hour, the weekend).

5.5 Conclusions

This chapter has described the principles and the dynamics of the coordination procedures developed for coordination on a microscopic and a macroscopic level. The microscopic coordination procedure can be used to coordinate intersections on the level of vehicles and platoons. The advantage of the kind of the coordination provided by the microscopic coordination procedure over traditional forms of coordination, is that it is adaptive. As traffic volumes and platoon ratios vary over time, there are times that it is not advantageous to provide a fixed progression for vehicles on a corridor. The distributed coordination procedure is able to adapt to different traffic volumes and platoon ratios, and to create and to dissolve progression between consecutive intersections on the fly.

The macroscopic coordination procedure can be used to coordinate all kinds of traffic control instruments (not just intersection controllers) on the level of capacities and flows. The advantage of the kind of the coordination provided by the macroscopic coordination procedure over traditional forms of coordination, is that it does not rely on fixed rules used to restrict the inflow of traffic in a certain area whenever traffic counts in the protected area exceed a certain level, but that it is adaptive.

The developed coordination procedures can be of considerable help in coordinating the individual autonomously functioning traffic management measures that are deployed along the roads today. By allowing the individual instruments to coordinate their actions based on the information they receive from sensors and each other, a finer means of controlling traffic on a network can be realized. This way the traffic operator can focus on the slower control loops of the higher layers of the traffic management system.

Chapter 6

Conclusions and recommendations

This chapter concludes the thesis. Section 6.1 draws some general conclusions. The conclusions drawn at the end of each chapter are summarized in Section 6.2. In Section 6.3 the main academic and practical contributions of this thesis are considered. Finally, in Section 6.4, suggestions are made for future research.

6.1 General conclusions

The objective of this thesis is to develop a distributed, multi-agent, approach to traffic control. The choice for a distributed approach was motivated by the fact that a centralized control approach is often not feasible due to computational complexity, communication overhead, and lack of scalability.

The creation of a distributed, multi-agent approach requires the subdivision of the control problem into several loosely coupled subproblems, such that the combination of all the solutions of the subproblems together approximate the solution of the original control problem. In the framework proposed by this thesis each piece of infrastructure is represented by an agent that tries to attain its local objective in close cooperation with the other agents.

The procedure developed in this thesis to coordinate the agents' efforts results in a process in which agents iteratively determine their optimal control setting in response to the information received from neighboring agents. The optimal control settings are based both on local information and on information gained from neighboring agents during the process. The impact of the local control setting on the performance of neighboring agents is incorporated in the optimization. The coordination procedures have been designed to allow coordination on both a macroscopic and a microscopic level. On a macroscopic level the coordination procedure is used to reach agreement on the level of inflow and outflow volumes, whereas on the microscopic level the coordination procedure is used to reach agreement on the timing of specific vehicle arrivals.

Negotiation at the microscopic level is primarily of importance for those parts of the network that are controlled by traffic signals. In order to perform coordination on the microscopic level a new look-ahead adaptive control algorithm was developed that is capable to incorporate the vehicles expected to be released by upstream agents as well as the signal plans from downstream agents in its decision making process.

6.2 Conclusions per chapter

This section summarizes the most important conclusions drawn at the end of each relevant chapter. In Chapter 2 the existing literature was reviewed on traffic control. This chapter showed how traffic control evolved from fixed-time traffic controllers (tuned off-line on the basis of historical traffic demand patterns) to fully adaptive controllers, that are capable to adapt on-line to changing traffic demands. Traffic-adaptive systems have been around since the mid 1970's, and offer considerable advantages to traffic-actuated control. However, despite these advantages, traffic-actuated systems are still the dominant type of control. This leads us to question why adaptive systems are so slowly adopted. It is the author's impression that the following factors actively contribute to the slow adoption of traffic-adaptive control: (a) the complexity of these systems, (b) the additional detector requirements, (c) uncertainty about the additional performance of these systems, and (d) different control philosophies between countries.

Numerous case studies have been undertaken to ascertain the benefits of adaptive signal control as opposed to more traditional types of control. However, in many of these "before and after" studies the before situation more often than not does not reflect the best that traditional control has to offer. The performance of the adaptively controlled network is for instance compared to the performance of a network of fixed-time controllers, that have not been updated for some time. This is explicable, because it is economically sound to first replace systems that have been written off. Furthermore, it is difficult to ascertain whether the results of the study can be be easily transfered to one's own local situation. Traffic control practices in the United States are for instance different from those in Europe. As comparison studies are costly, it is the impression of the author that there is a task for central governments to ascertain whether the different commercially available traffic-adaptive control systems are applicable and whether they show benefit for the local, national, situation. This way the market can be opened for such systems.

In Chapter 3 a more detailed review was done of existing traffic-adaptive systems to ascertain their similarities and their differences. This chapter makes clear that the differences between these systems are largely brought forth by the fact that it is computationally intractable to attain optimal controller settings in real-time. The goal of each adaptive system is thus to approach the optimum as close as possible give the time available. In order to accomplish this each adaptive system has to make compromises. The taxonomy classifies each traffic-adaptive control system based on the compromises made on the most important features common to traffic-adaptive systems. Of these features, the optimization method, the length and resolution of the planning horizon, the update frequency, the prediction model, and the architecture chosen, are the features on which traffic-adaptive systems differ most. Although the literature is not clear about the motivation for some of the compromises made by each adaptive system, the author believes that skill has been just as important a factor as science in the realization of these compromises. As skill is primarily brought forth with experience the author believes that these compromises would turn out differently when re-evaluated for different application areas.

Inspired by the analysis of the different approaches to traffic-adaptive control a new hybrid algorithm was developed. The developed algorithm, which is described in Chapter 4, is highly configurable and uses the more flexible movement-based approach instead of a stage-based approach. This algorithm significantly outperforms existing approaches to traffic-adaptive control in terms of speed. Typically, because of the complexity of these algorithms, the length of the horizon over which the optimization takes place is significantly shorter than the length of the horizon over which arrivals are predicted. The speed of the developed algorithm allows us to typically choose longer optimization horizons, which allows us to not only take imminent arrivals into account but also the arrivals expected of upstream intersections. The possibility to incorporate the expected arrivals from upstream intersections in the optimization of the intersection is a requirement in order to be able to adaptively coordinate the actions of nearby intersections.

In Chapter 5 a coordination mechanism is presented that is capable of dynamically coordinating nearby intersections making use of the algorithm presented in Chapter 4. This coordination mechanism is capable of creating and dissolving so called green waves on the vehicle level if it is beneficial to do so. Also in this chapter another coordination mechanism is presented that can be used to coordinate traffic control measures (not just traffic signals) on the level of flows. Using this coordination mechanism, upstream agents are informed whether the downstream network is either under-utilized or over-utilized so that agents can try to direct traffic to that part of the network or to steer it away.

6.3 Main contributions

The main contributions of this thesis are:

- the coordination mechanisms that were developed on the basis of which of control decisions can be adaptively coordinated on both a macroscopic and microscopic level.

- the intersection control algorithm developed, which incorporates a movement-based as opposed to a stage-based approach to traffic control and which outperforms current approaches to look-ahead traffic-adaptive control in terms of speed.

- the taxonomy that was created of existing traffic-adaptive systems on the basis of which the fundamentals of the different systems can be easily compared.

- the testbed that was created in order to properly analyze and to test the multi-agent framework and which allows to further improve upon the results of this thesis and also to experiment with different traffic control concepts.

6.4 Recommendations for future research

This section summarizes the most important recommendations for future research and also introduces some new directions of research. The recommendations for future research can broadly be categorized in recommendations to extend the models used in the optimization process, to extend the framework for other traffic control instruments, to research alternative coordination mechanisms, and to perform further experiments.

6.4.1 Model extensions

The models used to predict traffic flow on the approaches to the intersection are by no means perfect and the uncertainty in the predictions is considerable. A way to further improve upon the quality of the signal plan would be either to use more accurate prediction models or to explicitly incorporate uncertainty in the decision making process, maximizing expected performance. Uncertainty can be explicitly incorporated if a probabilistic model is employed in the optimization process. All of the real-time signal control algorithms surveyed employ a deterministic model instead of a probabilistic model. The look-ahead adaptive control algorithm developed in Chapter 4 is no exception. A probabilistic model does not assume that the next state is completely determined by the current state and the chosen control decision. Rather, there is a probability distribution for what the next state will be. However, this probability distribution still is completely determined by the current state and chosen control decision. A recommendation for future research is to ascertain the performance benefits of an algorithm that is able to explicitly incorporate uncertainty in its decision making process and to determine whether these performance benefits can be attained at an acceptable computational cost.

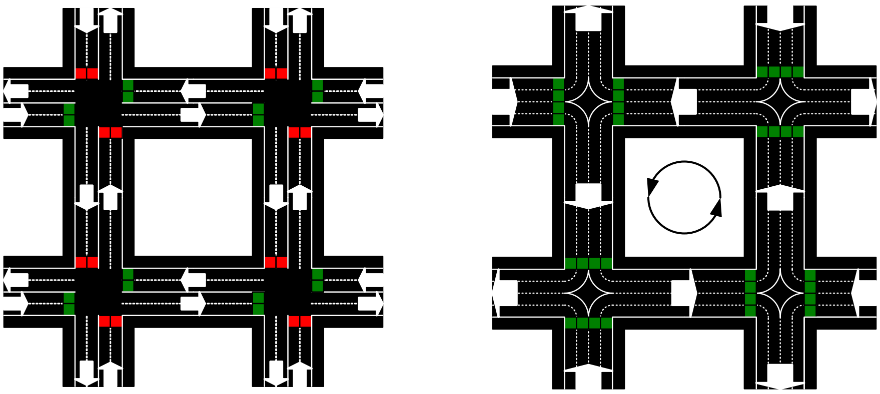
An assumption for all traffic-adaptive control systems is that some estimates for the turning rates at each node are given. The turning rates used in our simulations were therefore continuously updated to reflect current conditions. However, if circumstances are such that turning rates fluctuate rapidly (i.e. caused by the presence of route choice alternatives), current conditions do not represent future conditions well. In these cases the turning rates, like the arrival profiles, should be predicted. In order to accurately predict route choice the route choice model should be able to overview a larger part of the network and will thus be able to update the turning rates for a larger number of agents. As this model does not have to optimize and to coordinate the various control instruments, it can be relatively simple compared to the network models used by traffic-adaptive systems that employ a centralized architecture.

The multi-agent control approach developed in this thesis is capable of coordinating the decisions of nearby traffic control instruments and prevents that certain parts of the network become oversaturated. It is not able to optimize the distribution of flows in the network; they are considered given. In order to optimize the flows in the network the desired distribution of flows should be determined at the network level. The road-user can subsequently be tempted or discouraged to choose for certain routes by appropriately constraining or prioritizing the flows in the network. These constraints and priorities can subsequently be looked after by the multi-agent traffic-control system. These constraints and priorities are currently determined by the road operator. A recommendation for future research is to develop a model that is able to support the traffic operator in the determination and application of these priorities and constraints.

6.4.2 Additional traffic control measures

In this thesis it was assumed that the utilization of the road infrastructure could only be influenced by means of traditional control instruments, such as traffic signals, variable message signs, and ramp metering installations. However, in the foreseeable future, new ways to influence traffic will become available. Roadside systems will for instance be able to communicate directly with vehicles. In that case an intersection controller could inform approaching vehicles regarding the slots of green that are available to them, so that vehicles can aim directly for those slots. In oversaturated conditions platoon formation can thus be stimulated in order to efficiently make use of the available green time. Additionally, the communication between vehicles and infrastructure will make it possible for vehicles to make their itinerary known in advance, so that a more accurate prediction can be made of future traffic demand than can be done based on historical split ratios and volume information alone.

Instruments that allow to dynamically configure the infrastructure itself (e.g., through dynamic road marking and variable message signs) will also become more commonplace in the near future. An example of what could be accomplished by dynamically configuring infrastructure is shown in Figure 6.1. In Figure 6.1(a) a network is depicted that is controlled by traffic signals. In undersaturated conditions, traffic is allowed to flow through the network using the shortest route as the time lost waiting at an intersection for conflicting directions to clear is limited. In oversaturated conditions however the time lost at an intersection can increase significantly. As the performance of each intersection deteriorates the alternative network configuration of Figure 6.1(b) becomes increasingly attractive. Despite the fact that traffic has to travel longer distances in this network, this may be counteracted by the fact that the network itself is without conflicts and there are therefore no intersections to contribute to delay. A recommendation for future research is to incorporate these new control instruments in the multi-agent framework.



(a) Conflict-rich network (b) Conflict-free network

Figure 6.1: Dynamic configuration of infrastructure to reduce the number of conflicts in the network

6.4.3 Alternative coordination mechanisms

The coordination mechanism and the algorithm used for the local optimization of an intersection are modeled as separate processes within the multi-agent framework developed in this thesis. Although this limits the amount of communication between the agents, it also requires that iterations are needed to come up with an area-optimal signal plan. Another option would be to make the coordination algorithm an integral part of the optimization process. This would eliminate the need for iterations. Algorithms used in distributed constraint satisfaction and optimization [127] such as ADOPT [80] show much promise, but this field of research unfortunately currently proves to be too immature to consider it for a practical application such as the real-time coordination of traffic signals. A recommendation for further research is to further extend the applicability of these algorithms so that they could eventually be used for the distributed coordination of traffic signals.

6.4.4 Additional experiments

Although the algorithm developed in Chapter 4 has come forth from an analysis of the strengths and weaknesses of the existing traffic-adaptive systems, it has yet to be determined what the extent of the computational improvements is in practice. A recommendation for further research is therefore to determine the extent of the computational improvements for different intersection configurations and demand levels through simulations in which the developed algorithm is benchmarked against the current state of the art.

The computational improvements of the algorithm developed in Chapter 4 allow more time to be spent on other components of the traffic adaptive system. A question that remains is where this time can best be spent in order to further improve upon the traffic performance. The time gained can, e.g, be employed to further increase the update frequency of the optimization, to incorporate better (i.e. computationally more expensive) models in the optimization process, to increase the length of the planning horizon, to increase the number of blocks etc. A recommendation for further research is therefore to ascertain where this additional time can be spent best for different intersection configurations and demand levels.

The simulations done to test the algorithm developed in Chapter 4 were done with a simulation model that does not actively support modalities other than motorized traffic. In order to accurately assess the performance of the algorithm and traffic-adaptive control in case of an intersection where bicycles and pedestrians are controlled separately, additional simulations should be performed with a simulation model that does support the modeling of these modalities.

Additional simulation experiments are furthermore needed to ascertain the effectiveness of the coordination procedures of Chapter 5 for different network configurations. The added value of coordination, e.g., diminishes as a result of platoon dispersion when the intersections to be coordinated are located further apart. Furthermore, the accuracy of the predicted arrival times diminishes over distance. A recommendation for further research is therefore to determine the maximum distance for which coordination is still useful.

Bibliography

[1] A. Al-Mudhaffar. *Impacts of Traffic Signal Control Strategies*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2006.

[2] A.Wilson. *Handboek Verkeerslichtenregelingen*. C.R.O.W., 2006.

[3] J.F. Barriere, J.L. Farges, and J.J. Henry. Decentralization vs. hierarchy in optimal traffic control. In *Proc. 5th IFAC/IFIP/IFORS Symposium on Control in Transportation Systems*, Vienna, Austria, July 1986.

[4] M.C. Bell and R. D. Bretherton. Ageing of fixed-time traffic signal plans. In *Proc. IEE 2nd International Conference on Road Traffic Control, London*, 1986.

[5] M.G.H. Bell. The estimation of junction turning volumes from traffic counts: the role of prior information. *Traffic Engineering and Control*, 25(5):279–283, 1984.

[6] F. Bellifemine, G. Caire, T. Trucco, and G. Rimassa. *JADE Programmer's Guide*. TILab S.p.A., July 2002. URL: http://sharon.cselt.it/projects/jade/.

[7] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[8] C. Bielefeldt and F. Busch. MOTION - a new on-line traffic signal network control system. In *Proc. 7th International Conference on Road Traffic Monitoring and Control*, pages 55–59, April 1994.

[9] C. Bielefeldt, H. Condie, E. Kosmatopoulos, A. Richards, M. McDonald, J. Mück, and A. Hanitzsch. SMART NETS final report. Napier University, 2004.

[10] J. Boeve, G.R. de Goede, A. Tchernoff, J.I. Hennekeij, and H. Prins. *Handbook, Sustainable Traffic Management, A Guide for Users.* Rijkswaterstaat, November 2003. In Dutch.

[11] F. Boillot. Group-based safety constraint description of an intersection. In *Proc. 9th Mini-EURO Conference on Handling Uncertainty in the Analysis of Traffic and Transportation Systems*, Budva, Yugoslavia, June 2002.

[12] F. Boillot, J.M. Blosseville, J.B. Lesort, V. Motyka, M. Papageorgiou, and S. Sellam. Optimal signal control of urban traffic networks. In *Proc. 6th International Conference on Road Traffic Monitoring and Control*, London, April 1992.

[13] F. Boillot, S. Midenet, and J.C. Pierrelée. Real-life CRONOS evaluation. In *Proc. 10th International Conference on Road Traffic Information and Control*, pages 182–186, London, April 2000.

[14] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.

[15] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, 1999.

[16] L. Botelho. A control structure for agent interaction. In *Proc. IEEE Intelligent Vehicles Symposium (IVHS'00)*, pages 398–403, October 2000.

[17] R.D. Bretherton. SCOOT urban traffic control system: Philosophy and evaluation. In *Proc. 6th IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation*, pages 237–239, September, 1989.

[18] F. Busch and G. Kruse. MOTION for SITRAFFIC - a modern approach to urban traffic control. In *Proc. Intelligent Transportation Systems*, pages 61–64, Oakland, CA, USA, August 2001.

[19] G.E. Cantarella and G. Improta. Capacity factor or cycle time optimization for signalized junctions: a graph theory approach. *Transportation Research Part B*, 22(1):1–23, 1988.

[20] E.C.P. Chang. Improving traffic estimation and prediction through dynamic traffic assignment development. In *Proc. IEEE Networking, Sensing & Control*, 2004.

[21] E.C.P. Chang, S.L. Cohen, C. Liu, N.A. Chaudbury, and C. Messer. MAXBAND-86: Program for optimizing left-turn phase sequence in multiarterial closed networks. *Transportation Research Record*, 1181:61–67, 1988.

[22] A.M. Cheadle, W. Harvey A.J. Sadler, J. Schimpf, K. Shen, and M.G. Wallace. ECLIPSE: A tutorial introduction. Cisco Systems, November 2006.

[23] M.C. Choy, R.L. Cheu, D. Srinivasan, and F. Logi. Real-time coordinated signal control using agents with online reinforcement learning. In *Proc. 83rd Annual Meeting of the Transportation Research Board (TRB'03)*, November 2003.

[24] M. Conrad, F. Dion, and S. Yagar. Real-time traffic signal optimization with transit priority: Recent advances in the SPPORT model. In *Proc. 77th Annual Meeting of the Transportation Research Board*, 1998.

[25] M.R. Crabtree and I.R. Henderson. *MOVA Traffic Control Manual*, B edition, May 2006.

[26] J. Cuena, J. Hernández, and M. Molina. Knowledge-based models for adaptive traffic management systems. *Transportation Research Part C*, 3(5):311–337, October 1995.

[27] P. de Jong. Toelichting op de basisspecificatie. Rijkswaterstaat Adviesdienst Verkeer en Vervoer, 2002.

[28] P. de Jong. Basisspecificatie inclusief basisspecificatie speciaal: verkeerstechnische specificatie van de basisregels van regelingen. Rijkswaterstaat Adviesdienst Verkeer en Vervoer, 2005.

[29] P. B. Dell'Olmo and P. Mirchandani. REALBAND: An approach for real-time coordination of traffic flows on a network. *Transportation Research Record*, 1494:106–116, 1995.

[30] P. B. Dell'Olmo and P. Mirchandani. *Advanced Methods in Transportation Analysis*, chapter A Model for Real-Time Traffic Coordination Using Simulation Based Optimization, pages 525–546. Springer-Verlag, 1996.

[31] C. Diakaki. *Integrated Control of Traffic Flow in Corridor Networks*. PhD thesis, Department of Production Engineering an Management, Technical University of Crete, Chania, Greece, 1999.

[32] V. Dinopoulou, C. Diakaki, and M. Papageorgiou. Applications of the urban traffic control strategy TUC. *European Journal of Operational Research*, 175:1652–1665, 2006.

[33] F. Dion and B. Hellinga. A methodology for obtaining signal coordination within a distributed real- time network signal control system with transit priority. In *Proc. 80th Annual Meeting of the Tranportation Research Board (TRB'01)*, May 2001.

[34] F. Dion and B. Hellinga. A rule-based real-time traffic responsive signal control system with transit priority: Application to an isolated intersection. *Transportation Research Part B*, 36:325–343, May 2002.

[35] S.E. Dreyfus and A.M. Law. *Art and Theory of Dynamic Programming*. Academic Press, Orlando, FL, USA, 1977.

[36] J.L. Farges, I. Khoudour, and J.B. Lesort. PRODYN: On site evaluation. In *Proc. 3rd Int. Conference on Road Traffic Control*, London, England, 1990.

[37] K. Fehon. Adaptive traffic signals are we missing the boat. ITE District 6 Annual Meeting, 2004.

[38] E.D. Ferreira, E. Subrahmanian, and D. Manstetten. Intelligent agents in decentralized traffic control. In *Proc. IEEE Intelligent Transportation Systems (ITSC'01)*, pages 707–711, Oakland, CA, USA, August 2001. IEEE.

[39] N.V. Findler, S. Surender, and Ş. Catavra. A semi-autonomous decentralized system for controlling street traffic signals. In *Proc. IEEE Autonomous Decentralized Systems*, pages 377–383, April 1995.

[40] FIPA. The foundation for intelligent physical agents. URL: http://www.fipa.org, 2003.

[41] E. Friedman-Hill. Jess, the expert system shell for the java platform. URL: http://herzberg.ca.sandia.gov/jess/, May 2003.

[42] S. Gallivan and B. Heydecker. Optimising the control performance of traffic signals at a single junction. *Transportation Research Part B*, 22(5):357–370, 1988.

[43] N.H. Gartner. OPAC: A demand-responsive strategy for traffic signal control. *Transportation Research Record*, 906:75–81, 1983.

[44] N.H. Gartner, S.F. Assmann, F. Lasaga, and D.L. Hout. MULTIBAND — a variable-bandwidth arterial progression scheme. *Transportation Research Record*, 1287:212–222, 1990.

[45] N.H. Gartner, S.F. Assmann, F. Lasaga, and D.L. Hout. A multi-band approach to arterial traffic signal optimization. *Transportation Research Part B*, 25:55–74, 1991.

[46] N.H. Gartner, J.D.C. Little, and H. Gabbay. Simultaneous optimization of offsets, splits, and cycle time. *Transportation Research Record*, 596:6–15, 1976.

[47] N.H. Gartner, F.J. Pooran, and C.M. Andrews. Implementation of the opac adaptive control strategy in a traffic signal network. In *Proc. IEEE Intelligent Transportation Systems Conference*, August 2001.

[48] N.H. Gartner, C. Stamatiadis, and P.J. Tarnoff. Development of advanced traffic signal control strategies for intelligent transportation systems: Multilevel design. *Transportation Research Record*, 1494:98–105, 1995.

[49] N.H. Gartner, P.J. Tarnoff, and C.M. Andrews. Evaluation of the optimized policies for adaptive signal control strategy. *Transportation Research Record*, 1683:105–114, 1999.

[50] D.K. Hale. *Traffic Network Study Tool, TRANSYT-7F, United States Version*. McTrans Center, University of Florida, Gamesville, Florida 32611-6585, January 2006.

[51] E. Hauer, E. Pagitsas, and B.T. Shin. Estimation of turning flows from automatic counts. *Transportation Research Record*, 795:1 – 7, 1981.

[52] K.L. Head, P.B. Mirchandani, and D. Sheppard. Hierarchical framework for real-time traffic control. *Transportation Research Record*, 1360:82–88, 1992.

[53] J.J. Henry and J.L. Farges. PRODYN. In *Proc. 6th IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation*, Paris, France, September 1989.

[54] J.J. Henry, J.L. Farges, and J. Tuffal. The PRODYN real time traffic algorithm. In *Proc. 4th IFAC/IFIP/IFORS Symposium on Control in Transportation Systems*, pages 307–312, Baden-Baden, Germany, April 1983.

[55] P.B. Hunt, D.I. Robertson, R.D. Bretherton, and R.I. Winton. SCOOT - a traffic responsive method of coordinating signals. TRRL Laboratory Report 1014, TRL, 1981.

[56] G. Improta and G.E. Cantarella. Control system design for an individual signalized junction. *Transportation Research Part B*, 18:147–167, 1984.

[57] S. Irani and V. Leung. Scheduling with conflicts, and applications to traffic signal control. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 85–94, January 1996.

[58] J. Jenezon, J. Klijnhout, and H. Langelaar. Motorway control and signalling. *Traffic Engineering and Control*, 28(6), June 1987.

[59] S. Kang and D. Gillen. Assessing the benefits and costs of intelligent transportation systems: ramp meters. Research Report UCB-ITS-PRR-99-19, California PATH, Berkeley, 1999.

[60] J. Klijnhout. Motorway control and signalling: The test of time. *Traffic Engineering and Control*, 25(4), April 1984.

[61] I. Kosonen. Multi-agent fuzzy signal control based on real-time simulation. *Transportation Research Part C*, 11:389–403, 2003.

[62] A. Kotsialos and M. Papageorgiou. Efficiency and equity properties of freeway network-wide ramp metering with amoc. *Transportation Research Part C*, 12:401–420, 2004.

[63] A. Kotsialos and M. Papageorgiou. Motorway network traffic control systems. *European Journal of Operational Research*, 152:321–333, 2004.

[64] A. Kotsialos, M. Papageorgiou, M. Mangeas, and F. Haj-Salem. Coordinated and integrated control of motorway networks via non-linear optimal control. *Transportation Research Part C*, 10:65–84, 2002.

[65] B. Krause and C. von Altrock. A complete fuzzy logic control approach for existing traffic control systems. In *Mobility for Everyone, Proceedings of the 4th World Congress on Intelligent Transportation Systems*, Berlin, Germany, October 1997. Paper no. 2045.

[66] T.W. Lawson, D.J. Lovell, and C.F. Daganzo. Using inputoutput diagram to determine spatial and temporal extents of a queue upstream of a bottleneck. *Transportation Research Record*, 1572:140–147, 1997.

[67] J. Lei and Ü. Özgümer. Decentralized hybrid intersection control. In *Proc. IEEE Decentralized Hybrid Intersection Control*, volume 2, pages 1237–1242, December 2001.

[68] M.J. Lighthil and G.B. Whitham. On kinematic waves, II. a theory of traffic flow on long crowded roads. In *Proc. Royal Society (London), Series A 229*, pages 317–345, 1955.

[69] J.D.C. Little, M.D. Kelson, and N.H. Gartner. MAXBAND: A program for setting signals on arteries and triangular networks. *Transportation Research Record*, 796:40–46, 1981.

[70] X.M. Liu and X. Gong. Study of intersection traffic flow control on the basis of agents. In *Proc. IEEE Intelligent Transportation Systems (ITSC'03)*, volume 2, pages 1755–1758, October 2003.

[71] P.R. Lowrie. The Sidney Co-ordinated Adaptive Traffic System: Principles, methodology, and algorithms. In *Proc. IEE Conference on Road Traffic Signaling*, 207, pages 67–70, 1982.

[72] M.J. Maher. Inferences on trip matrices from observations on link volumes: a bayesian statistical approach. *Transportation Research Part B*, 17(6):435–447, 1983.

[73] P.T. Martin, J. Perrin, B.R. Chilukuri, C. Jhaveri, and Y. Feng. Adaptive signal control ii. UT-03.28, University of Utah, 2003.

[74] V. Mauro and C. Di Taranto. UTOPIA. In *Proc. 2nd IFAC-IFIP-IFORS Symposium on Traffic Control and Transportation Systems*, pages 575–597, 1989.

[75] V. Mauro and C. Di Taranto. UTOPIA. In *Proc. 6th IFAC/IFIP/IFORS Symposium on Control, Computers, and Communications in Transportation*, September 1989.

[76] A. Mekky. On estimating turning flows at road junctions. *Traffic Engineering and Control*, 20(10):486–487, 1979.

[77] A. Messmer and M. Papageorgiou. Motorway network control via nonlinear optimization. *International Transactions on Operations Research*, 2:187–203, 1995.

[78] A.J. Miller. A computer control system for traffic network. In *Proc. 2nd International Symposium on Theory of Road Traffic Flow*, pages 201–220, 1963.

[79] P. Mirchandani and L. Head. A real-time traffic signal control system: architecture, algorithms and analysis. *Transporation Research Part C*, 9:415–432, 2001.

[80] P.J. Modi, W. Shen, M.Tambe, and M. Yokoo. Adopt:asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180, 2005.

[81] M. Molina, J. Hernández, and J. Cuena. A structure of problem-solving methods for real-time decision support in traffic control. *International Journal of Human-Computer Studies*, 49(4):577–600, October 1998.

[82] E.A. Mueller. Aspects of the history of traffic signals. *IEEE Transactions on Vehicular Technology*, 19(1):6 –17, February 1970.

[83] T.H.J. Muller and M. de Leeuw. A new method to design traffic control programs. In *TRB 2006 Annual Meeting CD-ROM*, 2006.

[84] NEMA. NEMA standards publication TS 2- v02.06 traffic controller assemblies with NTCIP requirements. National Electrical Manufacturers Association, 2003.

[85] G.F. Newell. The rolling horizon scheme oftraffic control. *Transportation Research Part A*, 32:39–44, 1998.

[86] N.L. Nihan and G.A. Davis. Application of prediction-error minimization and maximum likelihood to estimate intersection O-D matrices from traffic counts. *Transportation Science*, 23:77–90, 1989.

[87] J.J. Odell, H. Van Dyke Parunak, and B. Bauer. *Agent-Oriented Software Engineering*, chapter Representing Agent Interaction Protocols in UML, pages 121–140. Springer-Verlag, 2001.

[88] M. Papageorgiou. An integrated control approach for traffic corridors. *Transportation Research Part C*, 3(1):19–30, 1995.

[89] M. Papageorgiou. An introduction to signal traffic control strategy TUC. In *Proc. SMART NETS Final Conference – Deliverable 25*, 2003.

[90] M. Papageorgiou. TUC implementation: Lessons learned and future steps. In *Proc. SMART NETS Final Conference – Deliverable 25*, 2003.

[91] B. Park, C. Messer, and T. Urbanik II. Enhanced genetic algortithm for signal timing optimization of oversaturated intersections. In *Proc. 79th Annual Meeting of the Transportation Research Board*, 2000.

[92] I. Porche, M. Sampath, R. Sengupta, Y.-L. Chen, and S. Lafortune. A decentralized scheme for real-time optimization of traffic signals. In *Proc. 1996 IEEE International Conference on Control Applications*, September 1996.

[93] Quadstone Ltd. PARAMICS: Microscopic traffic simulation. URL: http://www.paramics-online.com/, May 2003.

[94] D. Quinn. A review of queue management strategies. HETS, 1992.

[95] P.I. Richards. Shockwaves on the highway. *Operations Research*, 4:42–51, 1956.

[96] S. G. Ritchie. A knowledge-based decision support architecture for advanced traffic management. *Transportation Research Part A*, 24(1):27–37, January 1990.

[97] D.I. Robertson. TRANSYT: A traffic network study tool. Report LR 253, Road Research Laboratory, 1969.

[98] D.I. Robertson and R.D. Bretherton. Optimal control of an intersection for any known sequence of vehicle arrivals. In *Proc. 2nd IFAC-IFIP-IFORS Symposium of Traffic Control and Transport Systems*, 1974.

[99] D.I. Robertson and R.D. Bretherton. Optimizing networks of traffic signals in real time — The SCOOT method. *IEEE Transactions on Vehicular Technology*, 40(1):11–15, February 1991.

[100] A. W. Sadek, M. J. Demetsky, and B. L. Smith. Case-based reasoning for real-time traffic flow management. *Computer-Aided Civil and Infrastructure Engineering*, 14(5):347–356, September 1999.

[101] S. Sen and K.L. Head. Controlled optimization of phases at an intersection. *Transportation Science*, 3:5–17, 1997.

[102] S.G. Shelby. Single intersection evaluation of real-time adaptive traffic signal control algorithms. In *Proc. 84rd Annual Meeting of the Tranportation Research Board (TRB'04)*, Januari 2004.

[103] K.E. Stoffers. Scheduling of traffic lights - a new approach. *Transportation Research*, 2:199–234, 1968.

[104] S.R. Sunkari, R.J. Engelbrecht, and K.N. Balke. Evaluation of advanced coordination features in traffic signal controllers. FHWA/TX-05/0-4657-1, FHWA, September 2004.

[105] H. Taale. Optimising traffic signal control with evolutionary algorithms. In *Proc. 7th World Congress on Intelligent Transport Systems (ITS'00)*, November 2000.

[106] Texas Transportation Institute. *PASSER^{TM}III-98 Application and User's Guide*, 1998.

[107] TRB. *Highway Capacity Manual*. Transportation Research Board, National Research Council, Washington DC., 2000.

[108] TSS. AIMSUN NG: The integrated traffic environment. URL: http://www.aimsun.com/, August 2007.

[109] R. van Katwijk, B. De Schutter, and H. Hellendoorn. Look-ahead traffic adaptive control of a single intersection — A taxonomy and a new hybrid algorithm. In *Proc. 9th TRAIL Congress 2006 — TRAIL in Motion — CD-ROM*, Rotterdam, The Netherlands, November 2006.

[110] R.T. van Katwijk, B. De Schutter, and J. Hellendoorn. Traffic adaptive control of a single intersection: A taxonomy of approaches. In *Proc. 11th IFAC Symposium on Control in Transportation Systems*, pages 227–232, Delft, The Netherlands, August 2006.

[111] R.T. van Katwijk, B. De Schutter, and J. Hellendoorn. Look-ahead traffic-adaptive signal control. In *Proc. 6th European Congress on Intelligent Transport Systems and Services (ITS'07)*, Aalborg, Denmark, June 2007. Paper 2417.

[112] R.T. van Katwijk and P. van Koningsbruggen. Coordination of traffic management instruments using agent technology. *Transportation Research Part C*, 10:455–471, 2002.

[113] R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn. A test bed for multi-agent control systems in road traffic management. In P.H.L. Bovy, editor, *Proc. 8th TRAIL Congress 2004 — A World of Transport, Infrastructure and Logistics — Selected Papers*, pages 309–329, Rotterdam, The Netherlands, November 2004.

[114] R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn. A test bed for multi-agent control systems in road traffic management. In *Proc. 84th Annual Meeting of the Transportation Research Board*, Washington, DC, January 2005. Paper 05-0774.

[115] R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn. A test bed for multi-agent control systems in road traffic management. In F. Klügl, A. Bazzan, and S. Ossowski, editors, *Applications of Agent Technology in Traffic and Transportation*, Whitestein Series in Software Agent Technologies, pages 113–131. Birkhäuser Verlag, Basel, 2005.

[116] R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn. Test bed for multiagent control systems in road traffic management. *Transportation Research Record*, 1910:108–115, 2005.

[117] H.J. van Zuylen. The estimation of turning flows on a junction. *Traffic Engineering and Control*, 11:539–541, 1979.

[118] H.J. van Zuylen and L.G. Willumsen. The most likely trip matrix estimated from traffic counts. *Transportation Research Part B*, 14:281–293, 1980.

[119] S. Venglar, P. Koonce, and T. Urbanik II. *PASSER V*. Texas Transportation Institute, 2000.

[120] F.V. Webster. Traffic signal settings. *Road Research*, 39, 1958. Her Majesty's Stationery Office, London, England.

[121] F.V. Webster and B.M. Cobbe. Traffic signals. *Road Research*, 56, 1966. Her Majesty's Stationery Office, London, England.

[122] IVER werkgroep "Signaalgroepafwikkeling". Signaalgroepafwikkeling detectieconfiguratie. 02-01, IVER, 2002. In Dutch.

[123] M. Westerman and I. Hoeve. Pilot reistijd DRIP A13 - reistijdschatter. report Inro-VVG 1996-22, TNO, October 1996. In Dutch.

[124] M. Westerman, R. van Katwijk, R. Krikke, and W. Spit. Pilot reistijd DRIP A13 - validatie en tuning. Inro-VVG 1997-01, TNO, February 1997. In Dutch.

[125] Wikipedia. Traffic light — wikipedia, the free encyclopedia, 2007.

[126] S.C. Wirasinghe. Determination of traffic delays from shock wave analysis. *Transportation Research*, 12:343–348, 1978.

[127] Makoto Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-Agent Systems*. Springer Series on Agent Technology. Springer, 1998.

[128] H. Zhang and S. G. Ritchie. Real-time decision-support system for freeway management and control. *Journal of Computing in Civil Engineering*, 8(1):35–51, 1994.

Appendix A

The test environment

A.1 Introduction

To allow experimentation with different strategies for the application of multi-agent systems for dynamic traffic management and to examine their applicability a test bed was created. The test bed facilitates the development of multi agent systems for dynamic traffic management. The main requirements of the test bed are:

- The traffic managing multi-agent system can be configured easily.
- The traffic control logic of traffic engineers can be implemented easily, if possible by the traffic engineers themselves.
- The traffic managing multi-agent system can be evaluated in a realistic simulated traffic environment.
- The traffic managing multi-agent system can be easily transferred to a real-world application.

This section discusses how the test bed is set up. In traffic control, two processes can be distinguished. First, there is the traffic process. This process can be observed by means of monitoring equipment (e.g., induction loop detectors and floating-car data) and influenced by traffic control instruments (e.g., variable message signs, ramp metering installations, traffic signals), which form the basis for the second process, the control process, which is, in the case of the test bed, composed of multiple interacting intelligent agents. The test bed consists of an interaction model, intelligence models, and a world model. The interaction model is used to model the interactions between the agents. The intelligence models are used to model the intelligence of the agents that collectively give shape to the traffic control process. The world model is used to represent the outside world (i.e., the traffic process). These models are presented in the next subsections. Figure A.1 shows the relations between these models.

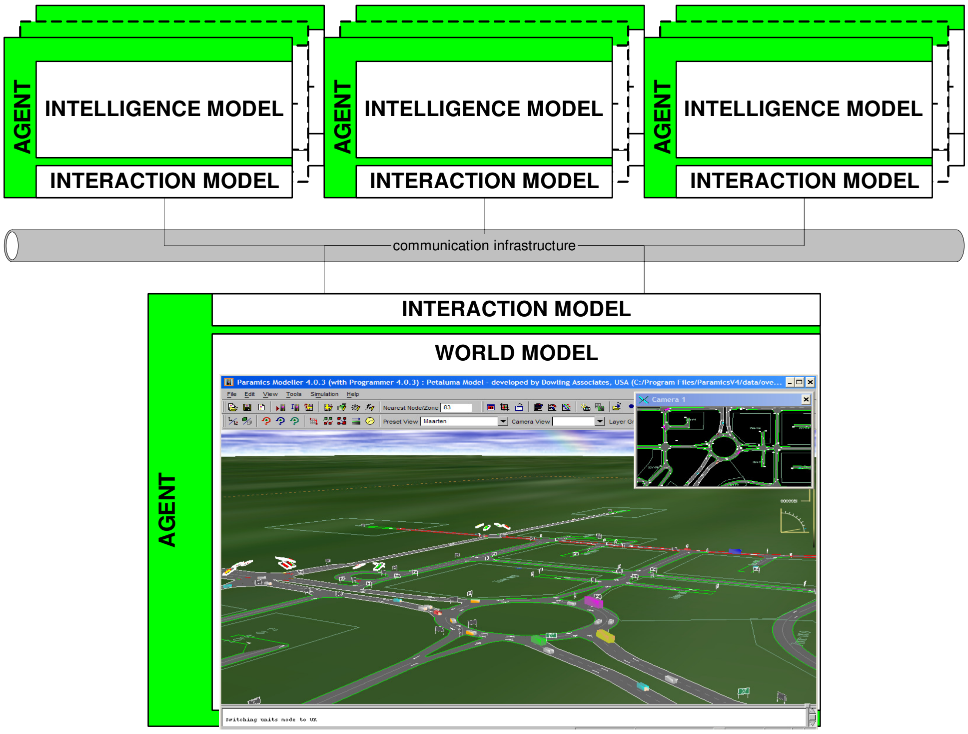


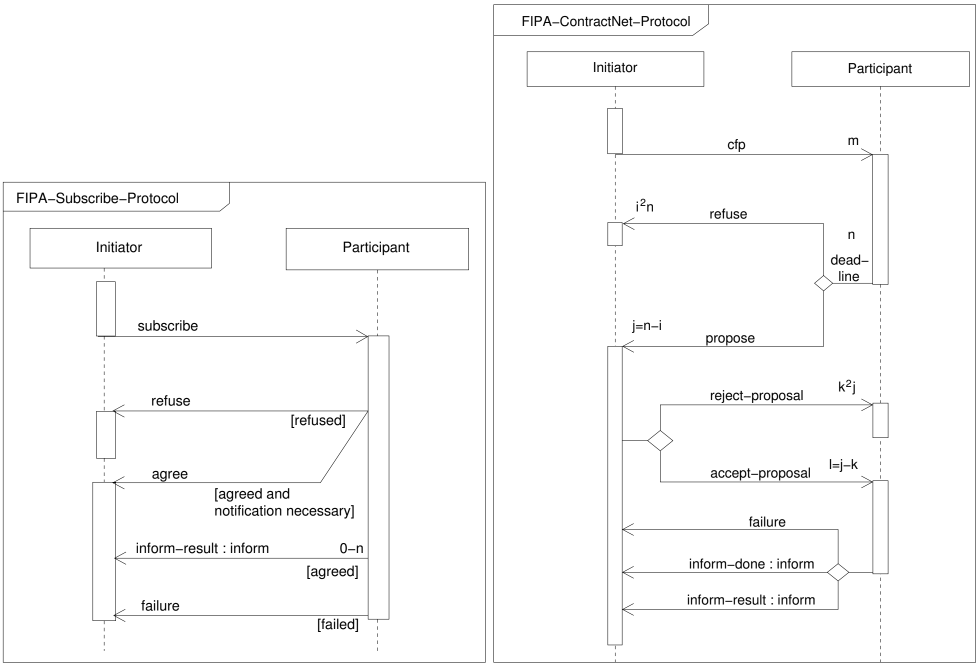
Figure A.1: Overview of the components of the test bed

A.2 Interaction model

All communications in the test bed conform to the specifications set forth by the Foundation for Intelligent Physical Agents (FIPA) [40], an approach also taken in [16] for a video-based traffic monitoring system. FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies. FIPA, the standards organization for agents and multi-agent systems was officially accepted by the IEEE as its eleventh standards committee on 8 June 2005. FIPA has adopted and is working on specifications that range from architectures to support agents' communicating with each other, communication languages and content languages for expressing those messages, and interaction protocols that expand the scope from single messages to complete transactions.

FIPA standards require that each agent publishes the services it provides to a directory facilitator. This directory facilitator is a component of the multi-agent system that provides a Yellow Pages directory service to agents. At least one directory facilitator must be present in the multi-agent system. The presence of a directory facilitator enables a dynamic configuration of the agent system. In this way, the location of a service an agent needs for its own operation does not have to be hard coded in the agent but can be found at run time through the directory facilitator.

FIPA's standard interaction protocols and communicative acts are currently sufficient for the purposes. The communication protocols of FIPA are expressed using a sequence



(a) FIPA subscription interaction protocol

(b) FIPA contract net interaction protocol

Figure A.2: FIPA interaction protocols

diagram that shows the allowed sequence of messages between agents and the constraints on the content of those messages. A sequence diagram is one of the dynamic models of the UML (Unified Modeling Language) and its graphical layout emphasizes the chronological sequence of communications between agents (For a more detailed discussion of sequence diagrams and the UML, see [14, 15]. The sequence diagrams used by FIPA incorporate the extensions to the UML proposed in [87]. Examples of FIPA's standard interaction protocols are the subscription interaction protocol (Figure A.2(a)), contract net interaction protocol (for negotiations) (Figure A.2(b)), the propose interaction protocol and the request interaction protocol, all of which are needed for the cooperating traffic agents. For this, reliance is on the JADE agent development environment [6]. The agent development environment also provides the tools needed to evaluate the performance of the multi-agent control system with respect to communication requirements.

A.3 Intelligence model

A fundamental decision in defining a problem is deciding how to model it. The dynamic traffic management domain has always been open to unconventional approaches from the field of artificial intelligence, including evolutionary algorithms, knowledge-based systems, neural networks, and multi-agent systems [26, 65, 81, 96, 100, 105, 128]. Sometimes experience is available to aid in choosing the best paradigm. Often a paradigm is selected on the basis of the applicant's familiarity with it. This is why conventional programming paradigms are often considered first. The test bed that we have designed allows programming of the intelligence of an agent through the use of a conventional programming paradigm using the C++ and Java languages but is not limited to these languages.

Currently the test bed supports rule based inference using JESS, which is a rule-based reasoning engine [41] developed in Java. Incoming messages are converted to facts and asserted into its working memory. Derived facts describing messages to be sent are translated into corresponding FIPA messages, after which JADE takes care of their delivery. The rule-based agent is typically used to program the expertise of a human expert and is as such an ideal prototyping and training tool for traffic operators. The traffic-control logic of traffic engineers can be easily implemented in an expert system as decision rules, which take the form of simple if-then statements. A simplified example of an if-then statement as used for network traffic management is shown in Table A.1.

Incoming messages are converted and assigned to variables in the network. The set of variables that has assigned values is called evidence. The resulting expectations corresponding to messages to be sent are translated into corresponding FIPA messages, after which JADE takes care of their delivery.

A.4 Virtual world

The microscopic traffic simulation packages Paramics (developed by Quadstone) and AIMSUN NG (developed by TSS) are used to represent the world. Both simulation models simulate traffic at the level of individual vehicles. Traffic simulation models often employ a time-step-based method to simulation as opposed to a discrete-event-based method. It is possible to retrieve detector data and modify the actuators between these time steps; this is shown schematically in Figure A.3. In contrast, the agents in a multi-agent system operate in continuous time. To bridge this gap, the world interaction agent stores the request and subscriptions from other agents until it is time to continue to the next time step. The decision to go to the next time step depends on the type of synchronization one wishes to apply. Since the traffic system is simulated with a single simulation process, there is only one agent that handles all outside-world requests and subscriptions. In the real world, each detector and actuator could in principle be represented by a specific agent. This is however a theoretical deployment scenario, which will be difficult to attain in practice. Traffic control centers are often equipped with different control applications, each representing a group of detectors or actuators from one manufacturer. A more realistic deployment scenario is that these applications are retrofitted with an agent wrapper (i.e., a piece of code that acts as an interface between the original application and other agents).

Simulation can be used to test various real-world application scenarios for multi-agent systems. To test whether the configured multi-agent system will function under real-time conditions, synchronization can be performed by slowing down the simulation such that simulation time equals wall clock time. However, the test bed will typically be used to test the performance of a configured multi-agent system with respect to traffic flow. In that case, it is required that the multi-agent system gets sufficient time to formulate the control decisions and that the simulations are repeatable. To guarantee that the multi-agent system is given sufficient time to formulate the control decisions, the point at which the agents in the multi-agent system have finished formulating their control decisions must be determined.

This is done with a special-purpose agent that requires an agent to report when it wants to change its state from busy to idle. This special-purpose agent is named MAI (maintainer of agent information). When all agents have reported to be idle, and thus all information on the basis of which control decisions can be formulated has been processed by the multi-agent system, the simulation is allowed to continue.

The FIPA Propose Interaction Protocol is used to convey an intended state change from an agent to the MAI. This protocol mandates that proposals are explicitly accepted or refused. The explicit acceptance is required since there is no way to guarantee that messages from the agents about an intended state change arrive in the order they are sent. Without explicit acceptance, the simulation can sometimes be allowed to continue before the agents are finished formulating their control decisions. An example of this is shown in Figure A.4(a), where the participant message informing the MAI that it changed its state to busy (at the initiation of a conversation) arrives at a later time than the protocol initiator's message informing the MAI that it has changed its state back to idle (when the conversation has ended). Figure A.4(b) shows the same communication trace, where each proposed state change is explicitly accepted. In this case, it is guaranteed that the simulation continues only when all agents are finished formulating their control decisions.

When the MAI mandates the agents to communicate intended state changes, all agents operate according to the higher-level state chart, as shown in Figure A.5. This figure shows that the busy state is a composite state encompassing the regular state charts of the agent when it is operating in unsynchronized mode. When an agent is changing state from idle to busy and vice versa, it first enters an intermediate pre-idle or pre-busy state, where it remains until it receives an "accept proposal" message from the MAI in reply to the proposed state change.

A.5 Conclusions

To aid the ongoing research in the field, a software environment was developed for rapid development of multi-agent control systems in road-traffic management. The presented test bed will be of great value for developments in traffic management. The compliance to FIPA-standards allows us to easily configure a multi-agent system thanks to the FIPA-required directory facilitator. The compliance to FIPA-standards allows us furthermore to transfer the traffic managing multi-agent system to a real-world application more easily. The rule-based intelligence model allows us to easily model the business logic of the traffic engineers.

However, the developed system still has opportunities for further extension. A graphical user interface can be developed in which agents can be created and the multi-agent system can be configured with only a few mouse clicks. This would further accelerate the implementation of the desired multi-agent system. Extending the number of available intelligence models could be another improvement. With the test bed, a tool has been developed to study the possibilities of applying multi-agent systems in dynamic traffic management. It proved to be a good starting point for our research in decentralized traffic control.

Business rule
If there is a route named *?route* that has an alternative named *?route-alt* for which the quality of traffic flow is higher **Then** direct traffic from the former route to the latter route using the following message "Congestion on route *?route*. Please take alternative *?route-alt*."

JESS rule

```
(defrule take-action
   (RouteAlternative
      (route ?route)
      (alternative ?route_alt))
   (Route
      (name ?route)
      (quality ?quality))
   (Route
      (name ?route_alt)
      (quality ?quality_alt&:
          (> ?quality_alt ?quality))
=>
   (assert
   (VMSSignal
     (text
       "Congestion␣on␣route" ?route "."
       "Please␣take␣alternative" ?route_alt ".")))
)
```

Table A.1: Rule-based intelligence

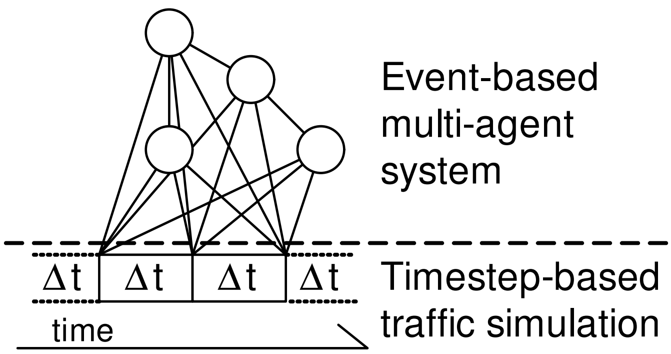
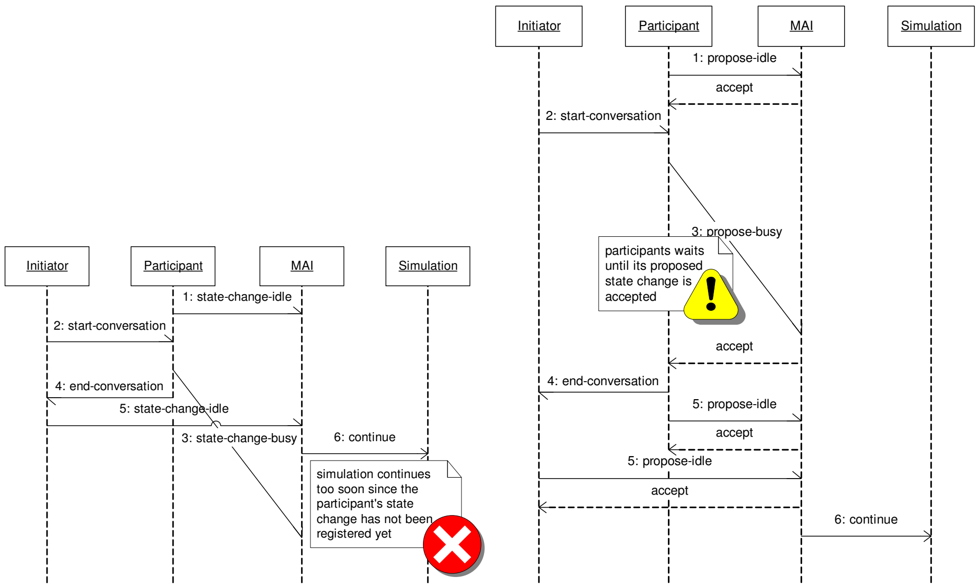


Figure A.3: Interaction in discretized time



(a) State changes are not confirmed: Unsuccessful synchronization

(b) State changes are confirmed: Successful synchronization

Figure A.4: Synchronization sequences

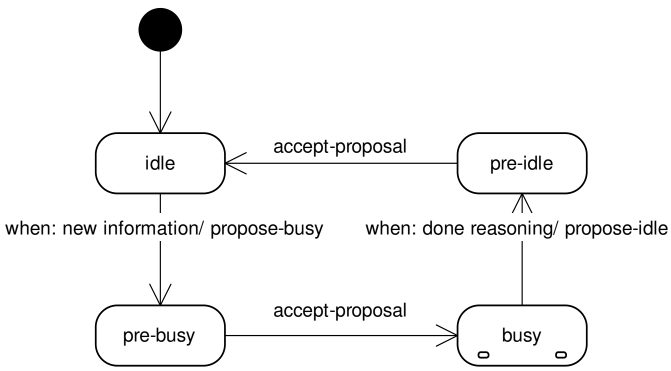


Figure A.5: States and transitions needed to ensure synchronized operation

Appendix B

Link state estimation

$v_{A,t}, v_{B,t}$	$\equiv$	Average speed calculated for location A and B respectively for the past minute expressed in kilometers per hour.
$I_{A,t}, I_{B,t}$	$\equiv$	The intensity/flow/volume for location A and B respectively for the past minute expressed in vehicles per hour.
L_{AB}	$\equiv$	The length of the link AB expressed in meters.
$N^i_{\mathrm{AB},t}$	$\equiv$	Number of vehicles present on link AB at time t according to the intensity method.
$N^\rho_{\mathrm{AB},t}$	$\equiv$	Number of vehicles present on link AB at time t according the speed method.
$N_{\mathrm{AB},t}$	$\equiv$	Number of vehicles present on link AB at time t according to the weighted combination of $N^i_{\mathrm{AB},t}$ and $N^\rho_{\mathrm{AB},t}$.
$T^i_{\mathrm{AB},t}$	$\equiv$	Travel time at link AB expressed in seconds for vehicles that enter the link at time t according to the intensity method.
$T^v_{\mathrm{AB},t}$	$\equiv$	Travel time at link AB expressed in seconds for vehicles that enter the link at time t according to the speed method.
$T_{\mathrm{AB},t}$	$\equiv$	Travel time at link AB expressed in seconds for vehicles that enter the link at time t according to the weighted combination of $T^i_{\mathrm{AB},t}$ and $T^v_{\mathrm{AB},t}$.
γ_{AB}	$\equiv$	Weighting factor for the travel time calculated for link AB that is calculated on the basis of the speed method

$$I_{A,t} = \sum_{a \in A} I_{a,t,int} \tag{B.1}$$

$N^i_{\mathrm{AB},t}$ *from measured volumes* The number of vehicles at link AB is estimated cumulatively on the basis of the measured inflow ($I_{A,t}$) and outflow ($I_{B,t}$) volume.

$$N^i_{\mathrm{AB},t} = N_{\mathrm{AB},t-1} + I_{A,t} - I_{B,t} \tag{B.2}$$

$N^\rho_{\mathrm{AB},t}$ *from derived densities* The number of vehicles at link AB is estimated on the basis of the densities derived for location A and B. The density is approximated as the quotient of the measured volume I_t and speed v_t. It is assumed that the density on the first and second half of the link is equal to the density on locations A and B respectively.

$$N^\rho_{\mathrm{AB},t} = \frac{L_{\mathrm{AB}}}{2}\left(\frac{I_{A,t}}{v_{A,t}} + \frac{I_{B,t}}{v_{B,t}}\right) \tag{B.3}$$

$N_{\mathrm{AB},t}$ as a weighted combination of $N^{i}_{\mathrm{AB},t}$ and $N^{\rho}_{\mathrm{AB},t}$

The number of vehicles at link AB that is used to estimate the travel time for link B on the basis of the intensity method, is a weighted combination of $N^{i}_{\mathrm{AB},t}$ and $N^{\rho}_{\mathrm{AB},t}$. The size of the weighting factor γ_{AB} is dependent on the geometry of the link and the actual traffic situation.

$$N_{\mathrm{AB},t} = \gamma_{\mathrm{AB}} N^{\rho}_{\mathrm{AB}} + (1-\gamma_{\mathrm{AB}}) N^{i}_{\mathrm{AB},t} \tag{B.4}$$

Travel time according to the intensity method

The intensity method assumes that the travel time is equal to the time needed by the vehicles that are present on the link (N_{AB}) to leave that link. The intensity with which vehicles are able to leave the link is equal to the outflow intensity (I_B)of the last K time intervals.

$$T^{i}_{\mathrm{AB},t} = \frac{N_{\mathrm{AB},t}}{\frac{1}{K}\sum_{i=0}^{K-1} I_{B,t-i}} \tag{B.5}$$

Travel time according to the speed method

The speed method assumes that the travel speed on the first and second half of the link is equal to the speeds method at location A and B respectively.

$$T^{v}_{\mathrm{AB},t} = \frac{L_{\mathrm{AB}}}{2}\left(\frac{1}{v_{A,t}} + \frac{1}{v_{B,t}}\right) \tag{B.6}$$

Samenvatting

Aanleiding

Verkeersregelinstrumenten worden ingezet om een veilig en efficiënt gebruik van de beschikbare infrastructuur te bewerkstelligen. Aangezien het aanbod van infrastructuur over het algemeen geen gelijke tred houdt met de toename van de mobiliteit en verkeersvraag, wordt een steeds groter beroep gedaan op verkeersregelinstrumenten om de beschikbare infrastructuur maximaal te benutten.

Het merendeel van de ingezette verkeersregelinstrumenten functioneert volledig autonoom. Lokale verstoringen in de verkeersafwikkeling worden afgehandeld zonder dat hierbij een beroep hoeft te worden gedaan op een hoger niveau van verkeersmanagement. Lokale verstoringen kunnen hierdoor tijdig en effectief worden afgehandeld. Naarmate het aantal verkeersregelinstrumenten echter groeit, wordt de kans steeds groter dat deze gaan interfereren. Verkeersregelinstrumenten kunnen dan niet meer afzonderlijk worden beschouwd, maar moeten worden beschouwd als onderdeel van een groter netwerk. Door naburige verkeersregelinstrumenten op elkaar af te stemmen, wordt voorkomen dat deze elkaar hinderen en bewerkstelligd dat deze als geheel beter functioneren.

Om het effectieve functioneren van het netwerk als geheel te kunnen bewerkstellingen, wordt vaak gebruik gemaakt van een gecentraliseerde ofwel een top-down benadering van verkeersmanagement. In theorie staat deze aanpak een netwerkoptimale instelling van de ingezette verkeersregelinstrumenten toe. Als gevolg van de complexiteit van de ingezette instrumenten en de frequentie waarmee de instellingen gewijzigd moeten kunnen worden, is dit in de praktijk niet te realiseren.

Aanpak

Doordat een gecentraliseerde aanpak als gevolg van de rekenkundige complexiteit, de hoeveelheid benodigde communicatie en de beperkte schaalbaarheid in de praktijk niet realiseerbaar is, wordt in dit proefschrift een gedistribueerde aanpak ontwikkeld. Een gedistribueerde aanpak kan goeddeels tegemoetkomen aan de tekortkomingen van een gecentraliseerd systeem.

Een gedistribueerde aanpak vereist dat het regelprobleem wordt opgedeeld in verscheidene losjes gekoppelde of onafhankelijke deelproblemen, zodat het samenstel van alle oplossingen van de deelproblemen tezamen de oplossing van het originele regelprobleem benadert. De term agent wordt gebruikt om een intelligente actor aan te duiden, welke door middel van sensoren en actuatoren interacteert met zijn omgeving. Een multi-agent systeem op zijn beurt is een systeem dat is opgebouwd uit verschillende agenten, die gezamenlijk in staat zijn doelstellingen te realiseren die moeilijk realiseerbaar zijn door een enkele agent of monolithisch systeem.

In de in dit proefschrift gekozen aanpak wordt ieder element van het netwerk gerepresenteerd door een agent. Het multi-agent systeem dat zo wordt gevormd, vormt een compleet "schaduw"-netwerk van het echte netwerk. Om eventuele negatieve effecten van de acties van de ene regelagent op de andere te voorkomen, dienen de acties van naburige regelagenten te worden gecoördineerd. Deze coördinatie kan op twee verschillende niveaus plaatsvinden: op microscopisch niveau, waarbij individuele voertuigaankomsten worden beschouwd en op macroscopisch niveau, waarbij alleen de aankomstvolumes worden beschouwd.

Lokale regeling

Teneinde de tot dusver lokaal opererende verkeersregelinstrumenten ontvankelijk te maken voor de belangen van naburige verkeersregelinstrumenten is een verruiming van het blikveld van de bestaande instrumenten vereist. De aandacht in dit proefschrift is daarbij vooral uitgegaan naar één van de oudste en meest gangbare verkeersregelinstrumenten waarover de wegbeheerder kan beschikken, het verkeerslicht.

Verkeerslichten zijn geëvolueerd van redelijk eenvoudige regelinstallaties met een starre regelcyclus tot complexe installaties die gebruik maken van meerdere detectoren op de aanvoerende wegen om de duur van de groentijd te bepalen. Het grootste deel van de Nederlandse regelingen zijn voertuigafhankelijk. Voertuigafhankelijke regelingen verlengen de groentijd voor de huidige richting afhankelijk van de aanwezigheid van voertuigen op de betreffende richting. Een beperking van dit type regeling is dat bij het besluit tot verlenging het verkeer op de andere, niet-groen zijnde, richtingen niet in beschouwing wordt genomen.

Verkeersafhankelijke regelingen verschillen van voertuigafhankelijke regelingen doordat zij een set van regelacties beschouwen en hun besluit bepalen op basis van de condities op het gehele kruispunt. Een vooruitziende regelaar is aanvullend in staat een besluit te nemen op basis van een langere termijn analyse waarin informatie van verder stroomopwaarts kan worden meegenomen. Dit stelt een vooruitziende regeling in staat om betere beslissingen voor de langere termijn te nemen.

Hoewel verkeersafhankelijke regelingen in tegenstelling tot de voertuigafhankelijke regeling een besluit nemen op basis van de condities op het complete kruispunt, is tot nog toe voor verkeerafhankelijke regelingen nog steeds de fase de kleinste regeleenheid en worden de richtingen niet afzonderlijk beschouwd. Als gevolg hiervan is het voor een verkeersafhankelijke regeling niet mogelijk om richtingen uit een volgende fase eerder groen te geven wanneer het groen voor conflicterende richtingen uit de huidige fase eerder beëindigd kan worden.

In dit proefschrift wordt een nieuw algoritme gepresenteerd voor de vooruitziende verkeersafhankelijke regeling van een kruispunt. Beslissingen worden genomen op basis van een langere termijn analyse. Deze langere termijnanalyse impliceert de zoektocht naar een optimale reeks van regelacties in een zoekboom die gevormd wordt door de selectie van regelacties over een tijdshorizon. Het algoritme beschouwt de richtingen afzonderlijk, waardoor het mogelijk wordt om richtingen uit een volgende fase groen te geven wanneer het groen voor conflicterende richtingen uit de vorige fase eerder beëindigd kan worden. Teneinde de resulterende zoekboom op een efficiënte wijze te doorzoeken bouwt het algoritme voort op het dit moment best presterende algoritme op basis van dynamisch programmeren.

Op basis van het ontwikkelde algoritme is het mogelijk de acties van verkeerslichten op microscopisch niveau, dat wil zeggen op het niveau van individuele voertuigaankomsten, te coördineren.

Coördinatie

De in dit proefschrift ontwikkelde procedure om de acties van de agenten te coördineren omvat een proces waarin agenten hun plannen iteratief bijstellen in reactie op de informatie die zij ontvangen van agents stroomopwaarts en agents stroomafwaarts. De coördinatieprocedures zijn zodanig vormgegeven dat zij coördinatie op verschillende niveaus toestaan. Aangezien verkeersregelinstrumenten op verschillende niveaus opereren, zijn ook de niveaus waarop coördinatie plaats kan vinden anders.

Coördinatie op microscopisch niveau Coördinatie op microscopisch niveau vindt plaats om overeenstemming te bereiken over de vertrek- en aankomsttijden van individuele voertuigen. Coördinatie op dit niveau is vooral van belang voor die delen van het netwerk die geregeld worden door verkeerslichten zodat zich groene golven kunnen vormen die flexibel inspringen op de veranderingen in het verkeer. Hiervoor is het vereist dat de voor een verkeerslicht verantwoordelijke agent zich bewust is van het effect van het handelen van de stroomopwaartse en stroomafwaartse agenten op zijn eigen prestatie en vice versa van het effect van het eigen handelen op de prestatie van de stroomafwaartse en stroomopwaartse agenten.

Om te voorkomen dat vertraging die stroomopwaarts voorkomen wordt verder stroomafwaarts teniet gedaan wordt, is hiervoor het vooruitziende verkeersafhankelijke algoritme zodanig uitgebreid dat tevens de verkeersprestatie stroomafwaarts wordt meegenomen in het besluitvormingsproces. De agenten wisselen hiervoor iteratief informatie uit om zodoende te komen tot een gedragen beslissing.

Het voordeel van dit type coördinatie ten opzichte van de wijze waarop dit moment verkeerslichten worden gecoördineerd, is dat deze zich aanpast aan de feitelijke verkeersomstandigheden.

Coördinatie op macroscopisch niveau Coördinatie op macroscopisch niveau vindt plaats om het volume van de geplande uitstroom onderling te coördineren. Coördinatie op macroscopisch niveau heeft tot doel om de oververzadiging van kwetsbare delen van het netwerk te voorkomen. Normaliter worden kwetsbare onderdelen van het netwerk beschermd door een set van vaste beslisregels die de instroom naar de kwetsbare delen van het netwerk beperken wanneer aan bepaalde condities wordt voldaan. In het schaduwnetwerk van agenten, houdt iedere agent voor zich in de gaten of hun deel van het netwerk onderbenut of juist overbenut is. Doordat de individuele agenten hun geplande uitstroom onderling coördineren, kan verkeer vroegtijdig worden omgeleid.

Ten einde te voorkomen dat verkeerafwikkeling instort op kwetsbare delen van het netwerk dient de instroom van verkeer richting deze delen van het netwerk te worden beperkt. De ontwikkelde coördinatieprocedure maakt hiervoor gebruik van twee typen beperkingen: (1) harde en (2) zachte randvoorwaarden. Met behulp van de harde randvoorwaarden wordt afgedwongen dat het verkeersvolume dat het kwetsbare deel van het netwerk instroomt het gewenste volume niet overschrijdt. De zachte randvoorwaarden worden ingezet om agenten verder stroomopwaarts in het netwerk te bewegen om verkeer van het kwetsbare gebied weg te sturen. Hierdoor kan de verkeersdruk op stroomafwaartse agenten die gestart zijn met het tegenhouden van verkeer wordt verlicht.

Bijdrage

In dit proefschrift is een gedistribueerde, multi-agent, aanpak voor verkeersregeling uitgewerkt. De belangrijkste bijdragen van dit proefschrift zijn:

- de ontwikkelde coördinatieprocedures op basis waarvan regelacties op zowel microscopisch en macroscopisch niveau kunnen worden afgestemd;
- het ontwikkelde vooruitziende verkeersafhankelijke kruispuntregelalgoritme dat de richtingen op het kruispunt afzonderlijk beschouwt en qua snelheid beter presteert dan bestaande benaderingen van vooruitziend verkeersafhankelijk regelen;
- de ontwikkelde taxonomie van bestaande benaderingen van vooruitziend verkeersafhankelijk regelen op basis waarvan de fundamenten van de verschillende systemen eenvoudig vergeleken kunnen worden;
- de ontwikkelde testomgeving waarmee de ontwikkelde multi-agent benadering is getest en geanalyseerd en waarmee ook geëxperimenteerd kan worden met andere verkeersregelconcepten.

Summary

Motivation

Traffic control instruments are used to realize a safe and efficient usage of the available infrastructure. Since the supply of infrastructure generally does not keep the same pace as the growth in mobility and traffic demand, a growing appeal is done on traffic control instruments to maximally utilize the available infrastructure.

The majority of the traffic control instruments functions fully autonomously. Local disturbances in the traffic flow are dealt with without having to appeal to a higher level of traffic management. Local disturbances can therefore be dealt with more timely and effectively. As more and more instruments are deployed, the probability of interference between control tools that are applied in the same area increases. Traffic control instruments can then no longer be considered separately, but have to be considered as being part of a larger network. By coordinating neighboring traffic control instruments, it can be prevented that they hinder one another and can be accomplished that they function better as a whole.

In order to be able to guarantee the effective functioning of the network as whole, often a centralized or top-down approach to traffic management is employed. Theoretically this allows for a network-optimal setting of each traffic control instrument. As a consequence of the complexity of the instruments deployed and the frequency with which adjustments have to be made, it is infeasible in practice.

Approach

As a centralized control approach is often not feasible in practice due to computational complexity, communication overhead, and lack of scalability, a distributed control approach is developed in this thesis. A distributed approach can solve the shortcomings of the centralized approach to a large extent.

A distributed approach requires that the control problem can be subdivided into several loosely coupled or independent subproblems, such that the combination of all the solutions of the subproblems together approximates the solution of the original control problem. The term agent is used to denote an intelligent actor, which interacts with his environment through means of sensors and actuators. A multi-agent system, in turn, is a system that is built up from different agents that together are capable to attain objectives that are hard to realize by a single agent or monolithic system.

In the approach chosen in this thesis each element of the network is presented by an agent. The multi-agent system that is thus formed, forms a complete “shadow”-network of

the real network. To prevent possible negative effects of actions of one control agent on another, the actions of neighboring control actions have to be coordinated. This coordination can take place on two different levels: on a microscopic level, where the arrivals of individual vehicles are considered, and on a macroscopic level, where just the arrival volumes are considered.

Local control

In order to make the thus far locally operating traffic control instruments susceptible to the needs of neighboring traffic control instruments it is necessary to broaden the horizon of the existing instruments. This thesis primarily focuses on one of the oldest and most common traffic control instrument, the traffic light.

Traffic lights have evolved from fairly simple installations that operate under a fixed-time regime to complex installations that determine the length of the green period using information from multiple detectors located at the approaches to the intersection. The majority of the traffic signal installations in The Netherlands are of the vehicle-actuated type. Vehicle-actuated controllers extend the green time for a movement dependent on the presence of vehicles for that movement. A limitation of this type of control is in the decision to extend the green time traffic on the other, non-green movements are not taken into consideration.

Traffic-adaptive controllers differ from vehicle-actuated controllers because they consider a set of control actions and come to a decision based on the conditions on the entire intersection. A look-ahead traffic-adaptive controller is additionally capable to determine its decision on the basis of a long term analysis in which information from further upstream can be incorporated. This enables a look-ahead traffic-adaptive controller to make better decisions for the long term.

Although traffic-adaptive controllers as opposed to vehicle-actuated control are able to make a decision on the basis of the conditions on the entire intersection, the phase is the smallest unit of control considered by current traffic-adaptive controllers. A traffic-adaptive controller is till thus far does not consider the individual movements. As a consequence, a traffic-adaptive controller is unable to allow a movement from a subsequent phase to turn green when the green for conflicting movements of the previous phase can be terminated earlier.

In this thesis a new algorithm is presented for the look-ahead traffic-adaptive control of an intersection. Decisions are made on the basis of a longer term analysis. This longer term analysis implies the search for an optimal sequence of control actions in a search tree that is formed by the selection of control actions over a time horizon. The algorithm considers the movements separately as a result of which it is possible to allow a movement from a subsequent phase to turn green when the green for conflicting movements of the previous phase can be terminated earlier. In order to efficiently search the resulting search-tree, the algorithm extends the currently best performing algorithm which is based on dynamic programming.

On the basis of the developed algorithm it is possible to coordinate the actions of traffic lights on a microscopic level that is on the level of the individual vehicle arrivals.

Coordination

De procedures developed in this thesis to coordinate the actions of the agents is comprised of a process in which agents iteratively adjust their plans in reaction to the information they receive from agents upstream and downstream. The coordination procedures are modeled such that they allow coordination on different levels. Since traffic control instruments operate on different levels, the levels on which coordination can take place are also different.

Coordination on the microscopic level Coordination on the microscopic level takes place in order to reach agreement on the departure times and arrival times of individual vehicles. Coordination on this level is primarily of importance for those parts of the network that are controlled by traffic lights so that green waves can be formed that flexibly adapt to changes in traffic. This requires that the agent responsible for the traffic light is aware of the effect of the actions of the agents downstream and upstream on its own performance and vice versa of the effect of its own actions on the performance of the agents downstream and upstream.

In order to prevent that delay that is prevented upstream is annulled further downstream, the developed look-ahead traffic-adaptive algorithm is extended such that the downstream performance is also taken into account in the decision process. For this, the agents iteratively exchange information in order to reach agreement.

The advantage of this type of coordination is that with regard how traffic lights are currently coordinated, this one adapts to the current traffic situation.

Coordination on the macroscopic level Coordination on the macroscopic level takes place in order to reach agreement on the volumes of the planned outflows. The objective of coordination on the macroscopic level is to prevent the oversaturation of vulnerable parts of the network. Normally the vulnerable pasts of the network are protected by a set of fixed decision rules that limit the inflow of traffic to vulnerable parts of the network when certain conditions are satisfied. In the shadow-network of agents, every agent keeps an eye on their part of the network in order to ascertain whether it is underutilized or overutilized. Because individual agents coordinate their planned outflow with one another, traffic can be rerouted early.

In order to prevent that the traffic flow collapses on vulnerable parts of the network, the inflow of traffic to these parts of the network has to be constrained. The developed coordination procedure uses two types of constraints: (1) hard constraints and (2) soft constraints. The hard constraints are used to enforce that the traffic volume that enters the vulnerable part of the network does not exceed the desired volume. The soft constraints are used to persuade the agents upstream in the network to steer traffic away from the vulnerable area. This way the pressure on downstream agents that have started to detain traffic from entering the vulnerable area can be relieved.

Contribution

In this thesis a multi-agent framework for the distributed and coordinated control of a traffic network is developed. The main theoretical and innovative contributions of this thesis with respect to the current state of the art are:

- the developed coordination procedures on the basis of which control actions can be coordinated on both the microscopic and the macroscopic level;

- the developed look-ahead traffic-adaptive intersection control algorithm that is able to consider the movements on the intersection individually and with regard to speed outperforms existing approaches to look-ahead traffic-adaptive control;
- the developed taxonomy of current approaches to look-ahead traffic-adaptive control on the basis of which the foundations of the different system scan be easily compared;
- the developed test bed on the basis of which the developed multi-agent approach has been tested and analyzed and which can also be used to experiment with different traffic control concepts.

About the author

Ronald van Katwijk (1973) obtained his MSc in Computer Science at the Delft University of Technology, faculty of Computer Science, The Netherlands. As part of his specialization in Knowledge Based Systems his Master's Thesis involved the creation of an Expert System for the classification of anomalies as found in the infrared reflections of the Dutch Railway's overhead wires. After his graduation he has been working as a researcher at TNO, the Netherlands Organisation for Applied Scientific Research, since 1997. At TNO he has been responsible for the development of both microscopic and macroscopic traffic and transport models, real-time data analysis and processing algorithms (e.g. for forecasting travel time based on detector loop and floating car data), and various decision support systems. He furthermore contributed to the creation of various architectures amongst which the European RIS (River Information System) architecture and the Dutch Architecture for Traffic Management.

The Dutch Architecture for Traffic Management defines a structured approach through which the cooperating road operators can formulate their joint objectives and ultimately determine the operational traffic management instruments needed to attain these objectives. His research has since focused on how to further operationalize the Dutch Architecture of Traffic Management (i.e., how to bridge the gap between traffic management and traffic control). It was during his work on TNO and the projects that he was involved in that he developed the basic ideas and notions that inspired him to pursue a PhD on the subject of multi-agent traffic control. This ultimately led in 2003 to a part-time position at the Delft Center for Systems of Control of the Delft University of Technology, where he has been working on traffic control, multi-agent systems, and optimization.

Articles

International journal papers

- B. De Schutter, and J. Hellendoorn, "Test bed for multiagent control systems in road traffic management", Transportation Research Record, no. 1910, pp. 108-115, 2005.
- R.T. van Katwijk, and P. van Koningsbruggen, "Coordination of Traffic Management Instruments Using Agent Technology", Transportation Research Part C, 2002.

Book contributions:

- R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn, "A test bed for multi-agent control systems in road traffic management", in Applications of Agent Technology in Traffic and Transportation (F. Klügl, A. Bazzan, and S. Ossowski, eds.), Whitestein Series in Software Agent Technologies, Basel: Birkhäuser Verlag, ISBN 3-7643-7258-3, pp. 113-131, 2005.

International conference papers:

- R.T. van Katwijk, B. De Schutter, and J. Hellendoorn, "Look-ahead traffic-adaptive signal control", Proceedings of the 6th European Congress on Intelligent Transport Systems and Services (ITS'07), Aalborg, Denmark, 12 pp., June 2007. Paper 2417.
- R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn, "Traffic Adaptive Control of Single Intersection: A Taxonomy of Approaches", Proceedings of the 11th IFAC Symposium on Control in Transportation Systems, Delft, The Netherlands, 2006.
- R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn, "A test bed for multi-agent control systems in road traffic management", Proceedings of the 84th Annual Meeting of the Transportation Research Board, Washington, DC, 22 pp., Jan. 2005. Paper 05-0774.
- R.T. van Katwijk, B. De Schutter, J. Hellendoorn, P. van Koningsbruggen, and S.P. van Gosliga, "A test bed for multi-agent control systems in road traffic management", presented at the Workshop on Agents in Traffic and Transportation (Workshop W7) at the 3rd International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS-2004), New York, New York, Jul. 2004.
- P.H . van Koningsbruggen, R.T. van Katwijk, and S.P. van Gosliga, "Operational Traffic Management: A continuous negotiation on three levels", ITS Madrid, 2003
- R.T.van Katwijk, F. van der Putten, and E. Kerckhoffs, "The Application of an Expert System for the Inspection and Maintenance of Control Wires", ESM Istanbul, 1997

National conference papers:

- R. van Katwijk, B. De Schutter, and H. Hellendoorn, "Look-ahead traffic adaptive control of a single intersection - A taxonomy and a new hybrid algorithm" Proceedings of the 9th TRAIL Congress 2006 - TRAIL in Motion - CD-ROM, Rotterdam, The Netherlands, 14 pp., Nov. 2006.
- R.T. van Katwijk, P. van Koningsbruggen, B. De Schutter, and J. Hellendoorn, "A test bed for multi-agent control systems in road traffic management", Proceedings of the 8th TRAIL Congress 2004 - A World of Transport, Infrastructure and Logistics - Selected Papers (P.H.L. Bovy, ed.), Rotterdam, The Netherlands, pp. 309-329, Nov. 2004.
- A.Th. van den Bosch, M.R. Menken, M. van Breukelen, and R.T. van Katwijk, "A Testbed for Multi-Agent Systems and Road Traffic Management", BNAIC, 2003

National journal papers:

- R.T. van Katwijk, "Een netwerk van zelforganiserende maatregelen: Intelligent Agents", NM-magazine 1, 2006

TRAIL Thesis Series

A series of The Netherlands TRAIL Research School for theses on transport, infrastructure and logistics.

Nat, C.G.J.M., van der, *A Knowledge-based Concept Exploration Model for Submarine Design*, T99/1, March 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Westrenen, F.C., van, *The Maritime Pilot at Work: Evaluation and Use of a Time-to-boundary Model of Mental Workload in Human-machine Systems*, T99/2, May 1999, TRAIL Thesis Series, Eburon, The Netherlands

Veenstra, A.W., *Quantitative Analysis of Shipping Markets,* T99/3, April 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Minderhoud, M.M., *Supported Driving: Impacts on Motorway Traffic Flow*, T99/4, July 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Hoogendoorn, S.P., *Multiclass Continuum Modelling of Multilane Traffic Flow*, T99/5, September 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Hoedemaeker, M., *Driving with Intelligent Vehicles: Driving Behaviour with Adaptive Cruise Control and the Acceptance by Individual Drivers*, T99/6, November 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Marchau, V.A.W.J., *Technology Assessment of Automated Vehicle Guidance - Prospects for Automated Driving Implementation*, T2000/1, January 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Subiono, *On Classes of Min-max-plus Systems and their Applications*, T2000/2, June 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Meer, J.R., van, *Operational Control of Internal Transport*, T2000/5, September 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Bliemer, M.C.J., *Analytical Dynamic Traffic Assignment with Interacting User-Classes: Theoretical Advances and Applications using a Variational Inequality Approach,* T2001/1, January 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Muilerman, G.J., *Time-based logistics: An analysis of the relevance, causes and impacts,* T2001/2, April 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Roodbergen, K.J., *Layout and Routing Methods for Warehouses*, T2001/3, May 2001, TRAIL Thesis Series, The Netherlands

Willems, J.K.C.A.S., *Bundeling van infrastructuur, theoretische en praktische waarde van een ruimtelijk inrichtingsconcept*, T2001/4, June 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Binsbergen, A.J., van, J.G.S.N. Visser, *Innovation Steps towards Efficient Goods Distribution Systems for Urban Areas,* T2001/5, May 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Rosmuller, N., *Safety analysis of Transport Corridors*, T2001/6, June 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Schaafsma, A., *Dynamisch Railverkeersmanagement, besturingsconcept voor railverkeer op basis van het Lagenmodel Verkeer en Vervoer*, T2001/7, October 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Bockstael-Blok, W., *Chains and Networks in Multimodal Passenger Transport. Exploring a design approach*, T2001/8, December 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Wolters, M.J.J., *The Business of Modularity and the Modularity of Business*, T2002/1, February 2002, TRAIL Thesis Series, The Netherlands

Vis, F.A., *Planning and Control Concepts for Material Handling Systems*, T2002/2, May 2002, TRAIL Thesis Series, The Netherlands

Koppius, O.R., *Information Architecture and Electronic Market Performance,* T2002/3, May 2002, TRAIL Thesis Series, The Netherlands

Veeneman, W.W., *Mind the Gap; Bridging Theories and Practice for the Organisation of Metropolitan Public Transport*, T2002/4, June 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Van Nes, R., *Design of multimodal transport networks, a hierarchical approach,* T2002/5, September 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Pol, P.M.J., *A Renaissance of Stations, Railways and Cities, Economic Effects, Development Strategies and Organisational Issues of European High-Speed-Train Stations*, T2002/6, October 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Runhaar, H., *Freight transport: at any price? Effects of transport costs on book and newspaper supply chains in the Netherlands*, T2002/7, December 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Spek, S.C., van der, *Connectors. The Way beyond Transferring*, T2003/1, February 2003, TRAIL Thesis Series, Delft University Press, The Netherlands

Lindeijer, D.G., *Controlling Automated Traffic Agents*, T2003/2, February 2003, TRAIL Thesis Series, Eburon, The Netherlands

Riet, O.A.W.T., van de, *Policy Analysis in Multi-Actor Policy Settings. Navigating Between Negotiated Nonsense and Useless Knowledge*, T2003/3, March 2003, TRAIL Thesis Series, Eburon, The Netherlands

Reeven, P.A., van, *Competition in Scheduled Transport*, T2003/4, April 2003, TRAIL Thesis Series, Eburon, The Netherlands

Peeters, L.W.P., *Cyclic Railway Timetable Optimization*, T2003/5, June 2003, TRAIL Thesis Series, The Netherlands

Soto Y Koelemeijer, G., *On the behaviour of classes of min-max-plus systems*, T2003/6, September 2003, TRAIL Thesis Series, The Netherlands

Lindveld, Ch.D.R., *Dynamic O-D matrix estimation: a behavioural approach*, T2003/7, September 2003, TRAIL Thesis Series, Eburon, The Netherlands

Weerdt, M.M., de, *Plan Merging in Multi-Agent Systems,* T2003/8, December 2003, TRAIL Thesis Series, The Netherlands

Langen, P.W, de, *The Performance of Seaport Clusters*, T2004/1, January 2004, TRAIL Thesis Series, The Netherlands

Hegyi, A., *Model Predictive Control for Integrating Traffic Control Measures*, T2004/2, February 2004, TRAIL Thesis Series, The Netherlands

Lint, van, J.W.C., *Reliable Travel Time Prediction for Freeways*, T2004/3, June 2004, TRAIL Thesis Series, The Netherlands

Tabibi, M., *Design and Control of Automated Truck Traffic at Motorway Ramps*, T2004/4, July 2004, TRAIL Thesis Series, The Netherlands

Verduijn, T. M., *Dynamism in Supply Networks: Actor switching in a turbulent business environment*, T2004/5, September 2004, TRAIL Thesis Series, The Netherlands

Daamen, W., *Modelling Passenger Flows in Public Transport Facilities*, T2004/6, September 2004, TRAIL Thesis Series, The Netherlands

Zoeteman, A., *Railway Design and Maintenance from a Life-Cycle Cost Perspective: A Decision-Support Approach*, T2004/7, November 2004, TRAIL Thesis Series, The Netherlands

Bos, D.M., *Changing Seats: A Behavioural Analysis of P&R Use*, T2004/8, November 2004, TRAIL Thesis Series, The Netherlands

Versteegt, C., *Holonic Control For Large Scale Automated Logistic Systems*, T2004/9, December 2004, TRAIL Thesis Series, The Netherlands

Wees, K.A.P.C. van, *Intelligente voertuigen, veiligheidsregulering en aansprakelijkheid. Een onderzoek naar juridische aspecten van Advanced Driver Assistance Systems in het wegverkeer*, T2004/10, December 2004, TRAIL Thesis Series, The Netherlands

Tampère, C.M.J., *Human-Kinetic Multiclass Traffic Flow Theory and Modelling: With Application to Advanced Driver Assistance Systems in Congestion*, T2004/11, December 2004, TRAIL Thesis Series, The Netherlands

Rooij, R.M., *The Mobile City. The planning and design of the Network City from a mobility point of view*, T2005/1, February 2005, TRAIL Thesis Series, The Netherlands

Le-Anh, T., *Intelligent Control of Vehicle-Based Internal Transport Systems*, T2005/2, April 2005, TRAIL Thesis Series, The Netherlands

Zuidgeest, M.H.P., *Sustainable Urban Transport Development: a Dynamic Optimization Approach*, T2005/3, April 2005, TRAIL Thesis Series, The Netherlands

Hoogendoorn-Lanser, S., *Modelling Travel Behaviour in Multimodal Networks*, T2005/4, May 2005, TRAIL Thesis Series, The Netherlands

Dekker, S., *Port Investment - Towards an integrated planning of port capacity*, T2005/5, June 2005, TRAIL Thesis Series, The Netherlands

Koolstra, K., *Transport Infrastructure Slot Allocation*, T2005/6, June 2005, TRAIL Thesis Series, The Netherlands

Vromans, M., *Reliability of Railway Systems*, T2005/7, July 2005, TRAIL Thesis Series, The Netherlands

Oosten, W., *Ruimte voor een democratische rechtsstaat. Geschakelde sturing bij ruimtelijke investeringen*, T2005/8, September 2005, TRAIL Thesis Series, Sociotext, The Netherlands

Le-Duc, T., *Design and control of efficient order picking*, T2005/9, September 2005, TRAIL Thesis Series, The Netherlands

Goverde, R., *Punctuality of Railway Operations and Timetable Stability Analysis*, T2005/10, October 2005, TRAIL Thesis Series, The Netherlands

Kager, R.M., *Design and implementation of a method for the synthesis of travel diary data*, T2005/11, October 2005, TRAIL Thesis Series, The Netherlands

Boer, C., *Distributed Simulation in Industry*, T2005/12, October 2005, TRAIL Thesis Series, The Netherlands

Pielage, B.A., *Conceptual Design of Automated Freight Transport Systems*, T2005/14, November 2005, TRAIL Thesis Series, The Netherlands

Groothedde, B., *Collaborative Logistics and Transportation Networks, a modeling approach to network design*, T2005/15, November 2005, TRAIL Thesis Series, The Netherlands

Valk, J.M., *Coordination among Autonomous Planners*, T2005/16, December 2005, TRAIL Thesis Series, The Netherlands

Krogt, R.P.J. van der, *Plan Repair in Single-Agent and Multi-Agent Systems*, T2005/17, December 2005, TRAIL Thesis Series, The Netherlands

Bontekoning, Y.M., *Hub exchange operations in intermodal hub-and-spoke networks. A performance comparison of four types of rail-rail exchange facilities*,T2006/1, February 2006, TRAIL Thesis Series, The Netherlands

Lentink, R., *Algorithmic Decision Support for Shunt Planning*, T2006/2, February 2006, TRAIL Thesis Series, The Netherlands

Ngoduy, D., *Macroscopic Discontinuity Modeling for Multiclass Multilane Traffic Flow Operations*, T2006/3, April 2006, TRAIL Thesis Series, The Netherlands

Vanderschuren, M.J.W.A., *Intelligent Transport Systems for South Africa. Impact assessment through microscopic simulation in the South African context*, T2006/4, August 2006, TRAIL Thesis Series, The Netherlands

Ongkittikul, S., *Innovation and Regulatory Reform in Public Transport*, T2006/5, September 2006, TRAIL Thesis Series, The Netherlands

Yuan, J., *Stochastic Modelling of Train Delays and Delay Propagation in Stations*, T2006/6, October 2006, TRAIL Thesis Series, The Netherlands

Viti, F., *The Dynamics and the Uncertainty of Delays at Signals*, T2006/7, November 2006, TRAIL Thesis Series, The Netherlands

Huisken, G., *Inter-Urban Short-Term Traffic Congestion Prediction*, T2006/8, December 2006, TRAIL Thesis Series, The Netherlands

Feijter, R. de, *Controlling High Speed Automated Transport Network Operations*, T2006/9, December 2006, TRAIL Thesis Series, The Netherlands

Makoriwa, C., *Performance of Traffic Networks. A mosaic of measures*, T2006/10, December 2006, TRAIL Thesis Series, The Netherlands

Miska, M., *Microscopic Online Simulation for Real time Traffic Management*, T2007/1, January 2007, TRAIL Thesis Series, The Netherlands

Chorus, C., *Traveler Response to Information*, T2007/2, February 2007, TRAIL Thesis Series, The Netherlands

Weijermars, W.A.M., *Analysis of Urban Traffic Patterns Using Clustering*, T2007/3, April 2007, TRAIL Thesis Series, The Netherlands

Zondag, B., *Joined Modeling of Land-use, Transport and Economy*, T2007/4, April 2007, TRAIL Thesis Series, The Netherlands

Bok, M.A. de, *Infrastructure and Firm Dynamics: A micro-simulation approach*, T2007/5, May 2007, TRAIL Thesis Series, The Netherlands

Fiorenzo-Catalano, M.S., *Choice Set Generation in Multi-Modal Transportation Networks*, T2007/6, June 2007, TRAIL Thesis Series, The Netherlands

Vlist, P. van der, *Synchronizing the retail supply chain*, T2007/7, June 2007, TRAIL Thesis Series, The Netherlands

Joksimovic, D., *Dynamic Bi-Level Optimal Toll Design Approach for Dynamic Traffic Networks*, T2007/8, September 2007, TRAIL Thesis Series, The Netherlands

Warffemius, P.M.J., *Modeling the Clustering of Distribution Centers around Amsterdam Airport Schiphol. Location Endowments, Economies of Agglomeration, Locked-in Logistics and Policy Implications*, T2007/9, September 2007, TRAIL Thesis Series, The Netherlands

Driel, C.J.G. van, *Driver Support in Congestion: an assessment of user needs and impacts on driver and traffic flow*, T2007/10, November 2007, TRAIL Thesis Series, The Netherlands

Gietelink, O.J., *Design and Validation of Advanced Driver Assistance Systems*, T2007/11, November 2007, TRAIL Thesis Series, The Netherlands

Nuttall, A.J.G., *Design Aspects of Multiple Driven Belt Conveyors*, T2007/12, November 2007, TRAIL Thesis Series, The Netherlands

Nederveen, A.A.J., *Ruimtelijke Inpassing van Lijninfrastructuur. Een onderzoek naar de geschiktheid van inspraakreacties voor het ontwerpen van lijninfrastructuur*, T2007/13, December 2007, TRAIL Thesis Series, The Netherlands

Negenborn, R.R., *Multi-Agent Model Predictive Control with Applications to Power Networks*, T2007/14, December 2007, TRAIL Thesis Series, The Netherlands

Houtenbos, M., *Expecting the Unexpected. A study of interactive driving behaviour at intersections*, T2008/1, January 2008, TRAIL Thesis Series, The Netherlands

Argiolu. R., *Office Location Choice Behaviour and Intelligent Transport Systems*, T2008/2, January 2008, TRAIL Thesis Series, The Netherlands

Katwijk, R.T. van, *Multi-Agent Look-ahead Traffic Adaptive Control*, T2008/3, January 2008, TRAIL Thesis Series, The Netherlands