

OPTIMIZING PRE-TIMED CONTROL AT ISOLATED INTERSECTIONS

STIJN FLEUREN

Optimizing pre-timed control at isolated intersections

Stijn Fleuren



Research School for Operations
Management and Logistics

© Stijn Fleuren, 2016

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-4209-3

Printed by: Ridderprint

Cover design: Kimberley Teunissen

This thesis is number D 201 of the thesis series of the Beta Research School for Operations
Management and Logistics

Optimizing pre-timed control at isolated intersections

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit
Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens,
voor een commissie aangewezen door het College voor Promoties, in het
openbaar te verdedigen op donderdag 19 januari 2017 om 16:00 uur

door

Stijn Theodorus Gerardus Fleuren

geboren te Nijmegen

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter: prof.dr. L.P.H. de Goey
1e promotor: prof.dr.ir. I.J.B.F. Adan
2e promotor: prof.dr.ir H. Nijmeijer
copromotor: dr.ir. A.A.J. Lefeber
leden: prof.dr.ir. S.P. Hoogendoorn (Technische Universiteit Delft)
 prof.dr. R. Núñez (Universiteit van Amsterdam)
 prof.dr. T. van Woensel
 prof.dr. O.J. Boxma

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Acknowledgements

It has been a great pleasure for me to conduct research at the Technical University of Eindhoven during the past four years. The results of this research would not have been possible without the help and support of others. Therefore, I would like to express gratitude to a number of people.

First, I would like to thank my coach and co-promotor Erjen Lefeber. We worked together for almost seven years: first my bachelors final project, followed by my graduation project and finally my Ph.D. project. During that time you have been a true mentor, motivator and role model. I will remember our fruitful discussions when a complex mathematical problem was not going to be solved easily. Especially the times that my gut feeling told me something was probably true, but I was not yet able to formally prove it. We would then alternate between trying to find a satisfying proof and, if we were not able to find one, try to disprove the statement. These discussions brought me great joy even when (after hours of thinking and discussing) we found out that my gut-feeling was wrong.

Furthermore, I would like to express gratitude to my first promotor Ivo Adan for providing an excellent working environment within the Manufacturing Networks group, which enabled me to develop my skills in scientific writing, teaching and performing research. I also want to thank you for thoroughly reading this thesis, your valuable comments, enthusiasm, and optimism.

I also want to thank my second promotor Henk Nijmeijer. You became my second promotor in an unconventional manner; through a reorganization of the faculty Mechanical Engineering you became responsible for yet another Ph.D. student. Thank you (and also Ivo Adan and Erjen Lefeber) for ensuring that this transition did not affect me, which allowed me to continue to perform research in the same manner as before. I also like to express gratitude to you for reading this thesis and for providing me with valuable feedback, which led to a significantly smaller and more readable thesis.

Moreover, I would like to thank the other members of my doctorate committee: Serge Hoogendoorn, Sindo Nuñez Queija, Onno Boxma, and Tom van Woensel. Thank you all for accepting our invitation, for reading this thesis with eagerness, and for your useful

feedback.

I would also like to thank Marko Boon and Yoni Nazarathy. You both were not directly involved in my PhD project, but through your supervision in earlier projects, your positivity, enthusiasm and knowledge you both had an invaluable impact on me and my scientific career.

Next I would like to express appreciation to my (former) colleagues at the Manufacturing networks group and the Systems Engineering group for the pleasant working environment. Special thanks to my (former) office mates (Allan, Lennart, Thomas, Tim, Aida, Martijn and Dirk) for the positive working vibe and, of course, for the 'freaky fridays'. I would also like to thank all students with whom I worked on their bachelor final project, internship or graduation project: Stefan Backhuijs, Joey Hendriks, Jeremy Karouta, Xander Koolen, Emile Manni, Yorick van der Schriek and Jan van der Vleuten.

Many thanks to my friends for the necessary distractions during my spare time. Thank you for all the enjoyable moments: the trips to Spain, the burgers and pubquizzes at 'De Wildeman', and the evenings at 'De Bock' and 't cafeetje'. Special thanks to my brother Wouter. One could not wish for a better brother and friend. I would also like to thank my parents for being supportive and loving, not only during the past years but during my entire life. Without you this result was not possible! Last but certainly not least I would like to thank Alja. Words cannot describe how thankful I am for the loving environment at home and your patience and understanding, which allowed me to work late (regularly) whenever I felt the pressure to finish this thesis in time.

Stijn Fleuren
November 2016

List of symbols

The next list describes many of the symbols used in this thesis.

Graphs

- C Cycle-arc incidence vector, page 52
- \mathcal{C} Cycle $\mathcal{C} \subseteq A$ in some graph $G = (V, A)$, page 48
- \mathcal{C}^- Arcs that the cycle $\mathcal{C} \subseteq A$ traverses in the backward direction (from head to tail), page 48
- \mathcal{C}^+ Arcs that the cycle $\mathcal{C} \subseteq A$ traverses in the forward direction (from tail to head), page 48
- d Cyclomatic number, page 52
- M, M_C Cycle-arc incidence matrix, page 106
- M_P Path-arc incidence matrix, page 108
- $\nu(G)$ Number of connected components of graph G , page 52
- P A polyhedron or a path arc-incidence vector, page 108
- \mathcal{P} Path $\mathcal{P} \subseteq A$ in some graph $G = (V, A)$, page 48
- \mathcal{P}^- Arcs that path $\mathcal{P} \subseteq A$ traverses in the backward direction (from head to tail), page 48
- \mathcal{P}^+ Arcs that path $\mathcal{P} \subseteq A$ traverses in the forward direction (from tail to head), page 48
- $w_{\mathcal{B}}$ Width of the integral cycle basis \mathcal{B} , page 54
- $w_{\mathcal{C}}$ Width of the cycle \mathcal{C} , page 54

Layout of intersection

- \mathcal{A}_i Set of lane-use arrows of leg i , page 160
- $\Delta_{i,l}^a$ Binary variable that indicates whether lane l of leg i is an arrival lane, page 166
- $\Delta_{i,l,j}$ Binary variable that indicates whether lane l of leg i is equipped with lane-use arrow j , page 166
- $i(m)$ Leg at which the traffic from vehicular movement m arrives, page 161
- $j(m)$ Lane-use arrow that traffic from vehicular movement m follows, page 161
- \mathcal{L} Set of legs at the intersection, page 160
- \overline{L}_i Maximum number of lanes at leg i , page 160

$\Psi_{\mathcal{M}}$	Set of conflicting movements, page 164	
$\Psi_{\mathcal{S}}$	Set of conflicting signal groups, page 41	
\mathcal{Q}	Set of queues used to model the traffic waiting at the intersection, page 39	
\mathcal{Q}_i	Set of queues whose access to the intersection is controlled by signal group i , page 39	
\mathcal{Q}_i^v	Set of queues dedicated to vehicular traffic, page 160	
\mathcal{Q}^{pc}	Set of queues dedicated to pedestrians or cyclists, page 162	
\mathcal{S}	Set of signal groups at the intersection, page 39	
Rates		
$\lambda_{i,j}$	Arrival rate of traffic following lane-use arrow j at leg i , page 163	[PCE/s]
$\lambda_{i,l,j}$	Arrival rate of traffic following lane-use arrow j at lane l of leg i , page 167	[PCE/s]
λ_q	Arrival rate of traffic at queue q , page 39	[PCE/s]
$\mu_{i,l,j}$	Saturation flow rate of lane l of leg i when traffic follows only lane-use arrow j at this lane, page 163	[PCE/s]
μ_q	Saturation flow rate of queue q , page 39	[PCE/s]
ρ_i^{SG}	The load of signal group i , page 78	
ρ_q	The load of queue q , i.e., $\rho_q := \lambda_q/\mu_q$, page 40	
Time-related		
\textcircled{i}_k (\textcircled{i})	End of the k th (first) realization of signal group i , page 43	
\textcircled{i}_k (\textcircled{i})	Start of the k th (first) realization of signal group i , page 43	
$\underline{c}_{i,j}$	Minimum clearance time, page 42	[s]
$f(\varepsilon)$	Time expressed as a fraction of the period duration at which event ε is scheduled, page 44	
$\gamma(\varepsilon_1, \varepsilon_2)$	Time between event ε_1 and event ε_2 expressed as a fraction of the period duration, page 44	
G_i	Total green time of signal group i , page 114	[s]
g_i	Total effective green time of signal group i , page 41	[s]
$g_{i,k}$	Duration of effective green interval k of signal group i , page 85	[s]
\underline{g}_i	Minimum effective green time of signal group i , page 43	[s]
\overline{g}_i	Maximum effective green time of signal group i , page 43	[s]
r_i	Total effective red time of signal group i , page 95	[s]
r'_i	Ratio of the total effective red time to the period duration, page 95	
$r_{i,k}$	Duration of the effective red interval preceding effective green interval k of signal group i , page 83	[s]
$r'_{i,k}$	Ratio of effective red time $r_{i,k}$ to the period duration T , page 83	
\underline{r}_i	Minimum effective red time of signal group i , page 43	[s]
\underline{r}_i	Minimum effective red time of signal group i , page 43	[s]
$t^e(t_i^e)$	Ending lost time (of signal group i), page 27	[s]
$t^s(t_i^s)$	Starting lost time (of signal group i), page 27	[s]
T	Period duration of the signal group diagram, page 36	[s]
T'	Reciprocal of the period duration T , i.e., $T' := 1/T$, page 44	[1/s]
\underline{T}	Lower bound on the period duration T , page 43	[s]

\bar{T}	Upper bound on the period duration T , page 43	[s]
Y (Y_i)	Yellow time (of signal group i), page 26	[s]

Other symbols

β	Growth factor of the arrival rates at the intersection, page 60
$b_{i,k}$	Binary variable that indicates whether signal group i has k (or more) realizations, page 87
K_i	Number of realizations of signal group i , page 76
\mathcal{K}_i^d	Difference between $\bar{\mathcal{K}}_i$ and $\underline{\mathcal{K}}_i$, page 86
\underline{K}_i	Minimum number of realizations of signal group i , page 86
$\underline{\mathcal{K}}_i$	$\mathcal{K}_i := \{1, \dots, \underline{K}_i\}$, page 86
\bar{K}_i	Maximum number of realizations of signal group i , page 86
$\bar{\mathcal{K}}_i$	$\mathcal{K}_i := \{1, \dots, \bar{K}_i\}$, page 86
\mathcal{K}_i	Set of all realizations of signal group i , page 77
P	A polyhedron or a path arc-incidence vector, page 108
w_q	The weight (importance) of the traffic arriving at queue q , page 45

Contents

Acknowledgements	i
List of symbols	iii
1 Introduction	1
1.1 Road congestion and its consequences	1
1.2 Measures for relieving congestion	4
1.3 Traffic intersections and congestion	8
1.4 A signalized intersections and its terminology	8
1.5 Traffic light control	13
1.6 Focus of this thesis and its motivation	20
1.7 Goal and contributions	21
1.8 Thesis overview	22
2 Modeling traffic lights and delay	25
2.1 Effective green and effective red modes	26
2.2 Approximating the delay for a traffic flow of only passenger cars	28
2.3 Approximating the delay for a mixed traffic flow	31
2.4 Approximating the delay for a cyclist flow	32
2.5 Approximating the delay for a pedestrian flow	32
2.6 Summary	33
3 Optimizing pre-timed control: single realization	35
3.1 Introduction	35
3.2 Formulating the MIP problem	37
3.3 Comparison with existing optimization formulations	62
3.4 Summary	73
4 Optimizing pre-timed control: variable number of realizations	75
4.1 Introduction	75

4.2	Fixed number of realizations	76
4.3	Variable number of realizations	86
4.4	Numerical results	93
4.5	Summary	101
5	Integral signal group diagrams	103
5.1	Introduction	103
5.2	Preliminaries	105
5.3	Notation	110
5.4	Input data	114
5.5	Structural property	114
5.6	First step of the two-step approach	115
5.7	Second step of the two-step approach: A rounding MILP	132
5.8	Numerical results	146
5.9	Summary	152
6	Optimizing intersection layout	157
6.1	Introduction	157
6.2	Input data	160
6.3	Formulating the MILP problem	166
6.4	Numerical results	180
6.5	Summary	181
7	Usage in practice	183
7.1	Introduction	183
7.2	Additional constraints	184
7.3	Obtaining multiple signal group diagrams	191
7.4	Optimize only a part of the signal group diagram	192
7.5	Intersection analysis	195
7.6	Input parameters that depend on the optimization outcome	196
7.7	Summary	198
8	Conclusions and recommendations	199
8.1	Conclusions	199
8.2	Recommendations	202
	Bibliography	209
A	Appendices of Chapter 1	215
A.1	Automatic generation of a single-ringed phase diagram	215
A.2	Actuated control may behave as a pre-timed control	228

B Appendices of Chapter 2	233
B.1 Propagation of the back of the queue and its effects	233
C Appendices of Chapter 3	237
C.1 A motivation for negative clearance times	237
C.2 An invalid triangular inequality on the minimum clearance times	239
C.3 An invalid circuital inequality on the minimum clearance times	240
C.4 Convexity of the approximation of Van den Broek	241
C.5 Proofs of lemmas	241
D Appendices of Chapter 4	249
D.1 Convexity of the approximation of Van den Broek extended to multiple realizations	249
D.2 Queue emptying and convexity of the delay formula	250
D.3 Proofs of lemmas	252
E Appendices of Chapter 5	255
E.1 Proof of lemmas	255
E.2 Description of the polyhedron $P(u)$	257
E.3 The two-step approach for a theoretical example	260
F Appendices of Chapter 6	267
F.1 Sustainability of the optimized layout	267
F.2 Optimizing intersection layout: exclusive bus lanes	273
F.3 Advanced layout optimization	275
Summary	279
Nederlandse samenvatting	283
Curriculum Vitae	287

Introduction

This chapter describes the scope of this thesis, elaborates on its contributions, and gives an overview of all remaining chapters. In Figure 1.1 we have visualized the structure of this introduction.

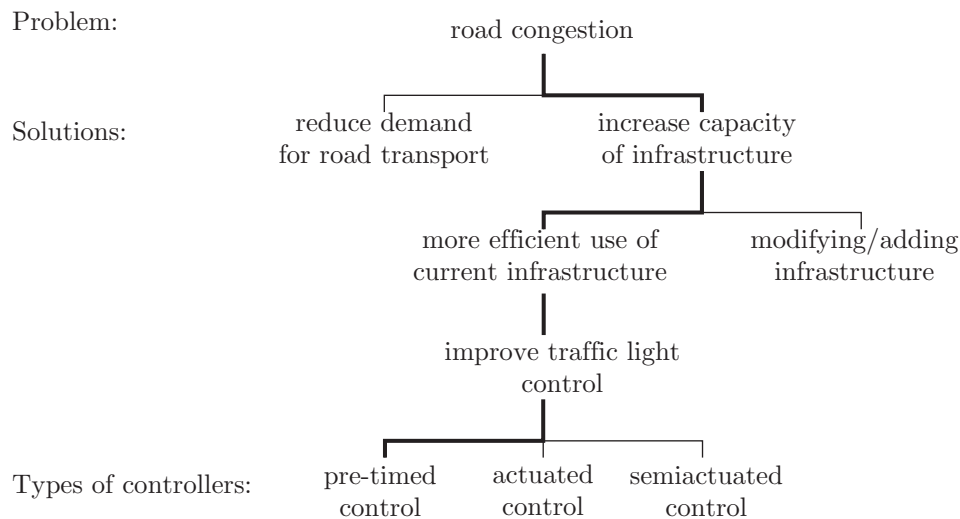


Figure 1.1: Structure of this introduction.

1.1 Road congestion and its consequences

An important ingredient of modern society is mobility. We often depend on vehicles, trains and metros as means for personal transportation, e.g., to get to work, visit friends and family, and to go on a holiday. In addition, mobility is also very important for the transportation of goods and raw materials. Products are often made in different countries (or even entirely different continents) from where they are sold. Therefore, before a product finally finds its way to ones home, it was probably subject to a lot of traveling. For example, components are delivered to a factory, the products leaving the factory are transported to a warehouse, and this warehouse replenishes local stores. The need for mobility has resulted in many different modes of transportation, e.g., aviation,

road transport and rail transport. According to (European commission, 2015), road transport is the primary mode of transportation for both passenger transport and the transportation of goods and raw materials (freight); see Figure 1.2 for the modal splits for passenger transport and freight transport.

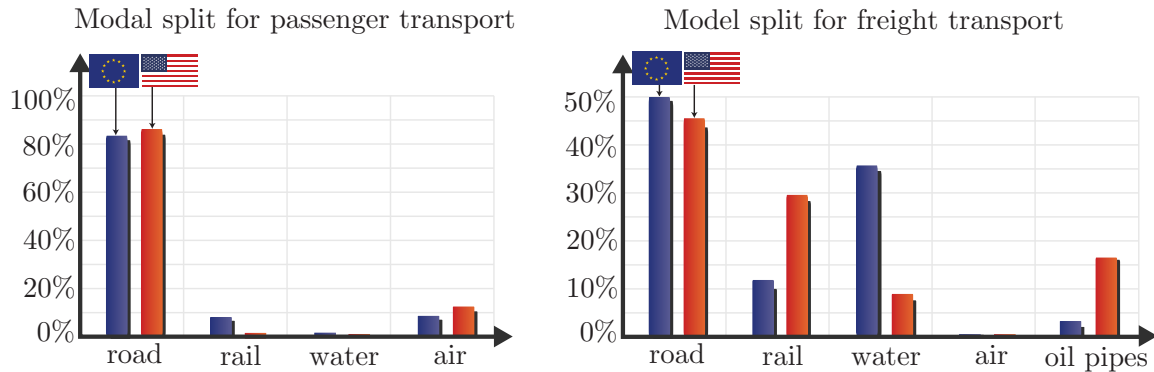


Figure 1.2: The modal split for domestic/intra-EU passenger transport (left) and domestic/intra-EU freight transport (right) in the European Union (blue) and the United States of America (red), obtained from (European commission, 2015). For passenger transport these modal splits are expressed as a percentage of the total distance traveled for passenger transport in pkm^1 and for freight transport these splits are expressed as a percentage of the total distance traveled by freight in tkm^2 .

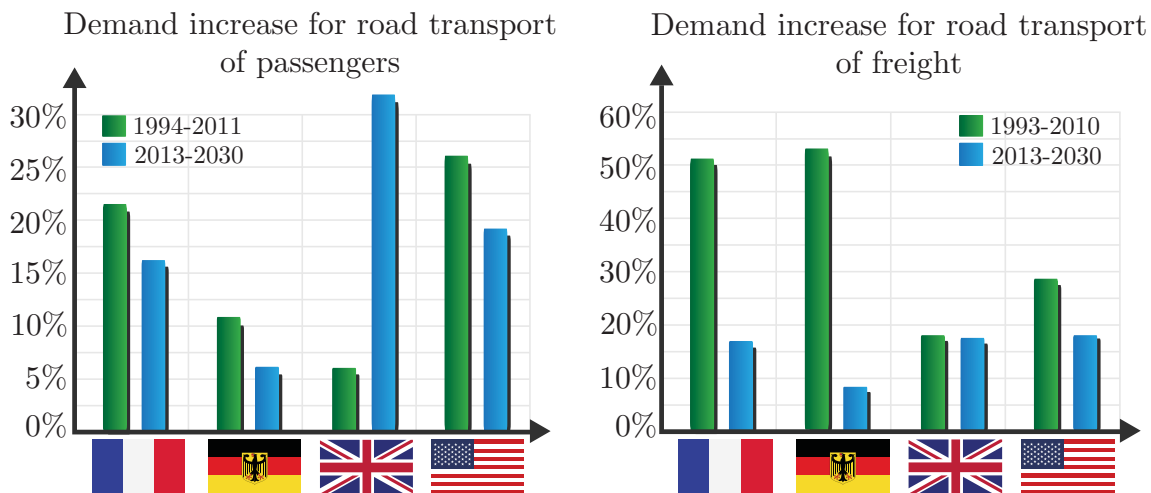


Figure 1.3: Historic and forecasted increase in demands for road transport of passengers (left) and freight (right) across four advanced countries, obtained from (INRIX, Inc., and CEBR, 2014). Demands for road (freight) transport are expressed in pkm^1 (tkm^2). Countries from left to right: France, Germany, United Kingdom, and the United States of America.

The demand for road transport is large and this demand is expected to increase even further as a consequence of population growth, economic growth, increasing income levels, and rising car ownership, amongst other factors (INRIX, Inc., and CEBR, 2014);

¹ pkm (passenger-kilometer) is a unit of measure: one passenger transported a distance of one kilometer.

² tkm (tonne-kilometer) is a unit of measure: one tonne transported a distance of one kilometer.

Figure 1.3 visualizes the historic and forecasted increase in demand for road transport for several advanced countries. Congestion of roads is a serious problem and, as a consequence of the increasing demand, this problem is expected to become even bigger. Congestion increases the time we spend on the road, it affects our daily living, the quality of the air we breathe, the price tag on the products we buy, the costs of the trips we make, et cetera. We elaborate on some of the effects of congestion:

Economic costs: The economic costs associated with congestion of roads are enormous.

Across the four advanced countries investigated in (INRIX, Inc., and CEBR, 2014) (these countries are France, Germany, the United Kingdom (UK), and the United States of America (USA)) the total economic costs from 2013 to 2030 are predicted to be 3.35 trillion euros; that is 3.35 million times a million euros! Across these four countries the annual costs accounted for ± 1440 euros per car commuting household in the year 2013 and these annual costs are predicted to be ± 1920 euro per car commuting household in the year 2030. The aforementioned economic costs are composed of direct and indirect costs. The direct costs apply (directly) to each road user. These direct costs account for the additional fuel that is wasted and the value of the additional time that is wasted on the road (time is money). The indirect costs (indirectly) affect everyone, also the people that do not travel by road. These indirect costs account for the increased cost of doing business; it is for example more costly to transport freight in these congested situations. These costs can be expected to be passed onto households through higher consumer prices; so indirectly consumers pay for these costs. The costs associated with pollution of the environment and the related health risks are not even included in the calculation of these economic costs.

Environmental costs: Traffic congestion affects, but is not limited to, climate change, deteriorating air quality, deteriorating water quality, land pollution and noise pollution. Road traffic plays a significant role in the emission of greenhouse gases and therefore also affects air quality and global warming. In the USA, over a fourth of the CO₂ emissions arise from road transport (Barth and Boriboonsomsin, 2009). These CO₂ emissions depend highly on the traffic conditions; in (Sjodin et al., 1998) it is shown that in congested situations the emissions can increase up to fourfold with respect to uncongested conditions; this increase in emissions is caused by an increase in the number of accelerations, decelerations, starts and stops. Furthermore, congestion contributes to the formation of acid rain and smog (Colville et al., 2001), which has a negative effect on aquatic life, animals, soil, trees, crops, plants, and buildings, see (Singh et al., 2007). Road traffic also contributes to water and land pollution as rain and snow running of the road has the tendency to collect gasoline, motor oil and other pollutants and drag it into nature (National Research Council (U.S.), 2009). Moreover, the noise caused by traffic might also affect wild life (Ra-

dle, 2007); it might affect animals psychology, behavior, energy budget, reproductive success, and long-term survival.

Health: The pollution of the environment also affects our physical and mental health. A causal relation exists between traffic-caused air pollution and cardiovascular disease (Hoffmann et al., 2007, 2006; Rosenlund et al., 2006; Tonne et al., 2007), asthma and other respiratory symptoms (Künzli et al., 2009; Lindgren et al., 2009; Modig et al., 2006), birth outcomes (for example low weight at birth and prenatal mortality) (Padula et al., 2012; Stieb et al., 2012; Wilhelm et al., 2012), and cancer (Apelberg et al., 2005; Grineski et al., 2013; Morello-Frosch and Jesdale, 2006). Furthermore, noise pollution may be related to sleep deprivation (Basner et al., 2011; Halonen et al., 2012), hypertension (high blood pressure) (Dratva et al., 2012; Van Kempen and Babisch, 2012), stroke (Floud et al., 2013; Sørensen et al., 2011), and cardiovascular disease (Babisch, 2000, 2008; Babisch et al., 1999). Besides the physical health, traffic congestion also affects the mental health; traffic-caused noise pollution can cause annoyance and affect performance (Basner et al., 2014), and traffic congestion itself results in increased driver stress (Hennessy and Wiesenhal, 1997; Stokols et al., 1978; Wickens and Wiesenhal, 2005).

Quality of life Traffic congestion affects the economy, the environment, and our health. As a result, traffic congestion affects the quality of life. Besides, nobody likes spending time in traffic and, as Figure 1.4 shows, the average amount of time wasted by congestion is substantial. This wasted time accounts not only for the extra time spent in traffic but also for extra 'planning time' that is needed to arrive at your destination in time; this extra planning time is larger when the travel time is uncertain.

1.2 Measures for relieving congestion

As we have seen, the congestion of roads has major consequences for the economy, the environment, our health, and the quality of life. Congestion might never be fully eliminated. However, several ways exist to relieve congestion of road networks and lessen its impact on road users. Which measures are most effective, affordable, and feasible likely depends on the local context. For example, in urban areas building new infrastructure (e.g., roads) is possibly infeasible due to the lack of space. However, in a rural area this may be an appropriate measure. Thus, not a single solution exists that works in each and every situation. Relieving congestion is an interplay between decreasing the demand for road transport and increasing the supply of road infrastructure. In this section we elaborate on a few possible measures. We do not claim this section to provide a complete overview, but merely want to convey the message that many different measures exist, and that with a combination of measures we might be able to mitigate congestion.

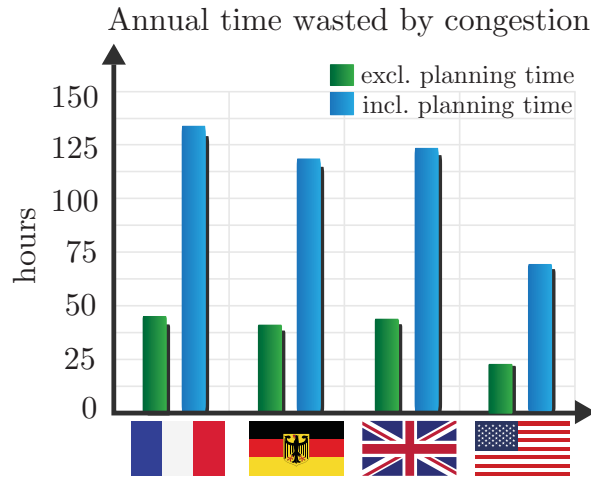


Figure 1.4: Average annual time wasted by congested roads in several advanced countries in the year 2013. The green bars represent the extra time stuck in traffic (due to congestion). In addition to the extra time spend in traffic (due to congestion), the blue bars also include the extra ‘planning time’ that is needed to reach the destination in time; this extra planning time is larger when the travel time is uncertain.

We categorize the measures using the categorization from (Amitran project, 2014). However, first, we mention some of the more obvious measures, some of which do not fit into this categorization. One possibility is to increase the supply side, e.g., by adding new infrastructure (for example building new roads) or modifying existing infrastructure (for example adding lanes to roads or adding traffic lights to intersections). Another possibility is to reduce the demand side. As a large part of commutes consists of the daily travel between home and work, promoting to live near your work might reduce demand. Moreover, promoting ride-sharing might be an effective measure. For example, with the designation of HOV³ lanes (Wu et al., 2015) and by promoting and improving services regarding ride-sharing (Furuhata et al., 2013). Moreover, demand for road transport might be reduced by promoting alternative modes of transportation (for example public transportation, cycling, and walking), e.g., by improving timetables for public transportation, developing systems that deliver personalized multi-modal trip planning (Zografos et al., 2009), and providing park-and-ride and kiss-and-ride facilities; park-and-ride facilities are also argued to possibly have the opposite effect of increasing road traffic (Mingardo, 2013) as some people traveling by car to the park-and-ride facilities used to travel by bike.

In the remainder of this section we use the categories for intelligent transport systems found in (Amitran project, 2014). The categories are as follows: navigation and traveller information, traffic management and control, demand and access management, driver behaviour change, and safety and emergency systems. Note that some measures can be categorized in multiple categories.

Tremendous progress in sensor and communications technology has made it possible to provide travelers with real-time information on network conditions. This information

³High-occupancy-vehicle (HOV) lanes: lanes that are designated to (only) vehicles carrying at least some minimum number of occupants.

enables road users to make better travel decisions, which in turn affects congestion (Papageorgiou et al., 2007). The category "navigation and traveller information" encompasses measures that inform road users either pre-trip (for example through pre-trip guidance via the Dutch websites www.vanAnaarBeter.nl and www.anwb.nl) or during their trip (for example through on-the-road guidance via variable message signs or navigation systems). Variable message signs can display information on crashes, road works, non-recurrent congestion, closed exit ramps, travel times et cetera. These systems should safely inform road users of current traffic and travel conditions and allow them to make well-informed decisions on a strategic level (e.g., route selection and congestion avoidance) as well as on a tactical level (e.g., their driving speed), while positively affecting travel conditions for the majority of users (Sharples et al., 2016). In addition to the more general information provided by variable message signs, most dynamic in-car navigation systems are capable of giving more personalized information (e.g., depending on your destination and preferences). These navigation systems may also suggest different routes to different people (driving in close proximity of each other) having the potential to reduce manifestations of Braess' paradox ⁴ (Liu et al., 2014).

Simply informing road users, however, is in many situations not enough. Traffic management and control strategies actively try to guide traffic streams, e.g., with active and smart parking management (Willson, 2015), by controlling traffic lights, regulating the access to freeways with ramp meters⁵ (Lu et al., 2011), opening the hard shoulder on highways during peak hours, and setting variable speed limits (Lu et al., 2011). According to (Djahel et al., 2015), traffic management and control consists of the following phases. First, data (such as traffic volumes, speeds, et cetera) is gathered using a wide range of sensors, e.g., wireless sensor networks, cellular networks, mobile sensing, and social media. Subsequently, this heterogeneous data is fused and aggregated to extract useful information (see also (Ou, 2011)), which is then used to perform analyses, e.g., compute optimal routes and perform short-term traffic forecasting (for example with (Van Lint et al., 2005; Vlahogianni et al., 2014)). Next, based on these analyses, traffic control actions are taken. Possible actions are, for example, closing lanes or roads, changing traffic rules (e.g., restricting trucks to use only the right-most lane), adjusting traffic light timings, setting variable speed limits et cetera (see also (Djahel et al., 2015)). In this section we also have to mention anticipatory control. Anticipatory control explicitly includes the closed interaction between traffic assignment (route choice of road users) and traffic control (e.g., changing traffic light settings) when determining the control actions, see for example (Rinaldi and Tampère, 2015; Taale and Hoogendoorn, 2013). In other words, anticipatory control explicitly accounts for the route choice of road users influencing the desired control actions, and the control action in turn influencing the route choice.

⁴Braess' paradox: adding capacity to a road network can possibly result in worse traffic conditions.

⁵Ramp meters can manage (usually by means of a traffic light) the access of vehicles to freeways by taking the current traffic situation into account; the more congested the freeway the less vehicles are allowed to enter the freeway.

Through demand and access management people are either discouraged to use specific roads, routes or areas, e.g., through road pricing⁶ and HOT lanes⁷ (Gillen, 2016), or the access to a part of the road network is regulated to maintain optimal travel conditions, e.g., with perimeter control (Geroliminis et al., 2013).

Another development is the rise of more intelligent vehicles, which is the focus of the fifth category: driver behaviour change. These smart vehicles are equipped with sensors and wireless data communication and can automatically (without help of a driver) drive within close proximity of each other, effectively forming a train of vehicles (Dey et al., 2016); reducing the space between cars may also reduce the demand for road infrastructure as the same infrastructure can be used more effectively. Smart vehicles also have the potential to be safer, resulting in less accidents (and as a consequence less accident related congestion) through, for example, the following innovations: collision warning, drowsy driver warning system, blind spot warning, lane departure warning systems, night vision systems (improving vision at night or in poor weather by projecting images on the dashboard or windshield), traffic sign recognition, and augmented awareness systems that provide information on potential danger sources, see also (Amitran project, 2014).

The latter safety measures can also be categorized in the category "safety and emergency systems". Another interesting emergency system is the eCall technology. Starting from the year 2018 it will (mandatorily) be installed in each new type of car in the European Union (EENA, 2015). This service automatically calls 112 in case of a serious accident, and, in addition, it sends information such as GPS coordinates, airbag deployment and impact information to the local emergency agencies. Besides its primary focus of increasing safety, it has the side effect of reducing accident related congestion through a decrease in response time of emergency agencies.

As previously mentioned, no 'miracle' solution exists that applies in every situation, but applying a combination of measures to reduce demand for road transport and increase the supply of road infrastructure may significantly mitigate the congestion problem. With this thesis we contribute to this mitigation; we consider the optimization of traffic light control at intersections, which fits in the category "traffic management and control"; note that with this topic we mainly affect traffic in urban areas (and not highway traffic). An advantage of improving traffic light control is that it requires no additional space to increase the capacity of road infrastructure. Since it requires no modification of the current infrastructure, it may also be a financially attractive solution.

⁶Road pricing: direct charges collected from road users for the use of roads. These charges can be based on the distance traveled, the time of day, the degree of congestion on the road, and the type of vehicle driven.

⁷HOT lanes are high occupancy vehicle (HOV) lanes that may also be used by single occupancy vehicles against a price that possibly depends on the current demand level.

1.3 Traffic intersections and congestion

Places where two or more roads meet, called intersections, play a significant role in the problem of congestion. Intersections have to be shared by different streams of traffic. As a result, road users often have to wait and yield the right of way to other road users before entering the intersection themselves. This sharing of the intersection is an important cause of congestion.

For safety reasons it must be clear when road users have to yield the right of way and when they have priority to cross the intersection. These priorities are often made explicit with signs, traffic lights, and/or road markings. If these priorities are not made explicit, (implicit) right of way rules apply, e.g., you have to yield to vehicles coming from the right.

An intersection that establishes these priorities by using traffic lights is called a *signalized intersection*. Otherwise we call the intersection unsignalized. Furthermore, we distinguish between conventional intersections and roundabouts, see Figure 1.5. Therefore, we distinguish between four types of intersections: a signalized conventional intersection, an unsignalized conventional intersection, a signalized roundabout, and an unsignalized roundabout. What type of intersection is most appropriate depends on the situation, see (Department of main roads, 2006). For example the preference for either a conventional intersection or a roundabout depends on many factors, e.g., percentage of heavy vehicles, distribution of traffic amongst the different roads, percentage of traffic making a left turn, available geometry, et cetera. Roundabouts perform best for roads with roughly similar traffic flows, where the percentage of heavy vehicles is low (heavy vehicles may have difficulty manoeuvring a roundabout), and where a large proportion of traffic makes a right turn (and therefore occupies the roundabout for a short amount of time), see (Department of main roads, 2006). However, conventional intersections may be preferred when the roads have very different traffic flows, when the percentage of heavy vehicles is large, and a large proportion makes a left turn, see (Department of main roads, 2006). Also the preference for either a signalized or an unsignalized intersection depends on the situation. An unsignalized intersection performs well whenever little to moderate amounts of traffic arrive at it. However, a signalized intersection seems to be superior when the amount of traffic arriving at the intersection is moderate to large.

As signalized intersections are best capable of sustaining large amounts of traffic, these are especially important in the fight against congestion. In this thesis we consider the problem how to *control* the traffic lights at *conventional intersections*, i.e., when should these traffic lights switch to green, yellow, and red?

1.4 A signalized intersections and its terminology

In Figure 1.6 we have depicted a typical Dutch intersection that is equipped with traffic lights. We use this intersection to introduce some terminology.

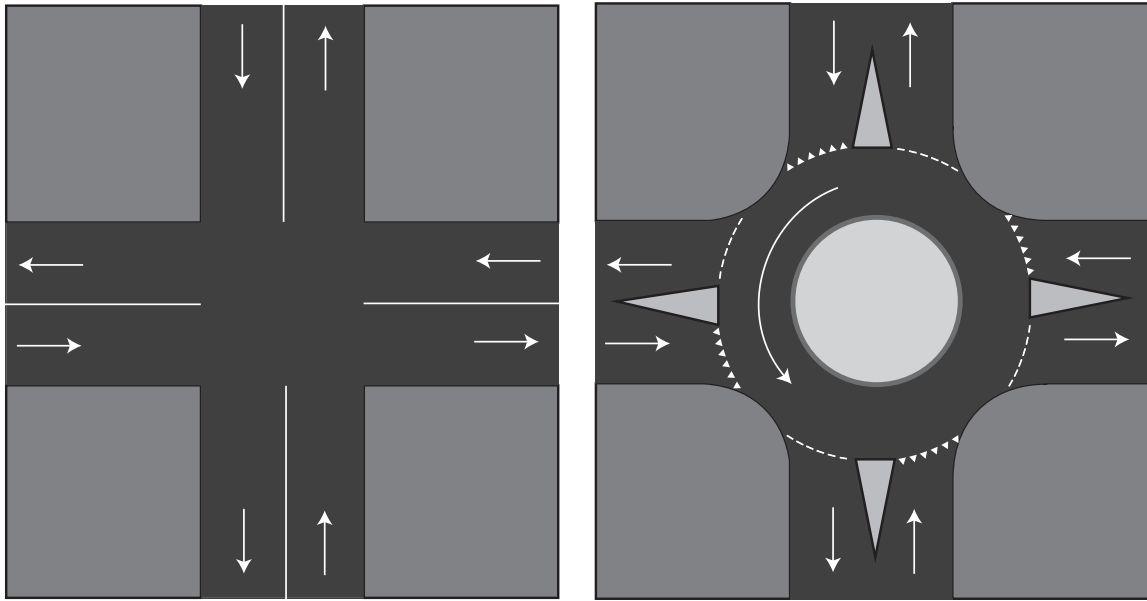


Figure 1.5: Two types of intersections: A conventional intersection (left) and a roundabout (right).

1.4.1 Lanes and legs

All sorts of traffic may arrive at an intersection, e.g., vehicular traffic, cyclists, and pedestrians. Vehicular traffic is restricted to the use of *lanes*; each such lane is intended for the use of a single line of vehicles. We distinguish between *arrival lanes* and *departure lanes*. As their names imply, vehicles approach the intersection via the arrival lanes and they leave the intersection via the departure lanes. Each arrival lane is accommodated with *lane-use arrows*, which indicate in which direction the vehicles on this arrival lane are allowed to drive. Several adjacent lanes together form a *leg*; a leg comprises all arrival lanes on which vehicles approach the intersection from a specific direction and all departure lanes on which vehicles leave the intersection in this specific direction. The intersection in Figure 1.6 has four legs. For this intersection cyclists and pedestrians are restricted to the use of bike paths respectively side walks.

1.4.2 Movements

Several streams of traffic, which we call movements, have to share the intersection. We distinguish between vehicular movements, pedestrian movements and cyclist movements.

Vehicular movements

Each vehicular movement is specified by the leg on which the traffic flow approaches the intersection and the leg on which the traffic flow leaves the intersection. For example, three vehicular movements originate from the northern leg: one going to the eastern leg (left-turn movement), one going to the southern leg (through movement), and one going to the western leg (right-turn movement). In total this intersection has 12 vehicular

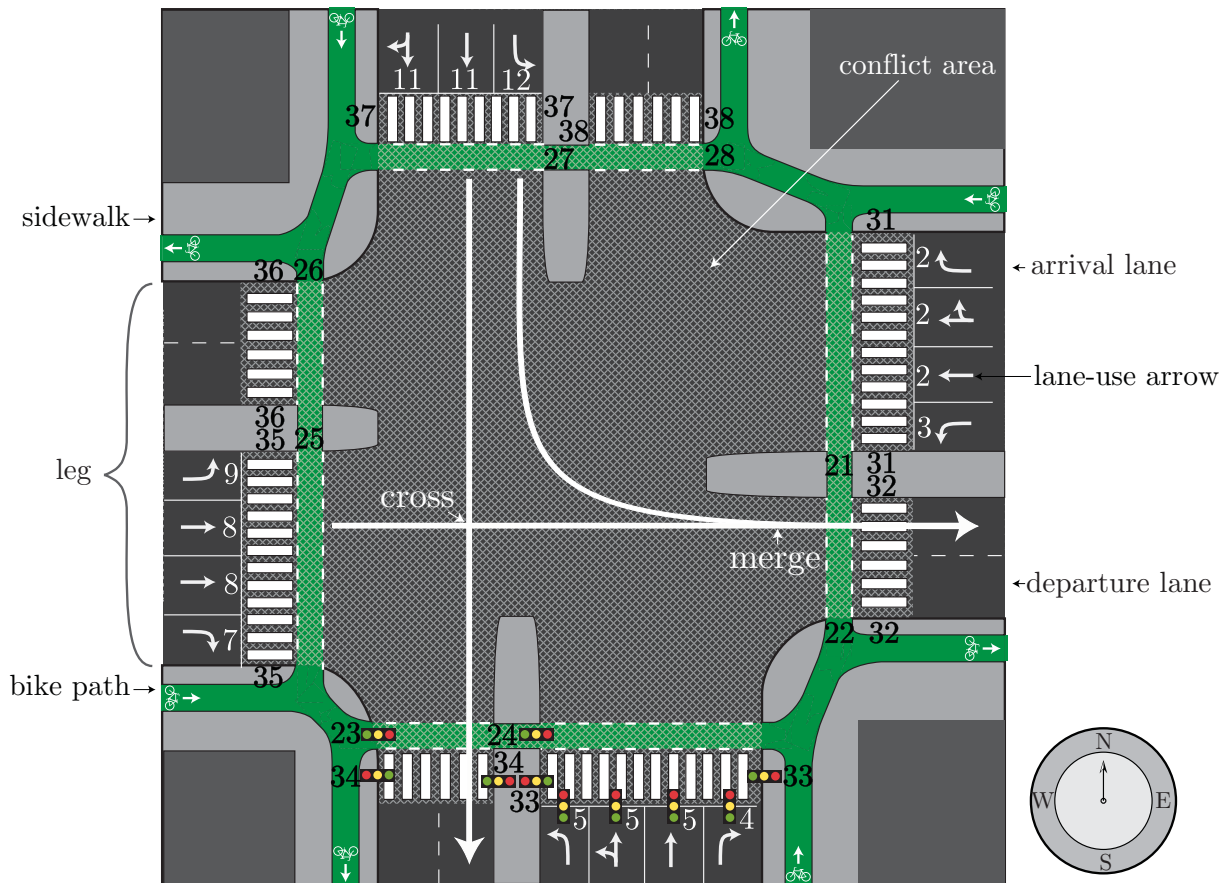


Figure 1.6: An example of a signalized intersection with vehicular traffic, pedestrians and cyclists. The numbers visualize in which signal groups the traffic lights are partitioned; the traffic lights that are partitioned in the same signal group receive identical indications. For reasons of clarity, we have visualized only the traffic lights positioned at the southern leg; for all other traffic lights we have only visualized the number of its signal group. The hatched area visualizes the conflict area.

movements (each leg has a left-turn movement, a through movement, and a right-turn movement). Note that multiple lanes may accommodate the same movement, e.g., the eastern leg has two lanes that accommodate a through movement. Also a lane may accommodate multiple movements, e.g., the eastern leg has a lane that accommodates a right-turn movement and a through movement.

Pedestrian movements and cyclist movements

With each pedestrian (cyclist) crossing we associate a pedestrian (cyclist) movement. Note that each leg has two pedestrian (cyclist) crossings for the intersection in Figure 1.6; one of these crossings, crosses the departure lanes of this leg, and the other one crosses the arrival lanes of this leg. Therefore, this intersection has 8 pedestrian movements and 8 cyclist movements.

For this intersection, pedestrians are allowed to cross a leg in both directions, whereas cyclists are allowed to cross a leg in only one direction. Therefore, each pedestrian crossing

has two traffic lights while each cyclist crossing has only one traffic light.

Protected, permitted, and prohibited movements

A movement can either be protected, permitted, or prohibited. Each of these types has its own traffic light indication, see Figure 1.7. A movement is protected whenever it has the exclusive right of way. In other words, the traffic stream does not cross or merge paths with any other movement. In contrast, a permitted movement might cross or merge paths with another movement; a permitted movement is made through the gaps of the merging or crossing traffic stream. A prohibited movement is not allowed, which may be temporarily (the traffic light(s) associated with this movement has (have) a red indication) or indefinitely (for example when the corresponding lane-use arrow is not present at the intersection).

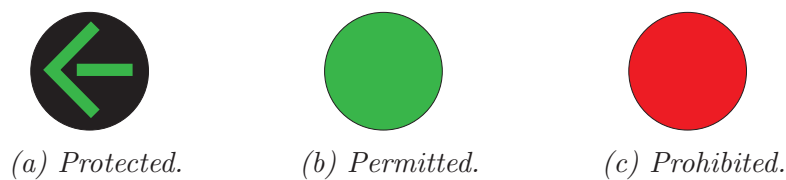


Figure 1.7: Examples of traffic light indications for protected, permitted and prohibited movements.

1.4.3 Signal groups

The task of a traffic light controller is to choose the indications (green, yellow or red) of the traffic lights so that each traffic stream can safely cross the intersection and such that road users experience as little hinder as possible. Two traffic lights associated with the same movement must receive the same indication. For example, the traffic lights positioned at both sides of a pedestrian crossing receive identical indications; when pedestrians can cross the pedestrian crossing in one direction, they can also cross this pedestrian crossing in the opposite direction. Also two adjacent lanes that have a common lane-use arrow must receive the same indication; otherwise, two vehicles headed in the same direction might receive different indications, which may be seen as 'unfair'. As a consequence, a traffic light controller usually does not choose the indication of each traffic light individually, but the traffic lights are partitioned into *signal groups*; the traffic lights partitioned in the same signal group receive identical indications at all times. For example the southern leg has two signal groups, but four traffic lights for vehicular traffic: one signal group consists of the three traffic lights associated with the left-turn movement and the through movement, and the other signal group consists of one traffic light that accommodates a right-turn movement.

There are 39 traffic lights positioned at the intersection in Figure 1.6: 15 of these traffic lights are dedicated to vehicular traffic (one traffic light for each lane), 16 are dedicated

to pedestrians (four traffic lights at each leg), and 8 are dedicated to cyclists (two traffic lights at each leg). However, this intersection has only 25 signal groups.

Standardized numbering

In the Netherlands a standardized numbering of these signal groups is used, see also Figure 1.6. Signal groups 1-12 are reserved for traffic lights that accommodate vehicular traffic. The numbers 1-3 are reserved for the eastern leg, the numbers 4-6 for the southern leg, the numbers 7-9 for the western leg, and the numbers 10-12 for the northern leg. The numbers 1, 2, 3 (as well as the numbers 4, 5, 6, the numbers 7, 8, 9, and the numbers 10, 11, 12) are reserved for the right-turn movement, through movement, and the left-turn movement respectively.

There are some exceptions. When a right-turn (left-turn) movement shares a lane with a through movement, all traffic lights accommodating this right-turn (left-turn) movement are partitioned in the signal group intended for the through movement; this is for example the case for the eastern leg and the southern leg in Figure 1.6. Also when a right-turn movement and a left-turn movement share a lane, all traffic lights that accommodate these turning movements are partitioned in the signal group intended for the through movement.

Similarly, signal groups 21-28 accommodate cyclists, signal groups 31-38 accommodate pedestrians, and 41-52 accommodate public transportation. The numbers 61 – 101 are reserved for signal groups at nearby intersections that are controlled with the same controller.

1.4.4 Conflicts

We define the *conflict area* of the intersection as the area that is shared by different traffic streams. The access to the conflict area is controlled with traffic lights. Some movements at the intersection are conflicting; these movements cannot safely cross the intersection simultaneously. Two traffic lights whose traffic streams either cross or merge are not necessarily conflicting; the corresponding movements may both be permitted at the same time.

A pair of signal groups has a *conflict* whenever they control the access to the intersection of at least one pair of conflicting movements. Consider a pair $\{i, j\}$ of conflicting signal groups. For each such pair of conflicting signal groups, *minimum clearance times* must be satisfied; a minimum clearance time from signal group i to signal group j is defined as the minimum amount of time that signal group i must be a red for, before signal group j may receive a green indication. Such minimum clearance times ensure that each traffic stream can safely cross the intersection without encountering traffic from a conflicting signal group.

1.5 Traffic light control

The task of a traffic light controller is to choose the indications (green, yellow or red) of the signal groups so that the minimum clearance times are satisfied (amongst other constraints) and such that road users experience as little hinder as possible. There are three types of traffic light control: pre-timed control, actuated control, and semiactuated control. A pre-timed controller requires no detectors, an actuated controller uses detector information to control all of the signal groups, and a semiactuated controller is a compromise between an actuated controller and a pre-timed controller; a semiactuated controller uses detector information to control some (but not all) of the signal groups. We elaborate on all three types. To this end, we use the intersection in Figure 1.8.

1.5.1 Pre-timed control

The most basic type of traffic light control is *pre-timed control*. This type does not require any detectors at the intersection; all detector information is ignored and all green, yellow, and red intervals are timed periodically. In other words, the signal groups operate according to a predetermined "fixed" period. In Figure 1.9 we have visualized such a predetermined "fixed" period for the intersection in Figure 1.8; such a diagram is called a *signal group diagram*.

A signal group diagram is usually found via mathematical optimization, seeking for the signal group diagram that minimizes (or maximizes) some objective function. For example seeking for the signal group diagram for which road users experience as little hinder as possible. We distinguish between the optimization for an *isolated intersection*, and the optimization for a *network of interconnected intersections*.

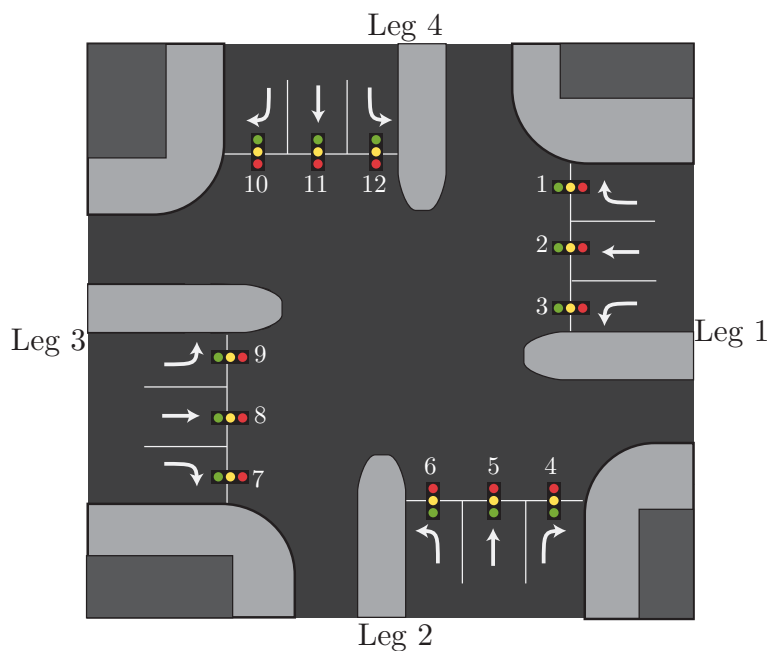


Figure 1.8: An intersection with 12 signal groups each consisting of a single traffic light.

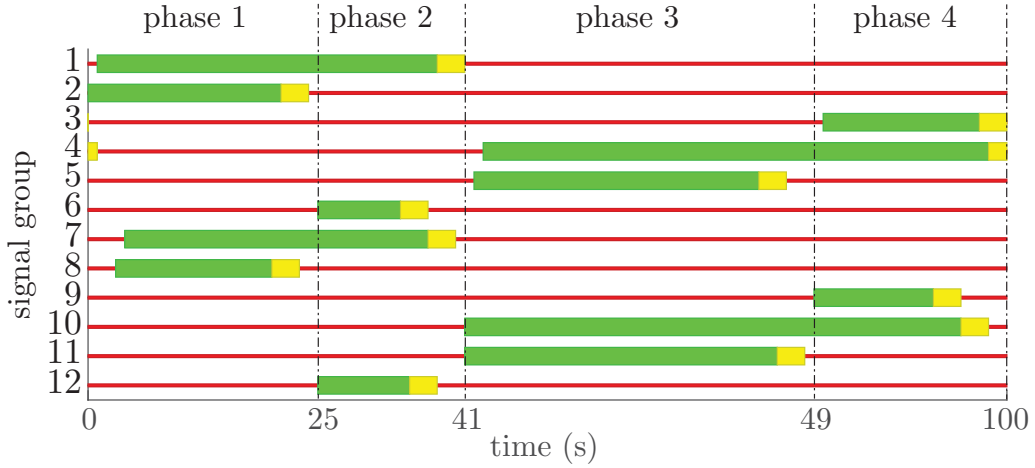


Figure 1.9: A signal group diagram for the intersection in Figure 1.8. This signal group diagram visualizes each green, yellow, and red interval during one period that repeats continuously. In this figure we have also visualized the phases associated with the single-ringed phase diagram in Figure 1.10, which we introduce in the next section.

Isolated intersections

An *isolated intersection* experiences negligible influence from other signalized intersections; the traffic arrives randomly over time (for example according to a Poisson process) and vehicles do not seem to travel together in platoons. For such intersections it is convenient and also justified to consider them in isolation.

Currently, two general approaches exist that optimize pre-timed control at isolated intersections: a phase-based approach and a group-based approach. A signal group diagram can be split up into phases; a phase is a set of signal groups that receives green simultaneously. A phase-based approach, e.g., (Allsop, 1971a,b, 1972, 1981; Gallivan and Heydecker, 1988; Han, 1990, 1996; Roupail and Radwan, 1990; van Zwieten, 2014; Webster, 1958), uses these phases as elementary building blocks. Such an approach first generates all (or a subset of all) sequences of phases. Subsequently, for each of these sequences the duration of the phases is optimized. On the one hand, if a single sequence or a small subset of sequences is generated, e.g., with (Hosseini and Orooji, 2009; Tully, 1966), the optimality of the resulting signal group diagram cannot be guaranteed. On the other hand, enumerating over all sequences is computationally expensive and might even be intractable.

In contrast to the phase-based approach, the group-based approach optimizes the structure of the signal group diagram and the duration of the green intervals simultaneously, e.g., (Cantarella and Improta, 1988; Improta and Cantarella, 1984; Sacco, 2014; Silcock, 1997; Wong and Heydecker, 2011; Wong and Wong, 2003; Wong, 1996; Yan et al., 2014). Some of the aforementioned papers, e.g., (Wong and Heydecker, 2011; Wong and Wong, 2003; Wong, 1996; Yan et al., 2014), are actually called lane-based methods in literature; in addition to optimizing the structure of the signal group diagram and the

duration of the green intervals simultaneously, these methods also optimize the lane-use arrows at the intersection. However, their modelling approach is the same as that of group-based methods (besides the optimization of the lane-use arrows). We elaborate on the aforementioned group-based approaches in Chapter 3 of this thesis.

In the Netherlands the tools VRIGen (Salomons, 2008) and COCON Graphium (Stolz and Veroude, 2013) are often used in practice to compute pre-timed controllers for isolated intersections, see also (Wilson, 2014). Both these tools use a phase-based approach and, as a consequence, both are computationally expensive. Moreover, VRIGen is designed to find the signal group diagram with the smallest period duration (and is not able to optimize any other objective function). In addition, as stated in (Stolz and Veroude, 2013), VRIGen is not always able to find the optimal signal group diagram as it considers only a subset of phase sequences.

Network of interconnected intersections

Signal group diagrams can also be optimized for a network of intersections. In such a network, the intersections are interconnected and, as a result, traffic often travels in platoons. To enhance the performance of the network, e.g., minimizing the hinder that road users experience, it may be desirable to create green waves between these intersections. For a green wave the traffic lights over several intersection are coordinated for one main direction, allowing continuous traffic flow (without stopping) over these intersections for this main direction; this allows a platoon of vehicles to flow through the network with as little hinder as possible.

Green waves can be created by first considering the intersections in isolation and creating a signal group diagram for each intersection separately. Thereupon, the timing of these different signal group diagrams with respect to each other can be optimized by solving a coordination problem (with for example (Gartner et al., 1975; Wünsch, 2008)), which creates the so called green waves. The result is a pre-timed controller for a network of intersections; not only the timings of the traffic lights at each of the intersections, but also the relative timings between the signal group diagrams of these intersections is important. Such a pre-timed controller for a network of intersections may form the basis for a semiactuated network controller; we elaborate on semiactuated controllers in Section 1.5.3.

One can argue that it might also be better to control a network of intersections with a pre-timed control because of its predictability. Today, traffic lights cause unpredictable delays when navigating through a network of signalized intersections. When using pre-timed control, the signal group diagram of each intersection can be communicated to the driver. This information can be used to visualize the future state of the upcoming traffic light to the road user, who can use this info to adjust its travel speed, reduce its waiting time at traffic lights, and save fuel. Furthermore, this information on the future state of the traffic lights can be used to obtain better estimates for travel times; these travel

time estimates include the estimated waiting time at the traffic lights. Car navigation systems can use these travel time estimates to calculate a smart route through a network of signalized intersections controlled by a pre-timed controller. Alternatively, predictions for the state of each traffic light can also be obtained from floating car data (Krijger, 2013). As (Krijger, 2013) shows, these predictions are particularly accurate for pre-timed control.

Fluctuations in traffic demand

A pre-timed controller operates according to a predetermined "fixed" period and does not adjust to changes in the traffic situation. However, the traffic demand at the intersection may be subject to large fluctuations. The amount of traffic arriving at the intersection probably differs between different segments of the day, e.g., morning rush hour, evening rush hours, midday, and night. This demand may also differ between days and is possibly especially large for 'special' situations like concerts and shopping nights. To account for these large fluctuations, a signal group diagram can be optimized for each of these situations separately.

Capacity of the intersection

When a traffic light controller has to be designed, usually demands are forecasted for the upcoming years. Typically the current demands are either estimated or counted, and some growth factor is applied to estimate the future demands. It is important to check whether the intersection can handle these forecasted demands or not. In this respect, the optimization of signal group diagrams can contribute; whenever through optimization we find a signal group diagram that can handle these forecasted amounts of traffic, other actions are not necessary. If no such signal group diagram can be found, other measures may be required to either reduce the forecasted demands at this intersection or increase the capacity of the intersection (for example by adding lanes to the intersection).

1.5.2 Actuated control

An *actuated controller* requires detectors to gather information on the current traffic situation. Vehicles are often detected with induction loops. Essential to the workings of such an induction loop is a powered copper wiring beneath the surface of the road. The electrical current that flows through this wire generates a magnetic field. The metal of a vehicle that passes over this detector causes a change in this magnetic field, which can be measured by a change in the current flowing through the copper wiring. To detect cyclists and pedestrians a simple button can be used. Also other types of sensors are used in practice; video detection, radar detection and infrared detection can all be used to detect vehicles, cyclists and pedestrians (Wilson, 2014). Since bikes are also made of metal, cyclists can also be detected with induction loops. Pedestrians, however, cannot.

An actuated controller uses the gathered information on the current traffic situation to control the different signal groups. Many different strategies exist for actuated control. Examples are reinforcement learning (El-Tantawy et al., 2013, 2014), model predictive control (Kamal et al., 2015; Sutarto, 2016), backpressure (Gregoire et al., 2015; Zaidi et al., 2016) self-control (Lämmer and Helbing, 2008; Płaczek, 2014), fuzzy logic (Chiou and Huang, 2013; Koukol et al., 2015), and genetic algorithms (Kaur and Agrawal, 2014).

Many of the aforementioned actuated controllers serve the signal groups in a pre-determined order, e.g., (Kamal et al., 2015; Kaur and Agrawal, 2014; Sutarto, 2016), as do many of the approaches that are popular in practice, e.g., SCOOT (Robertson and Bretherton, 1991), SCATS (McCann, 2014) and CCOL (Peters and Prinsen, 2001). This predetermined order can be visualized in a phase diagram and is usually obtained from a pre-timed controller, i.e., the actuated controller is based on a pre-timed controller; we elaborate on these phase diagrams later in this section.

Some other approaches choose, besides the duration of each green interval, also the order in which the signal groups receive their green indications, e.g., (Gregoire et al., 2015; Lämmer and Helbing, 2008; Płaczek, 2014; Zaidi et al., 2016). For example, in (Lämmer and Helbing, 2008) a priority index is associated with each signal group. This time-dependent priority index depends on the current and forecasted amount of traffic at each signal group (amongst other things). This priority index is used to decide which signal group will receive a green indication next; the larger the priority index of a signal group, the sooner it will receive a green indication. Another approach that chooses the order of the green intervals is the backpressure approach. This approach associates a pressure with each queue of vehicles; this pressure builds up as its queue grows, and it decreases as the downstream queues (that this traffic will join) grow. It chooses the next phase to be the one with the largest pressure release.

We believe that these truly adaptive controllers (also able to choose the order of the green intervals) are very important for the future of effective traffic management. However, we also believe that they are not the best choice in each and every situation finding their application especially in low and medium traffic conditions. In high traffic conditions, when the intersection cannot (or can just barely) handle the amount of traffic arriving at it, it becomes of extreme importance to use the intersection as effectively as possible; for some phase sequences (order of green intervals) the intersection can be used more efficiently than for others because for these sequences the time wasted on clearance times is smaller and, therefore, more time is available for green intervals. Due to the high computational complexity, it is impossible for an actuated controller to oversee all possible orders in which these signal groups can be served (in real time). Therefore, we expect an (appropriate) pre-timed controller (or an actuated controller based on this pre-timed controller) to outperform the truly adaptive controller in these high traffic situations. In fact, the truly adaptive controller can even fail completely in high traffic situations due to, for example, dynamic instabilities (Lämmer and Helbing, 2008); a famous example is using a clearing policy in a Kumar-Seidman network (Kumar and Seidman, 1990); for an

overview of these instabilities, we refer to (Lämmer and Helbing, 2008). Also the adaptive controller proposed in (Lämmer and Helbing, 2008) cannot adequately control different traffic streams in high traffic situations; they illustrate that the proposed adaptive controller fails at controlling a simple isolated intersection with four signal groups even when the demand is far from critical; under the same traffic conditions this isolated intersection can easily be handled by a pre-timed controller. To overcome this shortcoming, they propose to add a stabilization rule, which effectively applies a pre-timed controller when the queue lengths are large.

An additional advantage of keeping the order of the green intervals fixed is that road users know what to expect; if this order changes, road users might anticipate on the traffic light switching to a green indication resulting in an unsafe situation. When serving the signal groups in a pre-determined order, it is common to visualize this order in a *phase diagram*, see (National Research Council (U.S.), 2010; Wilson, 2014). Such a phase diagram visualizes a sequence of phases (a phase is a set of signal groups that receive green simultaneously), which repeats indefinitely. In Figure 1.10 we have depicted a potential phase diagram for the intersection in Figure 1.8. This phase diagram serves the through movement and the right-turn movement of legs 1 and 3 during phase 1. Subsequently, the through movements receive a red indication and the left-turn movements of legs 2 and 4 receive right-of-way in phase 2. Next, the through movement and the right-turn movement of legs 2 and 4 are served during phase 3, whereafter the through movements make way for the left-turn movements of legs 1 and 3 in phase 4. This process repeats. The actuated controller may also skip a phase whenever for example no traffic is waiting at the corresponding traffic lights.

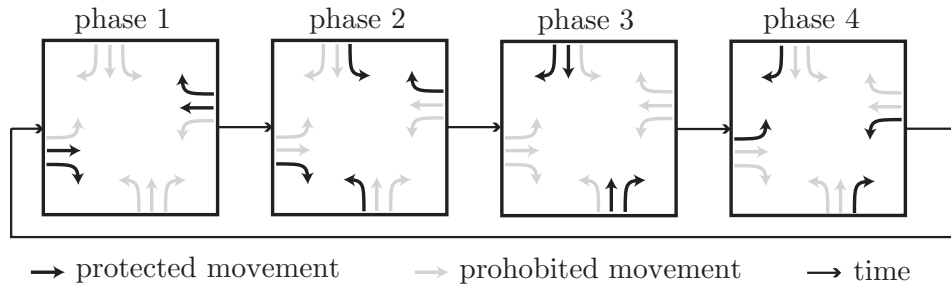


Figure 1.10: A potential phase diagram with one ring for the intersection in Figure 1.8.

The phase diagram in Figure 1.10 only has one *ring*. A ring is defined as a sequence of phases that repeats indefinitely. To provide more flexibility, a phase diagram is also allowed to have multiple rings that operate independently, see Figure 1.11; a movement is then allowed whenever it is allowed by the current phase of at least one of the rings. Such a multi-ring phase diagram uses *barriers*; a barrier may only be crossed by all the rings at the same time. Thus, for the double-ringed phase diagram of Figure 1.11 the transition from phase 1 to phase 2 of ring 1 must occur simultaneously with the transition from phase 5 to phase 6 of ring 2. These barriers provide some synchronisation between

the rings and are often needed to make sure that all conflicts are taken into account. For example, the through movement of leg 1, which is served in phase 1, conflicts with the through movement of leg 4, which is served in phase 7. Without the barriers, phase 7 (of ring 1) can occur simultaneously with phase 1 (of ring 2); this is undesired because the conflicting through movements of leg 1 and leg 4 can then receive a green indication simultaneously. Including the two barriers guarantees that phase 1 and phase 7 cannot co-occur and, therefore, as desired, these conflicting through movements cannot coincide.

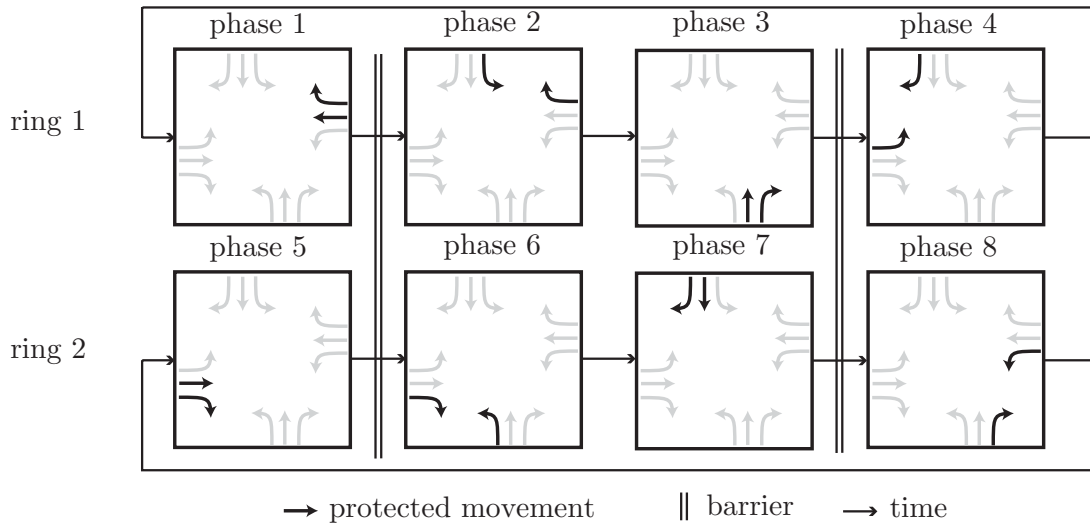


Figure 1.11: A potential phase diagram with two rings and two barriers for the intersection in Figure 1.8.

Using multiple rings may provide more flexibility. Compare for example the single-ringed phase diagram of Figure 1.10 with the double-ringed phase diagram of Figure 1.11. When we add two additional barriers to the double-ringed phase diagram (one after phases 2 and 6 and one after phases 4 and 8), ring 1 and ring 2 always jump to the next phase simultaneously. As a consequence, this double-ringed phase diagram then has the same phase transitions as the single-ringed phase diagram. This implies that all phase transitions of this single-ringed phase diagram are also allowed by the double-ringed phase diagram. However, this does not hold the other way around. For example consider the case that no traffic is waiting at signal group 6 (left-turn movement of leg 2, which is served in phase 6) and signal group 7 (right-turn movement of leg 3, which is served in phase 6) and that a lot of traffic is waiting at signal groups 10, 11 and 12 (the signal groups of leg 4, which are served in phase 2 and phase 7). With the double-ringed phase diagram, the actuated controller may decide to end the green interval of signal groups 6 and 7 (phase 6) prematurely, and serve signal groups 10, 11 and 12 simultaneously (phases 2 and 7 are active). This is not possible for the single-ringed phase diagram.

For the construction of actuated controllers often phase diagrams are desired. But how can we find a 'good' phase diagram? The common practice is to first find a pre-timed controller (National Research Council (U.S.), 2010; Wilson, 2014). For such a pre-timed controller all green, yellow, and red intervals are timed periodically, i.e., the signal groups

operate according to some predetermined "fixed" schedule. From this pre-timed controller we can then obtain: the phases that are used and the order in which they are used. This information can be used to construct a phase diagram. For example, the phase diagram in Figure 1.10 can be obtained from the pre-timed controller in Figure 1.9. In Appendix A.1 we show how to automatically obtain a single-ringed phase diagram from a pre-timed controller. As mentioned in the previous section, the traffic demand at the intersection may be subject to large fluctuations. The amount of traffic arriving at the intersection probably differs between different segments of the day, e.g., morning rush hour, evening rush hours, midday, and night, and may be especially large for 'special' situations like concerts and shopping nights. For each of these situations we can design a signal group diagram. When using a (semi)actuated controller, we can design an appropriate phase diagram (based on these signal group diagrams) for each of these different situations.

As we show in Appendix A.2, actuated controllers may behave as a pre-timed controllers under certain conditions. For example this may happen when the intersection under actuated control is nearly saturated or oversaturated, i.e., when the actuated controller cannot or can just barely handle the amount of traffic that arrives at it. Therefore, these situations call for the design of a pre-timed controller.

1.5.3 Semiactuated controller

Besides a fully actuated controller, which uses detector information to choose the timings of all signal groups, and a pre-timed controller, which uses no detector information, it is also possible to have a compromise between the two: the *semiactuated controller*. A semiactuated controller uses detector information only to choose the timings of *some* signal groups. Such a controller may be convenient in some situations. For example when creating green waves. When green waves are desired, a pre-timed controller can be designed for a network of intersections, see also Section 1.5.1. Such a pre-timed controller that optimizes the performance of a network of intersections may form the basis of a semiactuated controller. From this signal group diagram we can extract the phases that are used and the order in which they are used; this can form the basis for a phase diagram. Furthermore, this signal group diagram gives a predetermined "fixed" period for each signal group that participates in the creation of green waves; the signal groups that participate in the creation of the green waves can run according to this predetermined "fixed" period, which guarantees the creation of these green waves.

1.6 Focus of this thesis and its motivation

This thesis focusses on the optimization of *pre-timed control at isolated conventional intersections*. Below we briefly summarize the motivation given in the previous sections:

- For some intersections it may be justified and convenient to consider them in isolation as they are affected very little by neighboring intersections.
- Pre-timed control may form the basis for the design of actuated controllers and semiactuated controllers. We can construct a phase diagram from such pre-timed controllers, which may be an integral part of the (semi)actuated controller.
- In heavy traffic we expect an appropriate pretimed controller (or a (semi)actuated controller that is based on this pretimed controller) to outperform a truly adaptive controller that also chooses the order of the green intervals.
- A (semi)actuated controller may behave as a pre-timed controller in some situations. Such situations call for the design of a pre-timed controller.
- Pre-timed control at isolated intersections may form the basis for pre-timed control of a network of intersections; by synchronizing the pre-timed controllers of the individual intersections, green waves can be created and the overall network performance can be improved. The resulting pre-timed controller may form the basis of a semiactuated controller.
- The predictability of pre-timed controllers can possibly be used by smart cars. For example to visualize the future states of the traffic lights to road users and to calculate a smart route through a network of intersections that takes into account the future states of the traffic lights.
- By optimizing a pre-timed controller (for an isolated intersection) we can check whether the intersection can handle the amount of traffic that is forecasted to arrive at it, i.e., we can check whether the intersection has enough capacity.

1.7 Goal and contributions

The goal of this thesis is to develop approaches to mitigate congestion and reduce the hinder that road users experience at traffic lights. These approaches may aid traffic engineers in their process of designing a (pre-timed) traffic light controller (as well as their process of designing intersection layout) by returning better solutions and returning these solutions faster. More specifically, this thesis has the following contributions:

- An efficient optimization formulation is proposed, which allows us to obtain signal group diagrams much faster. During the process of designing a (pre-timed) traffic light controller, usually mathematical optimization is combined with expert knowledge of traffic engineers. Often multiple signal group diagrams are computed during such a process. Hence, the faster optimization can be used to improve work-flow for these traffic engineers.

- The proposed formulation also allows the optimization of the number of green intervals of each signal group. This allows us to optimize over a larger set of signal group diagrams and, as a consequence, find better signal group diagrams.
- The proposed formulation is extended to optimize integral signal group diagrams; such integral schedules are desired in practice as they are clear, presentable and easy to work with.
- The proposed formulation is extended to simultaneously optimize a signal group diagram and the layout of the intersection (arrival lanes, departure lanes, lane-use arrows et cetera), which can aid in the design of intersections. It allows us for example to find the smallest intersection that has sufficient capacity.

1.8 Thesis overview

First, in Chapter 2, we consider the mathematical modeling of traffic lights and the additional travel time (delay) that these traffic lights induce on road users.

Thereupon, in Chapter 3 we formulate an optimization problem that can find the optimal signal group diagram for which each signal group receives a single green interval. We consider three objective functions: minimizing the period duration of the signal group diagram, minimizing the average delay that road users experience, and maximizing the capacity of the intersection; when maximizing the capacity of the intersection, we seek for the signal group diagram that can handle the largest increase in the amount of traffic arriving at the intersection. When minimizing the period duration or maximizing the capacity, the optimization problem can be formulated as a mixed-integer linear programming (MILP) problem. When minimizing the average delay that road users experience at the intersection, we approximate the convex objective function with piecewise linear functions; the resulting optimization problem is then also a MILP problem. In Chapter 3 we also compare the proposed optimization problem with currently existing formulations via an extensive case study. The proposed formulation seems to be superior.

Subsequently in Chapter 4 we extend the formulated optimization problem so that it can also optimize the number of green intervals that each traffic light receives. We show that allowing traffic lights to receive multiple green intervals might significantly reduce the delay that road users experience at the intersection.

Thereupon, in Chapter 5 we elaborate on the problem of optimizing integral signal group diagrams. For such integral signal group diagrams all switches to green, yellow, and red are scheduled at an integral second. These integral signal group diagrams are desired in practice as they are clear, more presentable, and easier to work with.

In Chapter 6 we extend the optimization problem to also optimize the layout of the intersection. When designing an intersection it is important to choose these departure lanes, arrival lanes, and lane-use arrows, e.g., should a left-turn movement be accommo-

dated by one or two lanes? With the proposed extended optimization problem, we are able to simultaneously optimize the arrival lanes, departure lanes, the lane-use arrows, and a corresponding signal group diagram. We can then, for example, use this optimization framework to assist in the design of intersections as follows. We can first check whether the currently existing intersection has enough capacity. To this end we fix the number of arrival lanes, the number of departure lanes at each leg, and the lane-use arrows marked on each of the arrival lanes. Subsequently, we search for the signal group diagram that can handle the largest increase in the amount of traffic arriving at the intersection. If the intersection has enough capacity, then no other measures are necessary. However, if the capacity of this intersection is not sufficient, changing the lane-use arrows may be a financially attractive solution to increase the capacity. We can check if we can realize a sufficient increase in capacity by changing the lane-use arrows while maintaining the arrival lanes and the departure lanes. If the potential increase in capacity is still not sufficient, we can assess the effect of adding lanes to the intersection. We can, for example, try to find the minimum number of (arrival and departure) lanes for which the capacity of the intersection is sufficient.

In Chapter 7 we elaborate on how the proposed optimization framework can be used in practice. In that chapter we also elaborate on some issues that might be relevant in practice, but are not yet touched upon in this thesis. For example, we elaborate on some additional constraints that might be relevant in practice and we elaborate on how to find several signal group diagrams (instead of only the optimal one).

Finally, in Chapter 8 we give our concluding remarks and recommendations.

Modeling traffic lights and delay

The focus of this thesis is the optimization of signal group diagrams for isolated intersections. Such a signal group diagram can be used to construct a phase diagram, which may form the basis of an actuated controller. An important objective of an actuated controller is to minimize the hinder that road users experience. Therefore, we would like to find the signal group diagram that minimizes the additional travel time (delay) that road users experience due to traffic light control. Several formulae exist that relate (periodic) traffic light timings to the average delay that road users experience, e.g., the ones from (Miller, 1963; van den Broek et al., 2006; Webster, 1958). These formulae approximate the delay that road users experience at a first-in-first-out (FIFO) queue. In this chapter we elaborate on these approximate formulae and show how to model the traffic waiting at the intersection by using FIFO queues. In the next chapter we use these approximate formulae to formulate the objective function when searching for the signal group diagram that minimizes the average delay that road users experience at the intersection.

First, we elaborate on the assumptions on which the approximate formulae are based. Consider an arrival lane for which the access to the intersection is controlled with a traffic light. Such an arrival lane is intended for the use of a single line of vehicles; we refer to this arrival lane as queue q . The approximate formulae assume the following on queue q .

Assumption 2.1 (Single type of traffic). *The traffic arriving at queue q is composed of only one type of traffic, e.g., only passenger cars.*

In sections 2.3–2.5 we show how to use these approximate formulae also for mixed traffic flows (a mixed traffic flow consists of multiple types of vehicles), cyclist flows, and pedestrian flows.

In reality a traffic light can visualize three indications: green, yellow, and red. However, mathematically it suffices to have only two modes: effective green and effective red. We elaborate on these effective green and effective red modes in Section 2.1.

Assumption 2.2 (Pre-timed control). *The access to the intersection of queue q is controlled with a pre-timed controller. This pre-timed controller alternates between an effective green mode of g seconds and an effective red mode of r seconds.*

Thus, it is assumed that the traffic light receives a single green interval during a repeating period; in Chapter 4 we elaborate on the case that traffic lights receive multiple green intervals during a repeating period.

Assumption 2.3 (Independent and identical arrivals). *Let A_k be the time between the k th and the $k + 1$ st arrival at queue q . The times between two subsequent arrivals are independent and identically distributed (i.i.d.). In other words, the random variables A_0, A_1, A_2, \dots are i.i.d.*

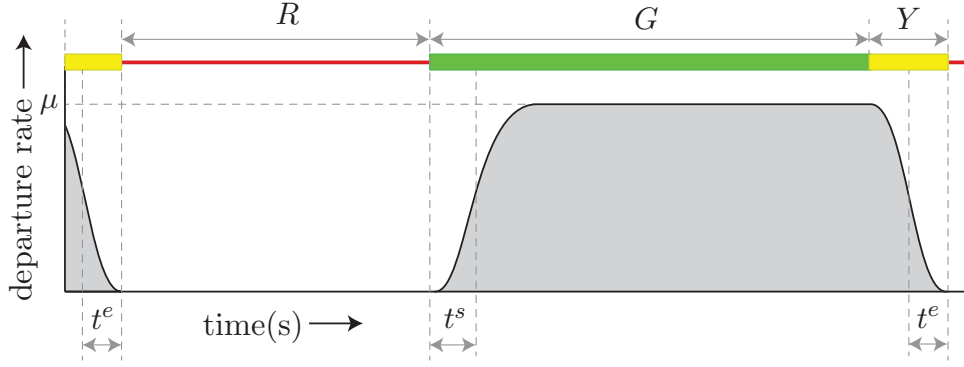
This implies that A_k does not depend on the historic interarrival times A_0, A_1, \dots, A_{k-1} and has exactly the same distribution as each of the stochastic variables A_0, A_1, A_2, \dots . To be more specific, in (Webster, 1958) the arrival process is assumed to be Poisson (so exponentially distributed interarrival times), whereas (Miller, 1963; van den Broek et al., 2006) assume a more general arrival process.

Assumption 2.4 (Deterministic departures). *Vehicles depart one by one in equal time intervals during an effective green interval of queue q .*

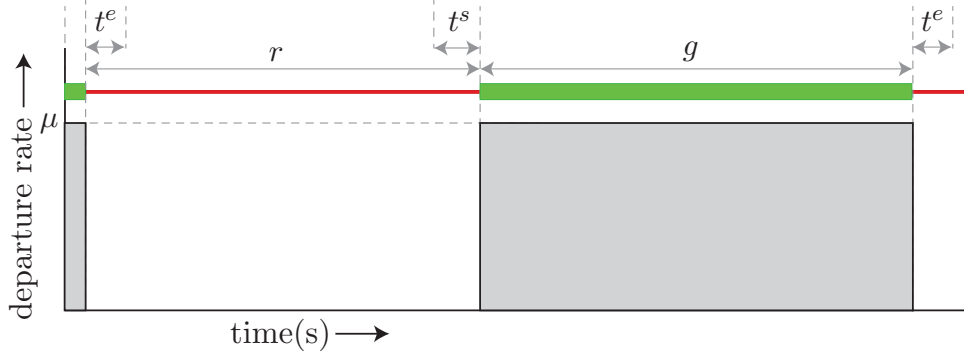
The approximate formulae relate the signal timings g and r (and characteristics of the arrival and departure process) to the average delay d that road users experience at this traffic light. In the next section we elaborate on the effective green and effective red modes. Subsequently, we give the approximate formulae; these formulae assume that the traffic arriving at the queue is composed of a single type of traffic, e.g., passenger cars. Thereupon, we show how to also use the aforementioned approximate formulae for mixed traffic flows, cyclist flows, and pedestrian flows.

2.1 Effective green and effective red modes

A traffic light has three indications: green, yellow, and red. Note that traffic lights intended for pedestrians and cyclists often have different indications: green, flashing green, and red. However, in this thesis we assume w.l.o.g. that all traffic lights have the indications green, yellow, and red. Instead of these three indications, mathematically it suffices to have only two modes: effective green and effective red. We elaborate on the differences between an indication and a mode. To this end, consider a lane for which the access to the intersection is controlled with a pre-timed controller. This pre-timed controller alternates between a green indication of G seconds, a yellow indication of Y seconds, and a red indication of R seconds. Define the departure rate to be the rate at which traffic enters the conflict area of the intersection, i.e., the rate at which traffic crosses the stop line. Figure 2.1a visualizes the average departure rate during a green, yellow and red indication assuming that the queue is never emptied. At the start of the green indication, the departure rate increases towards the *saturation flow rate* μ . Since, the departure rate does not immediately equal this saturation flow rate, capacity is lost at the start of the



(a) Average departure rate during a green (G), yellow (Y) and red (R) indication assuming that the queue is never emptied. The average departure rate does not equal the saturation flow rate (μ) at the start of the green interval and at the end of the yellow interval; this induces a starting lost time (t^s) and an ending lost time (t^e).



(b) Average departure rate during an effective green (g) and an effective red (r) mode assuming that the queue is never emptied. Traffic is able to depart at the saturation flow rate μ during the entire effective green interval.

Figure 2.1: Average departure rate during a green, yellow and red indication versus the average departure rate during an effective green and an effective red mode both under the assumption that the queue is never emptied.

green interval; this capacity loss is caused by the reaction time of road users and the (finite) acceleration of traffic. Moreover, capacity is lost at the end of a yellow indication, which is caused by the traffic stopping before the end of a yellow indication.

Instead of these three indications, mathematically it suffices to have only two modes: effective green and effective red. During an effective green interval, traffic can depart at the saturation flow rate μ and during an effective red interval no traffic departs. In Figure 2.1b we have visualized the effective green time g and the effective red time r . The traffic light is in the effective red mode during the first t^s seconds of the green indication; this models the capacity loss at the start of a green indication. Similarly, we model the capacity loss at the end of the yellow interval; the traffic light is in the effective red mode during the last t^e seconds of the yellow indication. We call t^s and t^e the starting lost time and the ending lost time respectively. We consider these lost times to be given and fixed values; the starting lost time and the ending lost time may differ between signal groups. We can then easily transform a signal group diagram with green, yellow, and red indications into a signal group diagram with effective green and effective red modes (and

vice versa).

2.2 Approximating the delay for a traffic flow of only passenger cars

We elaborate on the formulae that approximate the delay that road users experience at a lane under pre-timed control. For each signal group diagram we define the *number of realizations* of a signal group as the number of distinct (periodically repeating) green intervals that this signal group receives during one period. For example, for the signal group diagram in Figure 1.9 each signal group has one realization. All aforementioned formulae (Miller, 1963; van den Broek et al., 2006; Webster, 1958) assume that the number of realizations equals one for each of the signal groups. For these formulae, the delay d can be written as the sum of a deterministic delay term d_{det} and a stochastic delay term d_{stoch} , i.e., $d = d_{\text{det}} + d_{\text{stoch}}$. The deterministic delay term d_{det} is the same for all aforementioned approximate formulae, however, the stochastic delay term d_{stoch} is different. The first term (d_{det}) is based on the assumption of deterministic arrivals, and the stochastic delay term d_{stoch} accounts for the stochastic effects on the delay d that road users experience. We elaborate on the deterministic delay term d_{det} and the stochastic delay term d_{stoch} .

2.2.1 Deterministic delay term

Assume that the arrival process and the departure process are both deterministic and fluid-like, see Figure 2.2. For this fluid-like queueing process the number of passenger cars waiting in front of the traffic light does not have to be integral, i.e., the number of passenger cars in the queue may attain any non-negative real value. Let λ be the demand (in passenger cars per second) and μ be the saturation flow rate (in passenger cars per second); we may also call λ the arrival rate. Define the queue length to be the number of passenger cars that is waiting to cross the intersection. For this deterministic queueing process, the queue length increases linearly with a rate of λ passenger cars per second during an effective red interval. Subsequently, the queue length decreases linearly with a rate of $\mu - \lambda$ passenger cars per second during the effective green interval; traffic then departs at the saturation rate μ , however, still traffic arrives with rate λ . Whenever the queue is emptied during an effective green interval, this queue remains empty until the effective red period starts; all traffic arriving to an empty queue during an effective green interval can cross the intersection without experiencing any delay. Let g be the effective green time of the queue and let r be the effective red time of the queue. Assume that the queue is undersaturated, i.e., the (average) amount of traffic arriving during one period of $T := g + r$ seconds can depart during an effective green interval, which has a duration of g seconds. In other words, we assume:

$$\lambda(g + r) < \mu g \quad (2.1)$$

The queue is then empty at the end of an effective green interval for the deterministic and fluid-like queueing process. The deterministic delay term d_{det} , which is the delay that passenger cars experience for this deterministic queueing system, can then be calculated from the average queue length \bar{x} by using Little's law (Chhajed and Lowe, 2008):

$$d_{\text{det}} := \bar{x} / \lambda,$$

which gives

$$d_{\text{det}} := \frac{r^2}{2T(1 - \rho)}, \quad (2.2)$$

where $\rho = \lambda/\mu$ is called the load of the lane (and $\rho < 1$ by (2.1)).

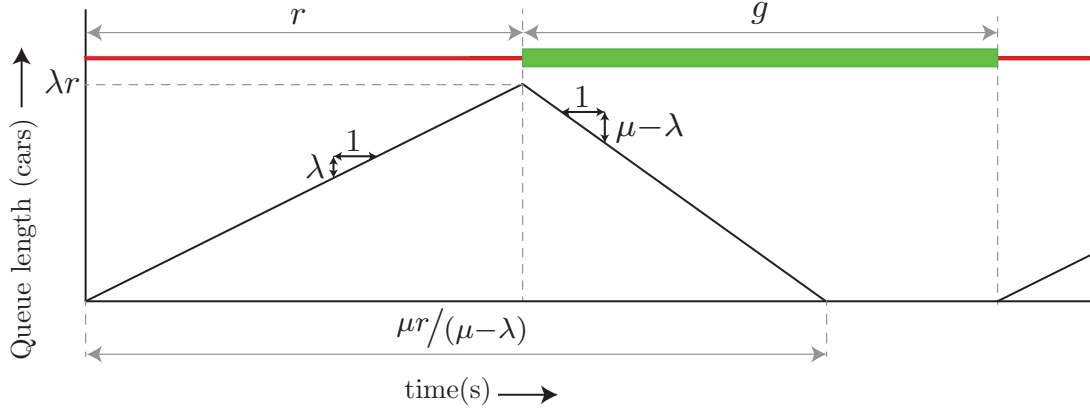


Figure 2.2: The queueing of passenger cars for the deterministic and fluid-like queueing system associated with the deterministic delay term d_{det} .

Remark 2.1. When obtaining the evolution of the queue length of Figure 2.2 we implicitly assume that all passenger cars in the queue do not occupy any space, i.e., the back of the queue is positioned at the stop line and passenger cars are added to the queue whenever they reach this back of the queue. In reality, this is not the case: passenger cars do occupy space. When the amount of space that a queue occupies depends linearly on the amount of traffic in the queue, then the back of the queue would propagate backwards during an effective red interval and, as a result, the arrival rate at the back of the queue does not equal λ (the arrival rate is larger due to the backward propagation of the queue). The back of the queue would propagate forward during an effective green interval and, as a result, the amount of traffic in the queue does not decrease with $\mu - \lambda$ (but with a larger rate). In Appendix B.1 we elaborate on this effect and show that we may assume w.l.o.g. that the queue does not occupy any space (the back of the queue is then positioned at the stop line).

2.2.2 Stochastic term

To obtain the deterministic delay term d_{det} , the demand and the departure process were assumed to be deterministic and fluid-like. Assuming that the queue is undersaturated, the queue is always emptied during each effective green interval for this deterministic queueing process. However, when the queueing process is not deterministic and fluid-like, e.g., the time between two subsequent arrivals follows some stochastic distribution, this does no longer hold; the queue might not always be empty at the end of an effective green interval. The passenger cars that are still in the queue at the end of an effective green interval have to wait at least an entire effective red interval before they can cross the intersection. The stochastic delay term d_{stoch} accounts for such stochastic effects. This stochastic term is based on the assumption that the queue is undersaturated, i.e., the inequality (2.1) is satisfied.

The stochasticity in the arrival process is expressed in a measure σ , which we define as follows. According to Assumption 2.4, vehicles depart one by one in equal time intervals, which we for the time being call *slots*. The stochasticity measure σ is defined to be the standard deviation of the number of cars that arrive during such a slot. This measure σ is used by the stochastic delay terms of the formulae in (Miller, 1963; van den Broek et al., 2006; Webster, 1958). The stochastic delay term $d_{\text{stoch}}^{\text{miller}}$ of the formula in (Miller, 1963) distinguishes between two situations:

$$d_{\text{stoch}}^{\text{miller}} = \begin{cases} \frac{r}{2\lambda(1-\rho)T} \left(\frac{\sigma^2}{1-\rho} + \frac{(2\rho-1)T+r}{((1-\rho)T-r)} \frac{\sigma^2}{\rho} \right) & \text{if } (2\rho-1)T+r \geq 0, \\ \frac{r}{2\lambda(1-\rho)T} \frac{\sigma^2}{1-\rho} & \text{otherwise.} \end{cases}$$

Note that from (2.1) it follows that $(1-\rho)T-r > 0$. The stochastic delay term $d_{\text{stoch}}^{\text{webster}}$ of the formula in (Webster, 1958) does not depend on σ because it assumes Poisson arrivals for which $\sigma^2 = \rho$,

$$d_{\text{stoch}}^{\text{webster}} = \frac{\rho T^2}{2\mu(T-r)((1-\rho)T-r)}.$$

The last stochastic delay term $d_{\text{stoch}}^{\text{vdbroek}}$ is the one from (van den Broek et al., 2006):

$$d_{\text{stoch}}^{\text{vdbroek}} = \frac{r}{2\lambda(1-\rho)T} \left(\frac{\sigma^2}{1-\rho} + \frac{r\rho^2\sigma^2T^2}{(1-\rho)(T-r)^2((1-\rho)T-r)} \right), \quad (2.3)$$

In (van den Broek et al., 2006) her own approximation is compared to the approximate formulae from (Miller, 1963) and (Webster, 1958); the approximate formula from (van den Broek et al., 2006) seems to be a very good approximation. In this thesis we only use the stochastic delay expression $d_{\text{stoch}}^{\text{vdbroek}}$, however, it is also possible to use the other two stochastic expressions.







Vehicle	Description	PCE value
	passenger car	1
	truck	1.5
	articulated truck	2.3
	bus	2
	motor	0.4
	bicycle	0.2

Table 2.1: Passenger car equivalent (PCE) values for different types of vehicles. These values are obtained from (Wilson, 2014)

2.3 Approximating the delay for a mixed traffic flow

In reality a traffic flow is not comprised of only passenger cars, but these cars often share the road with different types of vehicles, e.g., trucks, busses, and motors. In practice, it is common to convert such a mixed traffic flow by using passenger car equivalent values, see for example (National Research Council (U.S.), 2010; Wilson, 2014). A mixed traffic flow is then converted to some equivalent number of passenger cars, and, subsequently, the analysis is performed as if the traffic flow consists of only passenger cars.

These passenger car equivalent values are defined as follows, e.g., see (Wilson, 2014). Let t_i be the average time that is needed for the departure of a single vehicle of type i , and assume that type 1 vehicles are passenger cars. The effective green time t needed for the departure of n_1 type 1 vehicles, n_2 vehicles of type 2, n_3 vehicles of type 3 et cetera equals:

$$t := n_1 t_1 + n_2 t_2 + n_3 t_3 + \dots,$$

which can be rewritten to:

$$t := t_1 \left(n_1 + \frac{t_2}{t_1} n_2 + \frac{t_3}{t_1} n_3 + \dots \right).$$

We call t_i/t_1 the passenger car equivalent (PCE) value associated with type i vehicles. In Table 2.1 we depict the PCE values as given in (Wilson, 2014). In this table, a fixed value is given for each vehicle type. However, these PCE values may also depend on the characteristics of the lane for which we want to convert the mixed traffic flow; for example, the time t_i may also depend on the slope of the lane.

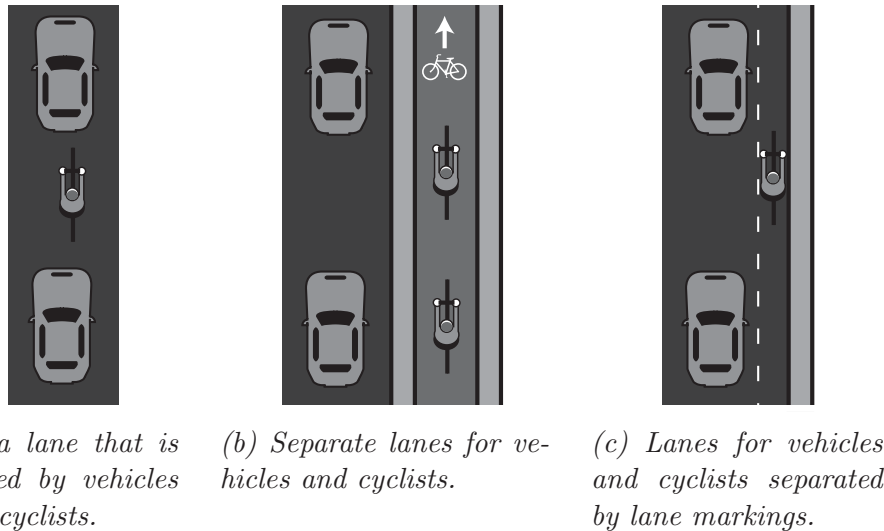


Figure 2.3: Different types of lanes intended for the use of cyclists.

2.4 Approximating the delay for a cyclist flow

In this section we distinguish between three different situations for cyclist flows, see also Figure 2.3. For the first situation cyclists share a lane with other types of vehicles, see Figure 2.3a; the mixed traffic flow forms one queue when awaiting a green indication. In this case the mixed traffic flow can be converted by using passenger car equivalent values as shown in the previous section. For the second situation, cyclists have their own dedicated lane segment, see Figure 2.3b; cyclists then form their own queue. In this case we can approximate the delay that cyclists experience by using the same approximations as we used previously, which are the ones from (Miller, 1963; van den Broek et al., 2006; Webster, 1958). We then use cyclists instead of passenger cars as our main unit. For the third situation, vehicular traffic and cyclists each have their own dedicated lanes, however, these lanes are only separated by some markings on the road, see Figure 2.3c. As a result, the vehicles and the cyclists still affect each other, e.g., a cyclist may (unintentionally) obstruct a vehicle from passing. When this effect is large, it may be preferred to model the vehicles and cyclists as being one flow of traffic and converting this stream using passenger car equivalent values; the vehicle flow and the cyclist flow are then together modelled as one queue. When this effect is small, it may be preferred to consider these traffic streams separately; the passenger flow and the cyclist flow are each modelled with their own queue.

2.5 Approximating the delay for a pedestrian flow

Several differences between a pedestrian flow and a traffic stream of passenger cars exist, see also Figure 2.4. First, cars depart one by one whereas pedestrians often cross the intersection in parallel. Second, a pedestrian crossing can often be traversed from two

directions. These two opposite pedestrian flows share the capacity of the pedestrian crossing. In this thesis, also for these pedestrian crossings we approximate the delay by using the aforementioned approximations, which are the ones from (Miller, 1963; van den Broek et al., 2006; Webster, 1958). We then use pedestrians instead of cars as our main unit. Furthermore, the saturation flow rate μ is then the (average) total number of pedestrians that can cross this pedestrian crossing per time unit (from both directions), and the demand λ is then the (average) total number of pedestrians arriving (from both directions) at this pedestrian crossing per time unit. Note that by using these approximate formulae, we implicitly assume that pedestrians cross the pedestrian crossing one by one. However, since the saturation flow rate μ of a pedestrian flow is usually very large this often has only little effect on the approximated delay.

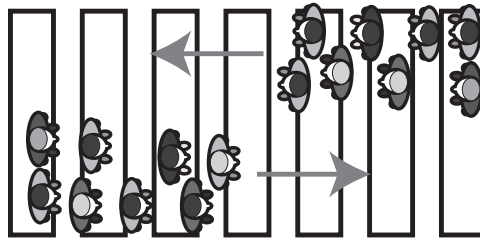


Figure 2.4: Parallel crossing of pedestrians at a two-sided pedestrian crossing. The pedestrian flows from both sides of the crossing share the capacity of the pedestrian crossing.

Remark 2.2. Recall that by applying the approximate formulae from (Miller, 1963; van den Broek et al., 2006; Webster, 1958) we assume that the interarrival times are independent and identically distributed (i.i.d.). Often pedestrians (cyclists) have to traverse multiple pedestrian (cyclist) crossings to cross a leg, see for example Figure 1.6. As a consequence, in practice these interarrival times are often not i.i.d.; the arrivals at one pedestrian crossing are (partially) influenced by the departures at another pedestrian crossing.

2.6 Summary

In this section we have shown how to model traffic lights by using two modes: an effective green mode and an effective red mode. Furthermore, we have shown how to model the traffic waiting at the intersection with (first-in-first-out) queues. For each of these queues we can approximate the average delay that road users experience with one of the many known approximations.

In the next chapter we use the introduced effective green and effective red modes, first-in-first-out queues and delay approximations to formulate an optimization framework to find the best signal group diagram. With that optimization framework we can, for example, find the signal group diagram with the smallest period duration, assess the

capacity of the intersection, or minimize the average delay that road users experience at the intersection.

Optimizing pre-timed control: single realization

3.1 Introduction

Two general approaches exist that optimize pre-timed control at isolated intersections: a phase-based approach and a group-based approach. A signal group diagram, defining when signal groups are green, yellow and red during a repeating period, can be split up into phases. During such a phase, the indicator lights of the signal groups do not change. A phase-based approach uses these phases as elementary building blocks. Examples of such phase-based approaches are the ones from (Allsop, 1971a,b, 1972, 1981; Gallivan and Heydecker, 1988; Han, 1990, 1996; Roupail and Radwan, 1990; van Zwieten, 2014; Webster, 1958). Such an approach, first generates all (or a subset of all) sequences of phases. Subsequently, for each of these sequences the duration of the phases is optimized. On the one hand, if a single sequence or a small subset of sequences is generated, e.g., with (Hosseini and Orooji, 2009; Tully, 1966), then the optimality of the resulting signal group diagram cannot be guaranteed. On the other hand, enumerating over all sequences is computationally expensive and might even be intractable.

In contrast to the phase-based approach, the group-based approach optimizes the sequence of the phases and the duration of the green intervals simultaneously. Examples of these group-based approaches are the ones from (Cantarella and Improta, 1988; Improta and Cantarella, 1984; Sacco, 2014; Silcock, 1997; Wong and Heydecker, 2011; Wong and Wong, 2003; Wong, 1996; Yan et al., 2014). Some of the aforementioned papers, e.g., (Wong and Heydecker, 2011; Wong and Wong, 2003; Wong, 1996; Yan et al., 2014), are actually called lane-based methods in the literature; in addition to optimizing the structure of the signal group diagram and the duration of these green intervals simultane-

This chapter is based on the following paper: Fleuren, S. and Lefeber, E. (2016c). Optimizing fixed-time control at isolated intersections: Part I: A single green interval per traffic light. Technical Report DC 2016.067, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven, The Netherlands. Available at http://mn.wtb.tue.nl/~lefeber/do_download_pdf.php?id=170. Submitted to Transportation Science

ously, these methods also optimize the lane-use arrows at the intersection. The modelling approach of these lane-based methods is however the same as that of the group-based methods (besides the optimization of the lane-use arrows). In this chapter we use 'group-based' to refer to both group-based and lane-based methods. We do not consider the optimization of the lane-use arrows; this extension is the topic of Chapter 6.

To model the structure of a signal group diagram, the aforementioned group-based methods introduce one binary-valued design variable for each pair of competing (conflicting) signal groups, i.e., the aforementioned group-based methods have one binary-valued design variable for each two signal groups whose traffic streams cannot safely cross the intersection simultaneously. This binary variable indicates which of the two signal groups switches to effective green first during the interval $[0, T)$, where T is the period duration of the pre-timed controller. In this chapter we show that the values of these binary variables are not uniquely defined for each signal group diagram, i.e., multiple values for the binary-valued design variables may be associated with the same signal group diagram. Since these optimization problems are usually solved with a branch and bound method, this is not a desirable property; multiple nodes in the search tree might be related to the same solution, which might result in a large search tree, and as a consequence, large computation times.

In this chapter we propose a group-based approach that does not possess this undesirable property; this group-based method is based on the mathematical framework of (Serafini and Ukovich, 1989a). In that paper Serafini and Ukovich present a generic approach to model periodic processes. This approach has already been used extensively to obtain timetables for public transportation and we show that it can also be used to optimize pre-timed control at isolated intersections. In (Serafini and Ukovich, 1989b) this generic approach is applied to the traffic light setting; they give an implicit enumeration algorithm to find a signal group diagram with a fixed and given period duration. However, in this chapter we consider the period duration to be a design variable and we do not try to find a feasible signal group diagram only, but we try to find 'the best' signal group diagram, i.e., we try to find the signal group diagram that minimizes some objective function. The resulting optimization problem can be solved to optimality when the objective function is convex in the design variables. Commonly used objectives are: minimizing the period duration of the signal group diagram, maximizing the capacity of the intersection, and minimizing the average delay that road users experience (due to traffic light control) at the intersection. In the former two cases the objective function is linear. In the latter case the objective function may be convex, depending on the formula used to approximate the (average) delay that road users experience at a queue under pre-timed control. This convex function may also be approximated by a piecewise linear function; the optimization problem can then still be formulated as a mixed-integer linear programming problem. In Chapter 2 we have introduced some formulae that approximate the average delay that road users experience at a traffic light under pre-timed control. In (van den Broek et al., 2006) different approximations are compared with each other.

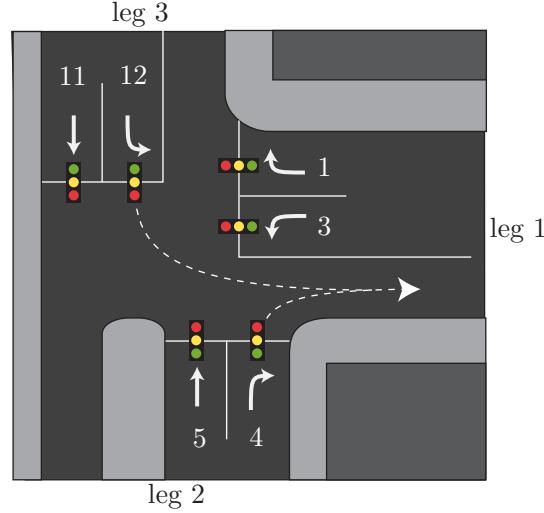
Especially, the formula that she proposes seems to be a very good approximation. The average delay approximations that have been introduced in Chapter 2 are all convex in the design variables.

The contributions of this chapter are the following. We formulate a novel group-based approach to find the optimal signal group diagram at an isolated intersection. This optimization problem adds an objective function to the linear constraints of (Serafini and Ukovich, 1989b). In addition, we have modified the constraints of (Serafini and Ukovich, 1989b) to allow the period duration to be a design variable. For the formulation of the optimization problem, we require an integral cycle basis of some graph. In this chapter we propose a different integral cycle basis than the one used in (Serafini and Ukovich, 1989b). For each cycle in this integral cycle basis we have one integral-valued design variable. For the integral cycle basis that we propose, many of these integral-valued design variables can only attain one value. Hence, for these integral-valued design variables we can fix their values. This is expected to reduce the time needed to solve the optimization problem. In this chapter, we also relate our approach to the currently existing group-based approaches and show that our integral-valued design variables aggregate the binary variables of the currently existing group-based approaches. By means of an extensive computational study in which we use real-life intersections, we compare this novel group-based approach with the currently existing ones. The results show superiority of the novel approach.

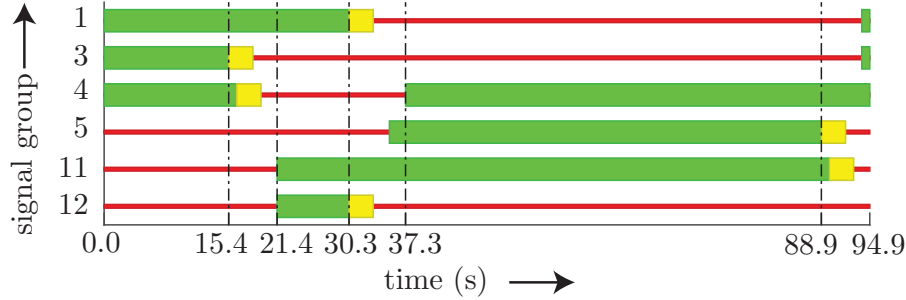
The organisation of this chapter is as follows. First, we formulate a mixed-integer programming (MIP) problem in Section 3.2. In that section we formulate the constraints of the MIP problem by using the mathematical framework of (Serafini and Ukovich, 1989a). Subsequently, we compare the proposed formulation of the MIP problem with the formulation of the aforementioned group-based approaches in Section 3.3; we relate the integral-valued design variables of both approaches and we compare both approaches by means of an extensive computational study. The summary can be found in Section 3.4.

3.2 Formulating the MIP problem

In this section we introduce a MIP formulation to optimize signal group diagrams. This formulation is based on the mathematical framework of (Serafini and Ukovich, 1989a). First, we consider an example: minimization of the (average) delay that road users experience at the T-junction visualized in Figure 3.1a; the optimal signal group diagram is visualized in Figure 3.1b and the associated switching times are given in Table 3.1. Subsequently, we give the MIP problem, including the different objective functions, for the generic case.



(a) Visualisation of the intersection. For each traffic light we have given the number of the corresponding signal group. These signal groups are numbered according to the standard Dutch numbering.



(b) The signal group diagram that minimizes the average delay that road users experience; we use the approximation of (van den Broek et al., 2006), see Section 2.2. This schedule has a period duration of 94.87 seconds. On average a road user experiences a delay of 26.416 seconds for this signal group diagram.

Figure 3.1: An example of a T-junction and its signal group diagram.

3.2.1 MIP formulation for a small example

In this section we formulate the MIP problem for a small example: minimization of the (average) delay that road users experience at the T-junction visualized in Figure 3.1a. In the upcoming section we give the required inputs for the optimization of a signal group diagram. Subsequently we introduce the real-valued design variables, we formulate the linear constraints and we introduce the objective function for this optimization problem.

Required inputs

In this section we give the inputs that are required to optimize a signal group diagram.

Signal groups Usually, the traffic lights at the intersection are divided amongst signal groups; the traffic lights in a signal group receive identical indications, i.e., all traffic lights

Signal group (i) (-)	t_i^G (s)	t_i^Y (s)	t_i^R (s)
1	93.87	30.35	33.35
3	93.87	15.43	18.43
4	37.35	16.43	19.43
5	35.35	88.87	91.87
11	21.43	89.87	92.87
12	21.43	30.35	33.35

Table 3.1: The times (rounded to hundreds of a second) at which the signal groups $i = 1, 3, 4, 5, 11, 12$ switch to green (t_i^G), yellow (t_i^Y) and red (t_i^R) for the signal group diagram in Figure 3.1b.

always display the same color. We require a set of signal groups \mathcal{S} for which we desire an optimized signal group diagram. For the example in Figure 3.1a we have:

$$\mathcal{S} = \{1, 3, 4, 5, 11, 12\},$$

where signal group $i \in \mathcal{S}$ contains only one traffic light, which is traffic light i . For this example each signal group contains only one traffic light; this is not necessarily the case in general.

In Chapter 2 we have discussed how to model the traffic that is waiting at the intersection by using queues, e.g., we can model the pedestrians waiting at a pedestrian crossing with a single queue. Let \mathcal{Q} denote the set of queues used to model the traffic waiting at the intersection. For each of the signal groups $i \in \mathcal{S}$, we require a set of queues \mathcal{Q}_i that it controls, i.e., traffic can depart from a queue $q \in \mathcal{Q}_i$ (on a first-in-first-out basis) during an effective green interval of signal group i and it cannot depart from this queue during an effective red interval of signal group i . For the intersection in Figure 3.1a we model each of the arrival lanes by a single queue. Therefore, each of the sets \mathcal{Q}_i contains only one queue; for ease of notation, we number the queue in the set \mathcal{Q}_i as follows for this example:

$$\mathcal{Q}_i = \{i\}, \quad i \in \mathcal{S}.$$

Thus, the traffic that is waiting at the intersection in Figure 3.1a is modelled with queues $\mathcal{Q} = \{1, 3, 4, 5, 11, 12\}$ and signal group $i \in \mathcal{S}$ controls queue i .

Arrival rates and saturation flow rates For each of the queues $q \in \mathcal{Q}$ we require an arrival rate λ_q that specifies how much traffic arrives at this queue per second. Furthermore, for each queue we require the saturation flow rate μ_q ; this saturation flow rate specifies the (maximum) amount of traffic that can depart from queue q per second during an effective green interval. The intersection in Figure 3.1a has the following arrival rates and saturation flow rates (in PCE/s):

$$\lambda_1 = \frac{320}{3600}, \lambda_3 = \frac{280}{3600}, \lambda_4 = \frac{180}{3600}, \lambda_5 = \frac{980}{3600}, \lambda_{11} = \frac{820}{3600}, \lambda_{12} = \frac{150}{3600},$$

$$\mu_1 = \frac{1615}{3600}, \mu_3 = \frac{1805}{3600}, \mu_4 = \frac{1615}{3600}, \mu_5 = \frac{1900}{3600}, \mu_{11} = \frac{1900}{3600}, \mu_{12} = \frac{1805}{3600}.$$

Note that the saturation flow rate associated with a right-turn movement, a through movement, and a left-turn movement equals $^{1615}/_{3600}$ PCE/s, $^{1900}/_{3600}$ PCE/s respectively $^{1805}/_{3600}$ PCE/s for this example. From the arrival rates and the saturation flow rates we can obtain the loads $\rho_q := \lambda_q/\mu_q$.

Starting lost time, ending lost time, and yellow time A traffic light has three indications: green, yellow, and red. In Chapter 2 we have motivated that, instead of these three indications, mathematically it suffices to have only two modes: effective green and effective red. The optimization problem optimizes the duration of the effective green and the effective red intervals of the signal groups $i \in \mathcal{S}$. A signal group diagram that specifies when each signal group is effective green and effective red can easily be transformed into a signal group diagram that specifies when each signal group is green, yellow and red. To this end, for each signal group $i \in \mathcal{S}$ we require the starting lost time l_i^s , the ending lost time l_i^e , and the yellow time Y_i , see also Section 2.1. For the intersection in Figure 3.1a we have:

$$\begin{aligned} l_i^s &= 1, & i \in \mathcal{S}, \\ l_i^e &= 1, & i \in \mathcal{S}, \\ Y_i^s &= 3, & i \in \mathcal{S}. \end{aligned}$$

With these lost times and yellow times we can transform the signal group diagram in Figure 3.1b, which specifies when each signal group is green, yellow and red, to the signal group diagram in Figure 3.2, which specifies when each signal group is effective green and effective red. The switching times associated with the latter signal group diagram are given in Table 3.2.

Remark 3.1. *By requiring the starting lost times, the ending lost times and the yellow times on a signal group basis, we implicitly assume that all queues in \mathcal{Q}_i , $i \in \mathcal{S}$ are simultaneously effective green and effective red. It is also possible to require these lost times and yellow times on a queue basis, i.e., for each queue in \mathcal{Q} . The effective green times of two queues in \mathcal{Q}_i , $i \in \mathcal{S}$ are then not necessarily equal. However, they are related. Consider for example two queues, say queue 1 and queue 2, that are both part of the same signal group $i \in \mathcal{S}$, i.e., $\{1, 2\} \subseteq \mathcal{Q}_i$. Assume that we require the lost times and the yellow times on a queue basis; let l_q^s , l_q^e , and Y_q be the starting lost time, ending lost time, and yellow time of queue $q = 1, 2$. Since queue 1 and queue 2 receive the same indication,*

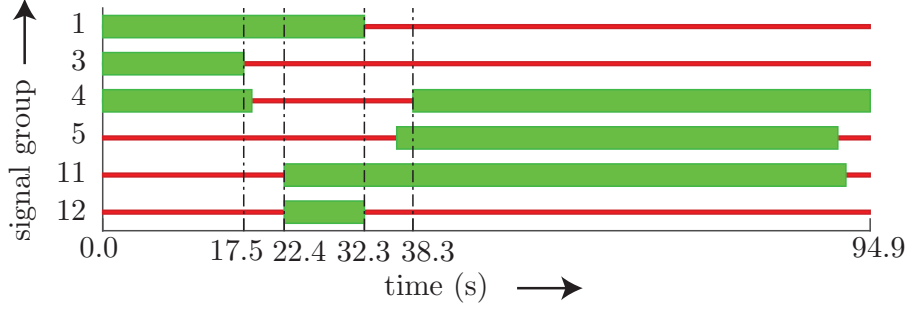


Figure 3.2: The signal group diagram of Figure 3.1b that visualizes when each signal group is effective green and effective red. This schedule has a period duration of 94.87 seconds. On average a road user experiences a delay of 26.416 seconds for this signal group diagram.

Signal group (i) (-)	t_i^g (s)	t_i^r (s)
1	0	32.35
3	0	17.43
4	38.35	18.43
5	36.35	90.87
11	22.43	91.87
12	22.43	32.35

Table 3.2: The times (rounded to hundreds of a second) at which the signal groups $i = 1, 3, 4, 5, 11, 12$ switch to effective green (t_i^g) and effective red (t_i^r) for the signal group diagram in Figure 3.2.

the effective green time g_1 of queue 1 and the effective green time g_2 of queue 2 are then related according to:

$$g_1 + l_1^s + l_1^e - Y_1 = g_2 + l_2^s + l_2^e - Y_2.$$

In this thesis, we assume w.l.o.g. that these lost times and yellow times are given on a signal group basis; all traffic lights associated with signal group i then have the same effective green (effective red) time and, as a result, the effective green time g_i of signal group i is also the effective green time of all queues in signal group i .

Conflicts When two traffic streams use a common part of the intersection that they cannot safely cross simultaneously, we call their corresponding signal groups conflicting. We require a set Ψ_S of conflicting signal groups. For the intersection in Figure 3.1a we have:

$$\Psi_S = \{\{1, 5\}, \{3, 5\}, \{3, 11\}, \{3, 12\}, \{4, 12\}, \{5, 12\}\}.$$

Minimum clearance times For each pair of conflicting signal groups $\{i, j\} \in \Psi_S$, the time between signal group i switching to a red indication and signal group j switching to

a green indication must exceed some minimum amount of time. This minimum amount of time ensures that the traffic from signal group j can safely cross the intersection without encountering any traffic from signal group i .

Using the starting lost times and the ending lost times, we can convert this minimum amount of time between signal group i switching to a red indication and signal group j switching to a green indication, to a minimum amount of time $\underline{c}_{i,j}$ between signal group i becoming effective red and signal group j becoming effective green. This minimum amount of time $\underline{c}_{i,j}$ is called a minimum clearance time; the interval between signal group i switching to effective red and a conflicting signal group j switching to effective green is called a clearance interval; its duration is called a clearance time.

The minimum clearance time $\underline{c}_{i,j}$ depends on the layout of the intersection amongst other factors. Consider for example the conflict between signal group 4 and signal group 12 in Figure 3.1a. The area where the traffic streams of signal groups 4 and 12 conflict is relatively close to the stop line of signal group 4 and relatively far away from the stop line of signal group 12. As a consequence the minimum clearance time $\underline{c}_{12,4}$ is larger than the minimum clearance time $\underline{c}_{4,12}$. We give the minimum clearance times for the intersection in Figure 3.1a below:

$$\begin{aligned} \underline{c}_{1,5} &= 4, & \underline{c}_{3,5} &= 4, & \underline{c}_{3,11} &= 5, & \underline{c}_{3,12} &= 5, & \underline{c}_{4,12} &= 4, & \underline{c}_{5,12} &= 4, \\ \underline{c}_{5,1} &= 4, & \underline{c}_{5,3} &= 4, & \underline{c}_{11,3} &= 3, & \underline{c}_{12,3} &= 5, & \underline{c}_{12,4} &= 6, & \underline{c}_{12,5} &= 4. \end{aligned}$$

For the T-junction of Figure 3.1 it holds that the minimum clearance times $\underline{c}_{i,j}$, $\{i, j\} \in \Psi_{\mathcal{S}}$ are non-negative. However, we do also allow negative clearance times; in case that $\underline{c}_{i,j}$, $\{i, j\} \in \Psi_{\mathcal{S}}$ is negative, signal group j is allowed to become effective green at most $\text{abs}(\underline{c}_{i,j})$ seconds before signal group i becomes effective red, where $\text{abs}(x)$ is the absolute value of x . In Appendix C.1 we motivate the use of negative clearance times.

We elaborate on two relations between these minimum clearance times that one might believe to be true for any real-life intersection, but which might *not* hold for some of them. Consider three signal groups i_1, i_2 and i_3 that are conflicting with each other. The first relation is a triangular equation:

$$\underline{c}_{i_1, i_3} \geq \underline{c}_{i_1, i_2} + \underline{c}_{i_2, i_3}.$$

In Appendix C.2 we give an intersection for which this triangular inequality is probably not satisfied. For the second relation, consider signal groups i_1, i_2, \dots, i_{N+1} , where $i_{N+1} := i_1$ and $\{i_k, i_{k+1}\} \in \Psi_{\mathcal{S}}$, $k = 1, \dots, N$. One might believe the following circuital inequality to be true for any real-life intersection:

$$\sum_{k=1}^N \underline{c}_{i_k, i_{k+1}} > 0,$$

which implies that a sequence of minimum clearance times that starts at some signal group i_1 and ends at that same signal group i_1 is non-negative. In Appendix C.3 we give an

intersection for which this circutal inequality is possibly not satisfied. In that appendix we also motivate that this circutal constraint is satisfied for $N = 2$, i.e., we motivate the following strict inequality for each pair of conflicting signal groups $\{i, j\} \in \Psi_{\mathcal{S}}$:

$$\underline{c}_{i,j} + \underline{c}_{j,i} > 0.$$

Minimum and maximum period duration Some safety measures prevent road users from becoming extremely impatient and believing that the traffic lights are defective or ensure that the different traffic streams can safely cross the intersection. One of these safety measures restricts the duration of the period. We have a (strictly positive) lower bound ($\underline{T} > 0$) and an upper bound (\bar{T}) on the period duration. For the example of Figure 3.1a we have:

$$\underline{T} = 30, \quad \bar{T} = 120.$$

Minimum and maximum green (red) times Another safety measure is to restrict the duration of a green (red) interval; road users may find it unfair whenever their red interval is very long or someone else receives a very long green interval. Furthermore, road users might not expect the traffic light to be green or red for a very short amount of time. Therefore, we have restrictions on the duration of a green (red) interval. These restrictions can be converted to restrictions on the duration of an effective green (effective red) interval. The duration of an effective green interval of signal group $i \in \mathcal{S}$ is bounded from below by the minimum effective green time \underline{g}_i and is bounded from above by the maximum effective green time \bar{g}_i . Similarly, we also have a (strictly positive) lower bound $\underline{r}_i > 0$ and an upper bound \bar{r}_i on the duration of an effective red interval of signal group $i \in \mathcal{S}$. For the T-junction of Figure 3.1a we have:

$$\begin{aligned} \underline{g}_i &= 6, \quad i \in \mathcal{S}, & \bar{g}_i &= \infty, \quad i \in \mathcal{S}, \\ \underline{r}_i &= 6, \quad i \in \mathcal{S}, & \bar{r}_i &= \infty, \quad i \in \mathcal{S}. \end{aligned}$$

Real-valued design variables

The goal is to find the optimal signal group diagram for the isolated intersection of Figure 3.1a. We have depicted this optimal signal group diagram in Figure 3.2. Let T be the period duration of the signal group diagram. Furthermore, let \textcircled{i} and \textcircled{i} denote periodic events. The periodic event \textcircled{i} (periodic event \textcircled{i}) is defined as the start (end) of the effective green interval of signal group $i \in \mathcal{S}$. Define \mathcal{E} as the set of the following periodic events:

$$\mathcal{E} = \{\textcircled{i} \mid i \in \mathcal{S}\} \cup \{\textcircled{i} \mid i \in \mathcal{S}\}.$$

i	$f(\textcircled{i})$	$f(\textcircled{\textcircled{i}})$
1	0/9487	3235/9487
3	0/9487	1743/9487
4	3835/9487	1843/9487
5	3635/9487	9087/9487
11	2243/9487	9187/9487
12	2243/9487	3235/9487

Table 3.3: The switching fractions $f(\textcircled{i})$ ($f(\textcircled{\textcircled{i}})$) at which the effective green interval of signal group $i \in \mathcal{S}$ starts (ends) for the signal group diagram in Figure 3.2.

With each of the events $\varepsilon \in \mathcal{E}$ we can associate a fraction $f(\varepsilon) \in [0, 1)$; this fraction $f(\varepsilon) \in [0, 1)$ is defined as the time, expressed as a fraction of the period duration T , at which the event ε is scheduled. In Table 3.3 we give these values for $f(\varepsilon)$ for the signal group diagram in Figure 3.2. A signal group diagram is completely specified by the period duration T and the fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$. However, we do not use the variables T and $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ directly in the MIP formulation. Below we introduce the real-valued design variables of the MIP problem. We visualize all design variables in bold; the variables that depend on these design variables are not visualized in bold.

From the periodicity of the signal group diagram it follows that the periodic event $\varepsilon \in \mathcal{E}$ occurs at times $(f(\varepsilon) + k)T$, $k \in \mathbb{Z}$. Let $\gamma(\varepsilon_1, \varepsilon_2)$ denote the time between an occurrence of periodic event ε_1 and an occurrence (the next or the previous occurrence) of periodic event ε_2 expressed as a fraction of the period duration. Thus, $\gamma(\varepsilon_1, \varepsilon_2)$ is defined as follows:

$$\gamma(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2) \quad (3.1)$$

for some integer $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, 1\}$. The variables $\gamma(\varepsilon_1, \varepsilon_2)$ that are subject to a safety constraint, constitute the real-valued design variables of the MIP problem together with the reciprocal of the period duration $\mathbf{T}' := 1/T$. All these real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$ and their integers $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously, i.e., for each of these variables $\gamma(\varepsilon_1, \varepsilon_2)$, both its own value and the value of its associated integer $z(\varepsilon_1, \varepsilon_2)$ are uniquely defined for each signal group diagram. We prove this unambiguity at the end of this section; we do so by proving for each of the real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$ that its value is included in some interval with a length that is strictly smaller than one, e.g., $\gamma(\varepsilon_1, \varepsilon_2) \in [0, 1)$. Furthermore, at the end of this section we show how to obtain the fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ from the solution of the MIP problem.

Objective function

A signal group $i \in \mathcal{S}$ controls the access of traffic to the intersection for the queues in the set \mathcal{Q}_i . For the T-junction in Figure 3.1a we want to minimize the average weighted

delay at the intersection:

$$D = \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q, \quad (3.2a)$$

where d_q is the (approximated) delay at queue q and w_q is the weight factor associated with this queue. To minimize the average delay of a road user at the intersection, we choose the weights proportional to the arrival rates:

$$w_q = \frac{\lambda_q}{\Lambda},$$

where,

$$\Lambda = \sum_{q \in \mathcal{Q}} \lambda_q.$$

We approximate d_q with the approximation of (van den Broek et al., 2006), which is the sum of the deterministic delay term (2.2) and the stochastic delay term (2.3), which gives the following approximation for each queue $q \in \mathcal{Q}_i$:

$$d_q = \frac{r'_i}{2(1 - \rho_q)\rho_q} \left(\frac{\sigma_q^2}{\mu_q(1 - \rho_q)} + \frac{\rho_q r'_i}{\mathbf{T}'} + \frac{r'_i \rho_q^2 \sigma_q^2}{\mu_q(1 - r'_i)^2(1 - r'_i - \rho_q)(1 - \rho_q)} \right), \quad (3.2b)$$

where $r'_i := \gamma(\textcircled{i}, \textcircled{i})$. If d_q is convex in the real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$ and \mathbf{T}' , then also the average weighted delay D is convex in these real-valued design variables; in Appendix C.4 we prove that (3.2b) is indeed convex.

Remark 3.2. For each queue $q \in \mathcal{Q}$ at which vehicles arrive, the arrival rate λ_q is expressed in passenger car equivalent per second (PCE/s); the unit 'passenger car equivalent' is used to convert a mixed traffic stream to an equivalent traffic stream of only passenger cars, see Section 2.3. For queues $q \in \mathcal{Q}$ representing cyclist crossings and pedestrian crossings, these rates are expressed in cyclists per second, respectively pedestrians per second. To calculate the average delay that road users experience, the weight w_q actually needs to equal the ratio of the arrival rate in road users per second to the total arrival rate (expressed in road users per second) at the intersection. Therefore, to calculate these weights w_q , $q \in \mathcal{Q}$ we also require the arrival rate expressed in road users per second for all queues. Note that for some queues this arrival rate cannot be obtained directly from λ_q ; a bus may for example count as 2 passenger cars (see Table 2.1), however, the average number of passengers on this bus might not equal that of two (average) passenger cars. In this thesis we assume for reasons of simplicity that both arrival rates (expressed in PCE/s and expressed in road users per second) are the same.

Linear constraints

A signal group diagram must satisfy some safety constraints. These safety constraints either prevent road users from becoming extremely impatient and believing that the

traffic lights are defective or ensure that the different traffic streams can safely cross the intersection. We have bounds on the period duration, which results in the following constraint on its reciprocal \mathbf{T}' :

$$1/\overline{T} \leq \mathbf{T}' \leq 1/\underline{T}. \quad (3.3a)$$

The effective green time of signal group $i \in \mathcal{S}$ is bounded from below and from above:

$$0 \leq \underline{g}_i \mathbf{T}' \leq \gamma(\textcircled{i}, \textcircled{i}) \leq \overline{g}_i \mathbf{T}', \quad (3.3b)$$

Note that $\gamma(\textcircled{i}, \textcircled{i})T$ is the effective green time of signal group i . Hence, (3.3b) restricts the effective green time of signal group i to be at least \underline{g}_i and at most \overline{g}_i . The effective red time of signal group $i \in \mathcal{S}$ is bounded from below and bounded from above in the same way:

$$0 < \underline{r}_i \mathbf{T}' \leq \gamma(\textcircled{i}, \textcircled{i}) \leq \overline{r}_i \mathbf{T}'. \quad (3.3c)$$

Each signal group $i \in \mathcal{S}$ must be stable, i.e., each queue $q \in \mathcal{Q}_i$ must be effective green for at least a fraction ρ_q of the period duration; on average, the amount of traffic that arrives at queue q during a period of length T can then depart during an effective green interval. Hence, it should hold for each signal group $i \in \mathcal{S}$ that:

$$0 < \max_{q \in \mathcal{Q}_i} \rho_q \leq \gamma(\textcircled{i}, \textcircled{i}). \quad (3.3d)$$

We call (3.3d) a *stability constraint*. Many approximate formulae are only valid when stability is guaranteed, e.g., the ones from (Miller, 1963; van den Broek et al., 2006; Webster, 1958).

For each two conflicting signal groups $\{i, j\} \in \Psi_{\mathcal{S}}$, minimum clearance times must be satisfied, which ensures that each traffic stream can safely cross the intersection without encountering traffic from a conflicting signal group:

$$\underline{c}_{i,j} \mathbf{T}' \leq \gamma(\textcircled{i}, \textcircled{j}). \quad (3.3e)$$

Remark 3.3. When the arrival process of traffic at the queues $q \in \mathcal{Q}_i$ is stochastic, stability constraint (3.3d) is not sufficient; we then require the strict inequality $\max_{q \in \mathcal{Q}_i} \rho_q < \gamma(\textcircled{i}, \textcircled{i})$ to ensure that the total amount of traffic waiting at the queues $q \in \mathcal{Q}_i$ remains bounded. This strict inequality can be forced by subtracting a small value $\epsilon > 0$ from the right-hand side of stability constraint (3.3d).

Remark 3.4. Instead of only forcing each signal group $i \in \mathcal{S}$ to be stable, it is also possible to force a signal group $i \in \mathcal{S}$ to have a degree of saturation of at most $100 \cdot s_i^{\max}$ percent; when $s_i^{\max} \in [0, 1)$, then signal group $i \in \mathcal{S}$ is stable and it remains stable when the arrival rates increase with a growth factor strictly smaller than $1/s_i^{\max}$. For example, when a signal group i has a degree of saturation of at most 90 percent ($s_i^{\max} := 0.9$), this

signal group can handle a growth factor up to (but not including) $10/9$. We can force signal group $i \in \mathcal{S}$ to have a degree of saturation of at most $100 \cdot s_i^{\max}$ percent by replacing its stability constraint (3.3d) with the following constraint:

$$\max_{q \in \mathcal{Q}_i} \rho_q / s_i^{\max} \leq \gamma((i), (i)) - \epsilon,$$

where ϵ is some small positive number.

Integer constraints

The variables $\gamma(\varepsilon_1, \varepsilon_2)$ are also related via circuital constraints, which are induced by the periodicity of the signal group diagram. These circuital constraints are best explained by using a directed graph $G = (V, A)$ with vertices V and arcs A :

$$V := \{(i) \mid i \in \mathcal{S}\} \cup \{(\bullet i) \mid i \in \mathcal{S}\},$$

$$A := A_g \cup A_r \cup A_c,$$

where,

$$A_g := \{((i), (\bullet i)) \mid i \in \mathcal{S}\},$$

$$A_r := \{((\bullet i), (i)) \mid i \in \mathcal{S}\},$$

$$A_c := \{((\bullet i), (\bullet j)) \mid \{i, j\} \in \Psi_{\mathcal{S}}\}.$$

The set of vertices V equals the set of periodic events \mathcal{E} . Therefore, with each of these vertices we can associate the fraction $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$; recall that $f(\varepsilon)$ is defined as the time (expressed as a fraction of the period duration) at which the event ε is scheduled. The set of arcs A represents the set of real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$, i.e., the set A consists of the arcs $(\varepsilon_1, \varepsilon_2)$ for which $\gamma(\varepsilon_1, \varepsilon_2)$ is subject to a (safety) constraint. Therefore, with each arc $(\varepsilon_1, \varepsilon_2) \in A$ we associate the variable $\gamma(\varepsilon_1, \varepsilon_2)$, see also Figure 3.3; recall that $\gamma(\varepsilon_1, \varepsilon_2)$ is the time (expressed as a fraction of the period duration) between an occurrence of the event ε_1 and an occurrence of the event ε_2 . Arc $a \in A_g$ of the graph G represents an effective green interval, arc $a \in A_r$ represents an effective red interval, and arc $a \in A_c$ represents a clearance interval. We refer to graph G as the *constraint graph*. We define a path, a walk, a cycle and a circuit in the constraint graph G as follows:

Definition 3.1 (Walk). *A walk is a sequence of vertices $v_1, v_2, \dots, v_N \in V$ for which each two subsequent vertices v_k and v_{k+1} are connected via a directed arc $(v_k, v_{k+1}) \in A$.*

A walk may only traverse arcs in the forward direction (from tail to head). A path may however also traverse arcs in the backward direction (from head to tail):

Definition 3.2 (Path). *A path is a sequence of vertices $v_1, v_2, \dots, v_N \in V$ for which each two subsequent vertices v_k and v_{k+1} are connected via either a directed arc $(v_k, v_{k+1}) \in A$ or a directed arc $(v_{k+1}, v_k) \in A$; a path traverses each arc $a \in A$ at most once.*

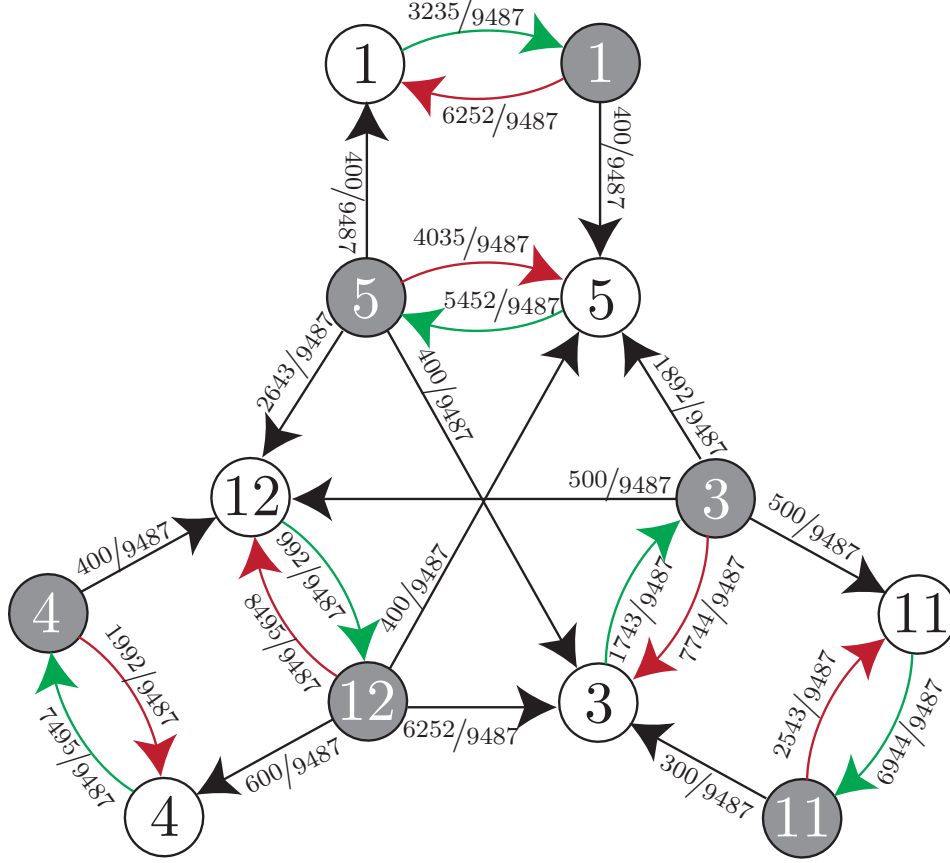


Figure 3.3: The constraint graph $G = (V, A)$ that corresponds to the signal group diagram in Figure 3.2. The arcs in green (red) visualise effective green (effective red) intervals. The arcs in black represent clearance intervals. Attached to the tail of each arc is the corresponding value of $\gamma(\varepsilon_1, \varepsilon_2)$.

A path can be represented by the sets \mathcal{P}^+ and \mathcal{P}^- , which denotes the set of arcs that this path traverses in the forward direction (from tail to head) respectively the set of arcs that this path traverses in the backward direction (from head to tail). Reorienting the arcs in \mathcal{P}^- results in a walk that traverses each arc at most once. We define $\mathcal{P} := \mathcal{P}^+ \cup \mathcal{P}^-$. A cycle is defined in accordance with (Kavitha and Krishna, 2009; Serafini and Ukovich, 1989b):

Definition 3.3 (Cycle). *A cycle is a closed path, i.e., a cycle is a path for which $v_1 = v_N$.*

Thus, a cycle is also allowed to traverse arcs in the backward direction. We can represent a cycle by the sets \mathcal{C}^+ and \mathcal{C}^- , which denotes the set of arcs that this cycle traverses in the forward direction (from tail to head) respectively the set of arcs that this cycle traverses in the backward direction (from head to tail). Reorienting the arcs in \mathcal{C}^- results in a closed walk that traverses each arc at most once. We define $\mathcal{C} := \mathcal{C}^+ \cup \mathcal{C}^-$.

Definition 3.4 (Circuit). *A circuit is a cycle for which the vertices v_1, v_2, \dots, v_{N-1} are all distinct, i.e., a cycle for which each vertex is visited at most once.*

Consider a path \mathcal{P} in the constraint graph G . Let \mathcal{P}^+ and \mathcal{P}^- be the set of arcs that this path traverses in the forward direction (from tail to head) respectively the set of arcs that this path traverses in the backward direction (from head to tail). For example, the constraint graph G in Figure 3.3 contains a path with forward arcs $\mathcal{P}^+ = \{((3), (3)), ((3), (12))\}$ and backward arcs $\mathcal{P}^- = \{((5), (12)), ((5), (5))\}$. Each such path can be associated with a journey through time; see for example Figure 3.4 for such a journey through time. Let v_s and v_e be the vertex at which path \mathcal{P} starts respectively ends. The time journey starts at the time $f(v_s)T$; by definition, event v_s occurs at this time. For each arc $(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^+$ ($(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^-$) this time journey jumps $\gamma(\varepsilon_1, \varepsilon_2)T$ seconds forward (backward) in time. The time journey then ends at a time at which event v_e occurs. To be more specific:

$$f(v_s) + \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^-} \gamma(\varepsilon_1, \varepsilon_2) = f(v_e) + z,$$

where z is some integer. To verify this equation, plug in the definition $\gamma(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)$. We then obtain $z := \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^+} z(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^-} z(\varepsilon_1, \varepsilon_2)$, which is indeed integral. This equation can be interpreted as follows. The term $f(v_s)$ is the fraction (the time expressed as a fraction of the period duration) at which the corresponding journey through time starts. The term $\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^+} \gamma(\varepsilon_1, \varepsilon_2)$ corresponds to the forward jumps in time of the associated time journey, and the term $\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^-} \gamma(\varepsilon_1, \varepsilon_2)$ corresponds to the backward jumps in time of the associated time journey. The sum of these three terms is some fraction at which event v_e occurs. From the periodicity of the signal group diagram it follows that this fraction equals $f(v_e) + z$ for some integer z .

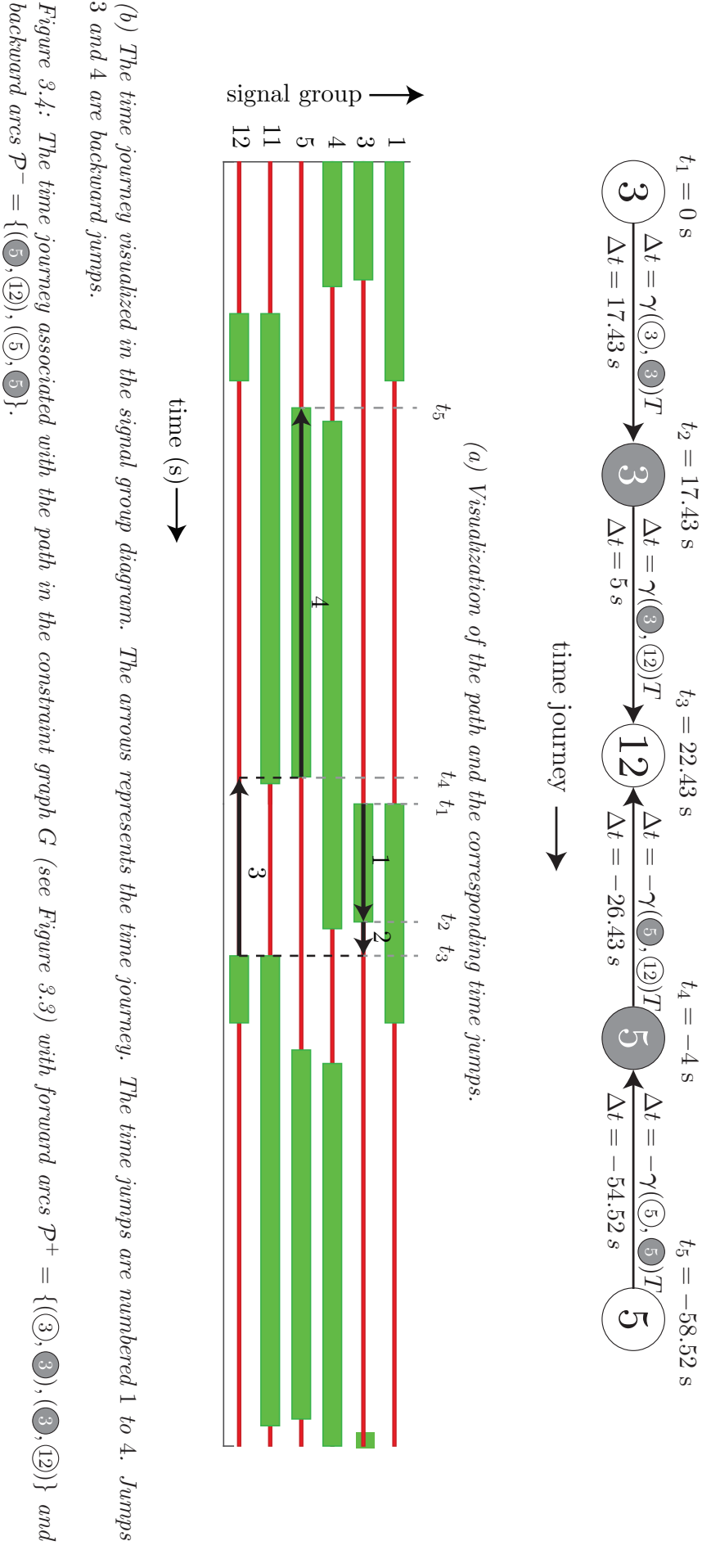
We explain this equation with an example. Consider again the path visualised in Figure 3.4. The path starts at the vertex (3) and the corresponding journey through time starts at the time:

$$\begin{aligned} t_1 &:= f((3))T, \\ &= 0 \text{ seconds.} \end{aligned}$$

At this time the event (3) takes place. The path then goes to the vertex (3) via a forward arc $((3), (3)) \in \mathcal{P}^+$ and the corresponding time journey jumps $\Delta t = \gamma((3), (3))T = 17.43$ seconds forward in time to the time:

$$\begin{aligned} t_2 &:= f((3))T + \gamma((3), (3))T, \\ &= f((3))T + z((3), (3))T, \\ &= 17.43 \text{ seconds,} \end{aligned}$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. To obtain the latter expression we have used the definition (3.1), which is the definition of $\gamma(\varepsilon_1, \varepsilon_2)$. Note that, by definition, event (3) occurs at time t_2 . Next, the path goes to the vertex (12) via the arc $((3), (12)) \in \mathcal{P}^+$ and the corresponding time journey jumps $\Delta t = \gamma((3), (12))T = 5$ seconds forward in time to the time:



$$\begin{aligned}
t_3 &:= t_2 + \gamma(\textcircled{3}, \textcircled{12})T, \\
&= f(\textcircled{12})T + z(\textcircled{3}, \textcircled{3})T + z(\textcircled{3}, \textcircled{12})T, \\
&= 22.43 \text{ seconds},
\end{aligned}$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. We have again used (3.1). Note that, by definition, event $\textcircled{12}$ is scheduled at time t_3 . Subsequently, the path goes to the vertex $\textcircled{5}$ via a backward arc $(\textcircled{5}, \textcircled{12}) \in P^-$ and the corresponding time journey jumps $\gamma(\textcircled{5}, \textcircled{12})T = 26.43$ seconds backward in time to the time:

$$\begin{aligned}
t_4 &:= t_3 - \gamma(\textcircled{5}, \textcircled{12})T, \\
&= f(\textcircled{5})T + z(\textcircled{3}, \textcircled{3})T + z(\textcircled{12}, \textcircled{12})T - z(\textcircled{5}, \textcircled{12})T, \\
&= -4 \text{ seconds},
\end{aligned}$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. At this time the event $\textcircled{5}$ occurs. Finally, the path goes to the vertex $\textcircled{5}$ via a backward arc $(\textcircled{5}, \textcircled{5}) \in P^-$ and the corresponding time journey jumps $\gamma(\textcircled{5}, \textcircled{5})T = 54.52$ seconds backward in time to the time:

$$\begin{aligned}
t_5 &:= t_4 - \gamma(\textcircled{5}, \textcircled{5})T, \\
&= f(\textcircled{5})T + z(\textcircled{3}, \textcircled{3})T + z(\textcircled{12}, \textcircled{12})T - z(\textcircled{5}, \textcircled{12})T - z(\textcircled{5}, \textcircled{5})T, \\
&= -58.52 \text{ seconds},
\end{aligned}$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. Indeed at time t_5 event $v_e := \textcircled{5}$ occurs.

For any cycle \mathcal{C} in the constraint graph G it holds that v_s equals v_e . As a consequence, the time journey of a cycle \mathcal{C} gives the following circuital constraint on the variables $\gamma(\varepsilon_1, \varepsilon_2)$:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2) = \mathbf{z}_{\mathcal{C}}, \quad (3.3f)$$

where $\mathbf{z}_{\mathcal{C}} \in \mathbb{Z}$ and where \mathcal{C}^+ respectively \mathcal{C}^- is the set of arcs that the cycle \mathcal{C} traverses in the forward direction (from tail to head) and the set of arcs that the cycle \mathcal{C} traverses in the backward direction (from head to tail). For example, Figure 3.3 has a cycle \mathcal{C} with forward arcs $\mathcal{C}^+ = \{(\textcircled{3}, \textcircled{3}), (\textcircled{3}, \textcircled{12}), (\textcircled{5}, \textcircled{3})\}$ and backward arcs $\mathcal{C}^- = \{(\textcircled{5}, \textcircled{12})\}$. Including constraint (3.3f) for every cycle \mathcal{C} ($\mathbf{z}_{\mathcal{C}}$ is an integral-valued design variable) would ensure periodicity; these constraints ensure that the time between two occurrences of the same periodic event $\varepsilon \in \mathcal{E}$ is an integral multiple of the period duration. We refer to these constraints as the *cycle periodicity constraints*.

For some cycles in the constraint graph G we need to fix the value for $\mathbf{z}_{\mathcal{C}}$. Consider the cycle $\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \textcircled{i}), (\textcircled{i}, \textcircled{j}), (\textcircled{j}, \textcircled{j}), (\textcircled{j}, \textcircled{i})\}$, where $\{i, j\} \in \Psi_S$. Since, the

effective green intervals of signal group i and signal group j must occur within one period, it should hold that:

$$\gamma((i, \odot)) + \gamma((\odot, j)) + \gamma((j, \odot)) + \gamma((\odot, i)) = 1, \quad \{i, j\} \in \Psi_{\mathcal{S}}, \quad (3.3g)$$

which implies that each period consists of an effective green interval of signal group i , a clearance interval from signal group i to signal group j , an effective green interval of signal group j and a clearance interval from signal group j to signal group i . Furthermore, we fix the value of $z_{\mathcal{C}}$ to one for each cycle $\mathcal{C} = \mathcal{C}^+ = \{((i, \odot), (\odot, i)), (\odot, i), (i, \odot)\}$, $i \in \mathcal{S}$:

$$\gamma((i, \odot)) + \gamma((\odot, i)) = 1, \quad i \in \mathcal{S}, \quad (3.3h)$$

which implies that the effective green interval of signal group i plus the effective red interval of signal group i constitute one period.

In general, the number of cycles in the constraint graph G may grow exponentially in the number of vertices. Hence, formulating constraint (3.3f) for every cycle in the constraint graph G may be intractable. However, it appears that we only have to formulate this constraint for some subset of cycles, which is called *an integral cycle basis*; this constraint is then automatically satisfied for every cycle in the constraint graph G . Let $|A|$ denote the number of elements in the set A . Let $C \in \{-1, 0, +1\}^{|A|}$ be a row vector associated with the cycle \mathcal{C} such that for each arc $a \in A$:

$$C(a) = \begin{cases} +1 & \text{if } a \in \mathcal{C}^+, \\ 0 & \text{if } a \notin \mathcal{C}, \\ -1 & \text{if } a \in \mathcal{C}^-. \end{cases}$$

We can then rewrite circuital constraint (3.3f) to $C\gamma = z_{\mathcal{C}}$, where γ is a column vector containing the variables $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$. We refer to the vector C as the *cycle-arc incidence vector* of the cycle \mathcal{C} . The cycle-arc incidence vectors C of the cycles in the constraint graph G generate a space, which we call the cycle space of the constraint graph G . Let $\nu(G)$ be the number of connected components of the constraint graph G ; no arc exists between each two vertices that are in different connected components. The cycle space has a dimension $d = |A| - |V| + \nu(G)$, see for example (Liebchen and Peeters, 2002). It suffices to formulate the cycle periodicity constraint (3.3f) only for the cycles in some cycle basis; this constraint is then automatically satisfied for each cycle in the constraint graph G . However, not just any cycle basis suffices: we need an integral cycle basis, which is defined as follows:

Definition 3.5 (Integral cycle basis). *An integral cycle basis is a set of cycles $\{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ such that any cycle \mathcal{C} in G can be written as:*

$$C = \alpha_1 C_1 + \dots + \alpha_d C_d,$$

where $\alpha_1, \dots, \alpha_d \in \mathbb{Z}$.

Such an integral cycle basis exists for each graph G , see for example (Kavitha et al., 2009). It is easy to prove that it suffices to include the cycle periodicity constraint (3.3f) only for the cycles in an integral cycle basis $\{\mathcal{C}_1, \dots, \mathcal{C}_d\}$. Assume that circuital constraint (3.3f) is satisfied for the cycles $\mathcal{C}_1, \dots, \mathcal{C}_d$ and define $\mathbf{z}_{\mathcal{C}_i} := C_i \boldsymbol{\gamma} \in \mathbb{Z}$, $i = 1, \dots, d$. For each cycle \mathcal{C} we can then find $\alpha_1, \dots, \alpha_d \in \mathbb{Z}$ such that:

$$C\boldsymbol{\gamma} = (\alpha_1 C_1 + \dots + \alpha_d C_d) \boldsymbol{\gamma} = \alpha_1 \mathbf{z}_{\mathcal{C}_1} + \dots + \alpha_d \mathbf{z}_{\mathcal{C}_d} \in \mathbb{Z},$$

which proves that circuital constraint (3.3f) is automatically satisfied for all other cycles \mathcal{C} in the constraint graph G . This also indicates why not just any cycle basis would suffice; for such a cycle basis the values α_i , $i = 1, \dots, d$ might not be integral and as a result $\alpha_1 \mathbf{z}_{\mathcal{C}_1} + \dots + \alpha_d \mathbf{z}_{\mathcal{C}_d}$ might not be integral as well. Hence, when constraint (3.3f) is satisfied for all the cycles in some (non-integral) cycle basis of the constraint graph G , this does not necessarily imply that constraint (3.3f) is satisfied for each cycle in the constraint graph G . In (Liebchen and Peeters, 2002) an example is given that shows this effect.

Obtaining an integral cycle basis of the constraint graph To formulate the cycle periodicity constraints (3.3f), we require an integral cycle basis of the constraint graph G . From the definition of the constraint graph $G = (V, A)$ it follows that $|V| = 2|\mathcal{S}|$ and $|A| = 2|\Psi_{\mathcal{S}}| + 2|\mathcal{S}|$. Therefore, the number of cycles in this integral cycle basis equals:

$$d := |A| - |V| + \nu(G) = 2|\Psi_{\mathcal{S}}| + \nu(G),$$

where $\nu(G)$ is the number of connected components of the constraint graph G . For each of the d cycles in the integral cycle basis, we have one integral-valued design variable. Define the slack of such an integral-valued design variable to be the difference between the smallest and the largest value that this variable can attain for any solution to the linear constraints (3.3). To improve the computation time required to solve the optimization problem we would like the slack of the integral-valued design variables to be as small as possible. Therefore, some integral cycle bases $\mathcal{B} := \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ may be better than others to formulate the optimization problem: a smaller slack in the integral-valued design variables $\mathbf{z}_{\mathcal{C}}$, $\mathcal{C} \in \mathcal{B}$ relates to a smaller computation time needed to solve the MIP problem. This claim is supported by computational studies, e.g., the ones from (Liebchen, 2003; Wünsch, 2008). In this section we attempt to find an integral cycle basis for which the slack in the integral variables is small. To this end, we define the *width* of a cycle \mathcal{C} as follows. Assume for the moment that we have some integral cycle basis \mathcal{B} that we use to formulate the constraints (3.3f). For each cycle \mathcal{C} in the constraint graph G , define $\underline{z}_{\mathcal{C}}$ ($\bar{z}_{\mathcal{C}}$) to be the minimum (maximum) value that:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2), \quad (3.4)$$

can attain for any solution to the linear constraints (3.3); these values can be obtained by solving two mixed-integer linear programming (MILP) problems: one minimizing (3.4)

subject to the constraints (3.3), and one maximizing (3.4) subject to constraints (3.3). Note that the value for $\underline{z}_{\mathcal{C}}$ ($\bar{z}_{\mathcal{C}}$) is integral and independent of the integral cycle basis that is used to formulate (3.3f); each integral cycle basis poses the same restriction on the real-valued design variables \mathbf{T}' and $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$. We define the width $w_{\mathcal{C}}$ of the cycle \mathcal{C} as follows:

$$w_{\mathcal{C}} := \bar{z}_{\mathcal{C}} - \underline{z}_{\mathcal{C}}.$$

This width $w_{\mathcal{C}}$ equals the slack of the integral-valued design variable $\underline{z}_{\mathcal{C}}$. Consider an integral cycle basis $\mathcal{B} := \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$. The vector of integral-valued design variables $\mathbf{z} = (z_{\mathcal{C}_1}, \dots, z_{\mathcal{C}_d})$ can then, possibly, attain $w_{\mathcal{B}} := \prod_{i=1}^d (w_{\mathcal{C}_i} + 1)$ different values; when enumerating over all values of \mathbf{z} in a brute force manner, then $w_{\mathcal{B}}$ iterations are needed. We call $w_{\mathcal{B}}$ the width of the integral cycle basis \mathcal{B} .

When forcing the periodicity as we do with constraints (3.3f), the computation times of the associated MIP problem depend on the width of the integral cycle basis, see (Liebchen, 2003; Wünsch, 2008); the smaller the width of the integral cycle basis the better. Therefore, we would like the integral cycle basis of the constraint graph G to include all the cycles associated with the circuital constraints (3.3g)–(3.3h); for such a cycle \mathcal{C} we know the integral value of $\mathbf{z}_{\mathcal{C}}$, which is one, and, therefore, such a cycle has a width of zero. We elaborate on how to find an integral cycle basis that includes these zero-width cycles. We build this integral cycle basis from a strictly fundamental cycle basis, which can be obtained from a spanning forest of the constraint graph G . To define a spanning forest, we first need to introduce the definition of a spanning tree:

Definition 3.6 (Spanning tree). *A spanning tree of a graph $G = (V, A)$ is defined as a subset $\mathcal{T} \subseteq A$ such that the graph $G = (V, \mathcal{T})$ contains no cycles and has one connected component.*

Note that such a spanning tree is defined for an undirected graph as well as for a directed graph. Furthermore, note that a graph that consists of multiple connected components has no spanning tree. However, this graph then has a spanning forest:

Definition 3.7 (Spanning forest). *Consider a graph G with $\nu(G) \geq 1$ connected components. Let \mathcal{T}_i be a spanning tree of connected component $i = 1, \dots, \nu(G)$. Then $\mathcal{F} = \bigcup_{i=1}^{\nu(G)} \mathcal{T}_i$ is a spanning forest of graph G .*

Adding only one arc to a spanning forest will create a cycle (circuit); such a cycle is called a *fundamental cycle* and it forms the basis of a *strictly fundamental cycle basis (SFCB)*. A strictly fundamental cycle basis is a special kind of integral cycle basis, see (Kavitha et al., 2009).

Definition 3.8 (Strictly fundamental cycle basis (SFCB)). *The set of cycles $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ is a strictly fundamental cycle basis whenever \mathcal{B} is the set of all the fundamental cycles associated with some spanning forest $\mathcal{F} \subseteq A$. In other words $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ is a*

strictly fundamental cycle basis whenever some spanning forest $\mathcal{F} \subseteq A$ exists such that $\mathcal{B} = \{\mathcal{C}_{\mathcal{F}}(a) \mid a \in A \setminus \mathcal{F}\}$, where $\mathcal{C}_{\mathcal{F}}(a)$ is the unique circuit in $\mathcal{F} \cup \{a\}$.

In this thesis we assume without loss of generality that the circuit $\mathcal{C}_{\mathcal{F}}(a)$ uses the arc a in the forward direction. We can find the spanning forest \mathcal{F} of the constraint graph G from a spanning forest of a smaller (undirected) graph $G' = (V', A')$. The graph G' has one vertex for each signal group and one (undirected) arc for each conflict $\{i, j\} \in \Psi_{\mathcal{S}}$:

$$\begin{aligned} V' &:= \mathcal{S} \\ A' &:= \Psi_{\mathcal{S}}. \end{aligned} \tag{3.5}$$

We call the graph G' a *conflict graph*. We can obtain a spanning forest of the constraint graph G from a spanning forest of the smaller conflict graph $G' = (V', A')$ as follows. Let \mathcal{F}' be a spanning forest of the conflict graph G' . Then the following set of arcs \mathcal{F} is a spanning forest of the constraint graph G , see also Figure 3.5:

$$\mathcal{F} := \{(\textcircled{i}, \textcircled{j}) \mid \{i, j\} \in \mathcal{F}', i < j\} \cup A_g \tag{3.6}$$

This spanning forest \mathcal{F} includes all the arcs that represent effective green intervals, and it includes an arc that represents a clearance interval for each arc in \mathcal{F}' . From spanning forest \mathcal{F} we can obtain a strictly fundamental cycle basis. This SFCB includes all the cycles associated with circuit constraints (3.3h). It does however not include all the cycles associated with circuit constraints (3.3g); this zero-width cycle is only included for each pair of conflicting signal groups $\{i, j\} \in \Psi_{\mathcal{S}}$ for which $\{i, j\} \in \mathcal{F}'$. The following lemma states however that we can use this SFCB to construct an integral cycle basis that does include all these zero-width cycles; the resulting integral cycle basis has replaced some of the cycles in the SFCB by cycles with a width of zero. As a consequence, this integral cycle basis has a smaller width than the SFCB from which it is built.

Lemma 3.1. *Let \mathcal{F}' be a spanning forest of the conflict graph G' and let \mathcal{F} be the spanning forest of the constraint graph G calculated with (3.6). Define $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ to be the SFCB of the constraint graph G that is defined by spanning forest \mathcal{F} , and let \mathcal{B}' be the set of cycles obtained from \mathcal{B} when, for each conflict $\{i, j\} \notin \mathcal{F}'$, $i < j$, we replace the cycle $\mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i}))$ by the cycle:*

$$\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \textcircled{i}), (\textcircled{i}, \textcircled{j}), (\textcircled{j}, \textcircled{j}), (\textcircled{j}, \textcircled{i})\}.$$

The set \mathcal{B}' is an integral cycle basis of the constraint graph G that includes all the cycles associated with circuit constraints (3.3g)–(3.3h).

Proof. See Appendix C.5.1. □

Thus, using Lemma 3.1 we can obtain an integral cycle basis of the constraint graph G that includes all cycles associated with circuit constraints (3.3g)–(3.3h). To this end, we need a spanning forest \mathcal{F}' of the conflict graph G' ; we calculate this spanning forest

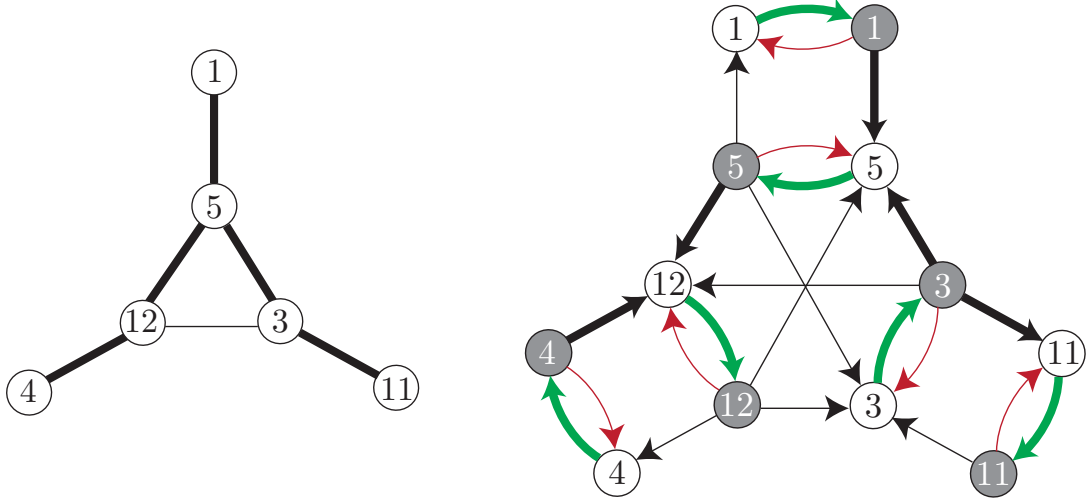


Figure 3.5: Visualization of the spanning forest \mathcal{F} of the constraint graph G (right) obtained from a spanning forest \mathcal{F}' of the conflict graph G' (left) for the intersection in Figure 3.1a; the arcs in the spanning forest are visualized in bold.

with the algorithm given in (Amaldi et al., 2004). This algorithm requires a weight for each arc; we take all arc weights to be the same.

For the constraint graph G associated with the intersection in Figure 3.1a, the resulting integral cycle basis consists of thirteen cycles. Six of these cycles correspond to circuitual constraints (3.3g) and six of them correspond to circuitual constraints (3.3h). The last cycle is the following one:

$$\mathcal{C} = \mathcal{C}^+ = \{((5), (5)), ((5), (3)), ((3), (3)), ((3), (12)), ((12), (12)), ((12), (5))\}. \quad (3.7)$$

The value of $\mathbf{z}_{\mathcal{C}}$ is only unknown for the latter cycle. Therefore, the optimization problem has only one integral-valued design variable for this example.

Remark 3.5. The integral cycle basis constructed with Lemma 3.1 includes each cycle associated with circuitual constraints (3.3g)–(3.3h). Therefore, circuitual constraints (3.3f) associated with these $|\Psi_{\mathcal{S}}| + |\mathcal{S}|$ cycles are made redundant by circuitual constraints (3.3g)–(3.3h). Hence, only $d - (|\Psi_{\mathcal{S}}| + |\mathcal{S}|) = |\Psi_{\mathcal{S}}| - |\mathcal{S}| + \nu(G)$ integral-valued design variables $\mathbf{z}_{\mathcal{C}}$ remain.

Well-posedness

For the T-junction of Figure 3.1 it holds that the minimum clearance times $\underline{c}_{i,j}$, $\{i, j\} \in \Psi_{\mathcal{S}}$ are non-negative; in Appendix C.1 we motivate the use of negative clearance times. In case that all minimum clearance times are non-negative, all real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$ are restricted to the interval $[0, 1)$. The inclusion $\gamma(\textcircled{i}, \textcircled{i}) \in [0, 1)$, $i \in \mathcal{S}$, follows from circuitual constraint (3.3h) combined with the non-negativity of each effective green time (3.3b) and the positivity of each effective red time (3.3c). The inclusion $\gamma(\textcircled{i}, \textcircled{i}) \in [0, 1)$, $i \in \mathcal{S}$, follows from circuitual constraint (3.3h) combined with stabil-

ity constraint (3.3d) and the positivity of each effective red time (3.3c). The inclusion $\gamma(\textcircled{i}, \textcircled{j}) \in [0, 1)$, $\{i, j\} \in \Psi_{\mathcal{S}}$ follows from circuital constraint (3.3g) combined with stability constraint (3.3d) and the non-negativity of the clearance times (3.3e). As a result, $\gamma(\varepsilon_1, \varepsilon_2)$ is the time (expressed as a fraction of the period duration) between an occurrence of the event ε_1 and the next occurrence of the event ε_2 .

From the inclusion $\gamma(\varepsilon_1, \varepsilon_2) \in [0, 1)$ it follows that the variable $\gamma(\varepsilon_1, \varepsilon_2)$ and its associated integer $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously, i.e., for each signal group diagram the value of the real-valued design variable $\gamma(\varepsilon_1, \varepsilon_2)$ and its integer $z(\varepsilon_1, \varepsilon_2)$ are defined uniquely; we can find these unique values by obtaining $f(\varepsilon_1)$ and $f(\varepsilon_2)$ and determining the integral value for $z(\varepsilon_1, \varepsilon_2)$ such that $f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2) \in [0, 1)$; the corresponding value for $f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)$ gives $\gamma(\varepsilon_1, \varepsilon_2)$.

Remark 3.6. *Whenever a design variable $\gamma(\varepsilon_1, \varepsilon_2)$ and its integer $z(\varepsilon_1, \varepsilon_2)$ would be defined ambiguously, possibly multiple solutions to MIP problem (3.3) are associated with the same signal group diagram. This is an undesired property; if the optimization problem is solved with a branch and bound technique, multiple nodes in the search tree may be related to the same solution, which might result in a large search tree and, as a consequence, larger computation times. Furthermore, when the variable $\gamma(\varepsilon_1, \varepsilon_2)$ is defined unambiguously, possibly its optimized value does not correspond to the correct interval in time. For example, when $f(\textcircled{i}) < f(\textcircled{j})$, the optimized value for $\gamma(\textcircled{i}, \textcircled{j})$ might correspond to the interval $[f(\textcircled{i})T, f(\textcircled{j})T + T]$ instead of the interval $[f(\textcircled{i})T, f(\textcircled{j})T]$. As a consequence, the constraints on the intended interval might not be satisfied, e.g., the minimum clearance time might not be satisfied; the resulting signal group diagram might then be infeasible.*

Obtaining the signal group diagram

The objective function (3.2a) together with linear constraints (3.3) constitute the MIP problem. Solving this MIP problem, which can be done by using a branch and bound strategy, gives us the optimal values for the variables $\mathbf{T}' := 1/T$ and $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$. For the example of Figure 3.1a we want to find the signal group diagram that minimizes the average delay that road users experience at this intersection. This signal group diagram can be found by solving the mixed-integer (convex) programming problem.

From the real-valued design variables γ and \mathbf{T}' we can find the fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ as follows. We can assume w.l.o.g. that the effective green interval of signal group 1 starts at fraction (time) zero, i.e., $f(\textcircled{1}) = 0$. From the definition of $\gamma(\varepsilon_1, \varepsilon_2)$ it follows that event ε_2 occurs at fraction $f(\varepsilon_1) + \gamma(\varepsilon_1, \varepsilon_2)$ and, by periodicity, it then follows that:

$$f(\varepsilon_2) := f(\varepsilon_1) + \gamma(\varepsilon_1, \varepsilon_2) \mod 1.$$

Similarly, we can also obtain the relation:

$$f(\varepsilon_2) := f(\varepsilon_1) - \gamma(\varepsilon_2, \varepsilon_1) \mod 1.$$

Since, the $\gamma(\varepsilon_1, \varepsilon_2)$'s are only known for the arcs $(\varepsilon_1, \varepsilon_2) \in A$ we cannot immediately find all fractions $f(\varepsilon)$ from $f(\textcircled{1})$; see the constraint graph G in Figure 3.3. However, we can obtain $f(\textcircled{1})$ and $f(\textcircled{5})$ from $f(\textcircled{1})$. Subsequently, we can obtain $f(\textcircled{3})$, $f(\textcircled{5})$ and $f(\textcircled{12})$ from $f(\textcircled{5})$, et cetera. Since, the graph of Figure 3.3 consists of one component we can obtain all fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ in this way.

Remark 3.7. *When creating harmonizations between intersections, it is desirable that the signal group diagrams of these intersections have a common period duration. Such a common period duration can be found by optimizing the signal group diagrams of these intersections simultaneously. In that case the constraint graph G consists of multiple connected components; each such component corresponds to one intersection. We can then obtain all fractions $f(\varepsilon)$ by setting one fraction to zero for each component. We can do so, because when the events ε_1 and ε_2 are in different connected components, then, by definition, the fraction $\gamma(\varepsilon_1, \varepsilon_2)$ is not restricted by any constraints. Hence, we can set the signal group diagrams of the different connected components independently.*

3.2.2 The generic MIP problem

In this section we give the MIP problem for the generic case. Let $\gamma \in \mathbb{R}^{|A|}$ be a column vector containing the variables $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$. Furthermore, let z be a column vector containing the integral-valued design variables z_C . The MIP problem is as follows:

$$\underset{\mathbf{T}', \gamma, z}{\text{minimize}} \quad J(\mathbf{T}', \gamma, z), \quad (3.8a)$$

subject to:

$$0 \leq 1/\overline{T} \leq \mathbf{T}' \leq 1/\underline{T} \quad (3.8b)$$

$$0 \leq \underline{g}_i \mathbf{T}' \leq \gamma(\textcircled{i}, \textcircled{i}) \leq \overline{g}_i \mathbf{T}', \quad i \in \mathcal{S}, \quad (3.8c)$$

$$0 < \underline{r}_i \mathbf{T}' \leq \gamma(\textcircled{i}, \textcircled{i}) \leq \overline{r}_i \mathbf{T}', \quad i \in \mathcal{S}, \quad (3.8d)$$

$$0 < \max_{q \in \mathcal{Q}_i} \rho_q \leq \gamma(\textcircled{i}, \textcircled{i}), \quad i \in \mathcal{S}, \quad (3.8e)$$

$$\underline{c}_{i,j} \mathbf{T}' \leq \gamma(\textcircled{i}, \textcircled{j}), \quad \{i, j\} \in \Psi_{\mathcal{S}}, \quad (3.8f)$$

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2) = z_C, \quad \forall \mathcal{C} \in \mathcal{B} \quad (3.8g)$$

$$\gamma(\textcircled{i}, \textcircled{i}) + \gamma(\textcircled{i}, \textcircled{j}) + \gamma(\textcircled{j}, \textcircled{j}) + \gamma(\textcircled{j}, \textcircled{i}) = 1, \quad \{i, j\} \in \Psi_{\mathcal{S}} \quad (3.8h)$$

$$\gamma(\textcircled{i}, \textcircled{i}) + \gamma(\textcircled{i}, \textcircled{i}) = 1, \quad i \in \mathcal{S} \quad (3.8i)$$

$$\gamma(\textcircled{i}, \textcircled{i}) + \gamma(\textcircled{i}, \textcircled{j}) \geq \epsilon \mathbf{T}', \quad \{i, j\} \in \Psi_{\mathcal{S}}. \quad (3.8j)$$

where ϵ is some (small) prefixed value strictly greater than zero, \mathcal{B} is some integral cycle basis of the constraint graph G , and J is the objective function; we elaborate on different objective functions later in this section. Constraints (3.8b)–(3.8i) correspond to the constraints (3.3a)–(3.3h). Constraints (3.8j) are needed to obtain a well-posed problem

formulation. These constraints are redundant whenever the minimum clearance times $\underline{c}_{i,j}$ are non-negative, which was the case for the example considered in the previous section. We elaborate on this constraint later in this section. First, we give a quick recap of all constraints that have already been introduced in the previous section. Constraint (3.8b) restricts the period duration T to the correct interval $[\underline{T}, \overline{T}]$. Constraints (3.8c) (constraints (3.8d)) gives a lower and an upper bound on each effective green (effective red) time. Constraints (3.8e) guarantee stability of each signal group. Constraints (3.8f) ensure that the minimum clearance times between conflicting signal groups are satisfied. Constraints (3.8g)–(3.8i) are the cycle periodicity constraints; these constraints model the periodicity of the signal group diagram. As previously mentioned, (3.8g) should be satisfied for each cycle \mathcal{C} in the constraint graph G ; however, as we have shown in Section 3.3, we only have to include this constraint for a subset of the cycles: an integral cycle basis of the constraint graph G . Constraints (3.8j) need some more explanation. These constraints make sure that a clearance time $\gamma(\textcircled{i}, \textcircled{j})T$, $\{i, j\} \in \Psi_S$ is restricted to the interval $(-T, T)$. To obtain this interval, first combine the constraint (3.8j) with the circuitual constraint (3.8h) to obtain $0 < \gamma(\textcircled{i}, \textcircled{i}) + \gamma(\textcircled{i}, \textcircled{j}) < 1$. Furthermore, combine the positivity of the effective green time (3.8e) with the positivity of the effective red time (3.8d) and the circuitual constraint (3.8i) which gives $0 < \gamma(\textcircled{i}, \textcircled{i}) < 1$. The inequalities $0 < \gamma(\textcircled{i}, \textcircled{i}) + \gamma(\textcircled{i}, \textcircled{j}) < 1$ and $0 < \gamma(\textcircled{i}, \textcircled{i}) < 1$ together imply that $\gamma(\textcircled{i}, \textcircled{j})$ is restricted to the interval $(-1, 1)$ and therefore that $\gamma(\textcircled{i}, \textcircled{j})T$ is restricted to the interval $(-T, T)$; a clearance time $\gamma(\textcircled{i}, \textcircled{j})T$ then, as desired, refers to the time between an occurrence of the event \textcircled{i} and the next or the previous occurrence of the event \textcircled{j} depending on the sign of $\gamma(\textcircled{i}, \textcircled{j})$. Note that the MIP problem (3.8) permits negative clearance times. However, if a clearance time $\gamma(\textcircled{i}, \textcircled{j})T$ is negative, then (3.8j) permits the effective green interval of signal group j to start only after the effective green interval of signal group i has started.

Well-posedness

What remains to be proved is that each real-valued design variable $\gamma(\varepsilon_1, \varepsilon_2)$ and its associated integer $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously; recall that $\gamma(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)$. As already stated, it holds that $\gamma(\textcircled{i}, \textcircled{i}) \in (0, 1)$, which implies that $\gamma(\textcircled{i}, \textcircled{i})$ and its corresponding integer $z(\textcircled{i}, \textcircled{i})$ are defined unambiguously. Similarly, we have $\gamma(\textcircled{i}, \textcircled{i}) \in (0, 1)$, which implies that $\gamma(\textcircled{i}, \textcircled{i})$ and its corresponding integer $z(\textcircled{i}, \textcircled{i})$ are also defined unambiguously; this inclusion follows from the positivity of the effective green time (3.8e), the positivity of the effective red time (3.8d) and circuitual constraint (3.8i). Furthermore, from the already proved inequality $0 < \gamma(\textcircled{i}, \textcircled{i}) + \gamma(\textcircled{i}, \textcircled{j}) < 1$ it follows that $\gamma(\textcircled{i}, \textcircled{j}) \in (-\gamma(\textcircled{i}, \textcircled{i}), 1 - \gamma(\textcircled{i}, \textcircled{i}))$; since $\gamma(\textcircled{i}, \textcircled{i})$ is defined unambiguously, this inclusion implies that $\gamma(\textcircled{i}, \textcircled{j})$ and its integer $z(\textcircled{i}, \textcircled{j})$ are both defined unambiguously as well.

Remark 3.8. Note that constraint (3.8j) is not needed if $\underline{c}_{i,j} \geq 0$; it then holds that

$\gamma(\textcircled{i}, \textcircled{j}) + \gamma(\textcircled{j}, \textcircled{i}) \geq \max_{q \in \mathcal{Q}_i} \rho_q > 0$. Hence, we did not need the constraints (3.8j) in the previous section.

Remark 3.9. In Appendix C.3 we have motivated that each pair of conflicting signal groups $\{i, j\} \in \Psi_S$ satisfies the following strict inequality:

$$\underline{c}_{i,j} + \underline{c}_{j,i} > 0.$$

When we combine this inequality with circuital constraint (3.8h) and the non-negativity of a green time (3.8c), we obtain the inclusion $\gamma(\textcircled{i}, \textcircled{j}) \in [\underline{c}_{i,j}\mathbf{T}', 1 - \underline{c}_{i,j}\mathbf{T}']$. This inclusion implies the unambiguous definition of $\gamma(\textcircled{i}, \textcircled{j})$ and its integer $z(\textcircled{i}, \textcircled{j})$. Therefore, we do not necessarily need the constraints (3.8j) to have an unambiguous definition of the variables $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$ and their associated integers $z(\varepsilon_1, \varepsilon_2)$. We however include these constraints for the following reasons. First, this constraint restricts the value of $\gamma(\textcircled{i}, \textcircled{j})T$, $\{i, j\} \in \Psi_S$ to the interval $(-T, T)$. The clearance time $\gamma(\textcircled{i}, \textcircled{j})T$ then, as desired, refers to the time between an occurrence of the event \textcircled{i} and the next or the previous occurrence of the event \textcircled{j} depending on the sign of $\gamma(\textcircled{i}, \textcircled{j})$. The inclusion $\gamma(\textcircled{i}, \textcircled{j})T \in (-T, T)$ might not be satisfied if we do not include the constraints (3.8j). Second, including these constraints leads to more consistency throughout this thesis as we may in some proofs require the strict positivity of a green time plus a subsequent clearance time. For example, we need this strict positivity to relate the integral-valued design variables of our formulation to the binary-valued design variables of the currently existing group-based approaches; we give this relation in Section 3.3.

Objective functions

Objective functions that are commonly used when optimizing signal group diagrams are: minimizing the period duration of the signal group diagram, maximizing the capacity of the intersection, and minimizing the average delay that road users experience (due to traffic light control) at the intersection.

Minimizing the period duration is equivalent to minimizing $-\mathbf{T}'$. Therefore, the objective function is linear in the design variables when minimizing the period duration of the signal group diagram; the resulting problem is a mixed-integer linear programming (MILP) problem.

When maximizing the capacity of the intersection, we multiply the left-hand sides of stability constraints (3.8e) ($\max_{q \in \mathcal{Q}_i} \rho_q$) by a common scaling factor β , which we call the *growth factor*. The objective is to maximize β (minimize $-\beta$), i.e., the objective is to find the maximum growth factor β^{\max} of the arrival rates λ_q , $q \in \mathcal{Q}$ for which we can find a solution to (3.8b)–(3.8j). A maximum value β^{\max} that is less than one, indicates that the intersection is overloaded by $100(1 - \beta^{\max})$ percent. A maximum value β^{\max} that exceeds one, indicates that the intersection has $100(\beta^{\max} - 1)$ percent of reserve capacity. Note that, when maximizing the capacity of the intersection, the constraints remain linear

and the objective function is linear as well. Therefore, the resulting problem is a MILP problem.

The third objective is to minimize the average weighted delay that road users experience at the intersection. This average weighted delay equals $D = \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q$, where d_q is the (approximated) delay that road users experience at queue q and w_q is the weight factor of this queue. Assume that d_q is convex in the real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$ and \mathbf{T}' . The average weighted delay D is then also convex in these real-valued design variables. The function d_q is convex for the approximations from (Miller, 1963; van den Broek et al., 2006; Webster, 1958). The resulting problem is then a mixed-integer convex programming problem.

Remark 3.10. *When maximizing the capacity of the intersection, we apply a homogeneous growth factor β to all arrival rates, i.e., all arrival rates are scaled with the same growth factor. It is, however, also possible to have heterogeneous growth factors. With these heterogeneous growth factors, it is for example possible to analyse the situation for which the demand is expected to increase drastically for some signal groups, while this demand is expected to increase only moderately for other signal groups.*

To apply these heterogeneous growth factors, we use a 'global' growth factor β . Furthermore, for each signal group $i \in \mathcal{S}$ we require an input parameter β_i . This input parameter β_i indicates how much the traffic at signal group $i \in \mathcal{S}$ increases with respect to the global growth factor, i.e., the arrival rates of the queues $q \in \mathcal{Q}_i$ are scaled with a factor $\beta_i \beta$. For example, the value $\beta_i = 0.5$ indicates that the growth factor at each queue $q \in \mathcal{Q}_i$ is only half that of the global growth factor β . We can then replace the constraints (3.3d) by:

$$\max_{q \in \mathcal{Q}_i} \rho_q + (\beta - 1) \beta_i \max_{q \in \mathcal{Q}_i} \rho_q \leq \gamma(\textcircled{i}, \textcircled{i}). \quad (3.9)$$

Note that when $\beta_i = 1$ for all signal groups, then we have the original MILP problem formulation that applies a homogeneous growth factor β to all arrival rates.

Note that it is also possible to have a negative value for β_i ; the amount of traffic arriving at signal group $i \in \mathcal{S}$ then decreases as the global growth factor β decreases. We elaborate on some technicalities that arise when $\beta_i < 0$ for some signal groups. When $\beta_i \geq 0$ for all signal groups, each signal group diagram that can handle a growth factor of β^{\max} can also handle any growth factor $\beta \in [0, \beta^{\max}]$; a growth factor $\beta < \beta^{\max}$ is less restrictive than a growth factor of β^{\max} . However, this is no longer the case when $\beta_i < 0$ for some signal groups; the arrival rates at the queues $q \in \mathcal{Q}_i$ then increase when β decreases. Furthermore, when $\beta_i < 0$, the left hand-side of (3.9) may become negative, which indicates that the arrival rates at the queues $q \in \mathcal{Q}_i$ are negative. However, this is not a problem as (3.9) is then simply redundant; a negative left-hand side of (3.9) would have the same effect as a left-hand side that equals zero, which would indicate that the arrival rate at each queue $q \in \mathcal{Q}_i$ is zero.

3.3 Comparison with existing optimization formulations

The currently existing group-based approaches (Cantarella and Improta, 1988; Improta and Cantarella, 1984; Sacco, 2014; Wong and Heydecker, 2011; Wong and Wong, 2003; Wong, 1996; Yan et al., 2014) are essentially modelled in the same way. To compare these group-based approaches to the new approach of Section 3.2, we first consider the small example of Figure 3.1. For this intersection, the optimal signal group diagram is the one in Figure 3.2. In this section we relate the integral variables z_c of the novel approach to the binary variables of the existing approaches. At the end of this section we give an elaborate numerical comparison of the currently existing group-based approaches and the approach proposed in this chapter.

3.3.1 Currently existing group-based approaches

To model the structure of a signal group diagram, the currently existing group-based approaches introduce one binary-valued design variable $\Omega_{i,j}$ for each pair of conflicting signal groups $\{i, j\} \in \Psi_S$. Thus, the number of integral-valued design variables is equal to the number of conflicts at the intersection, which is $|\Psi_S|$; for the example in Figure 3.1 we have 6 binary-valued design variables: $\Omega_{1,5}$, $\Omega_{3,5}$, $\Omega_{3,11}$, $\Omega_{3,12}$, $\Omega_{4,12}$ and $\Omega_{5,12}$. The binary variable $\Omega_{i,j}$ is based on the fractions $f(\textcircled{i})$ and $f(\textcircled{j})$: $\Omega_{i,j} = 0$ if $f(\textcircled{i}) \leq f(\textcircled{j})$ and $\Omega_{i,j} = 1$ otherwise; the fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ are visualized in bold as these are real-valued design variables for the group-based approaches. We can prove that $f(\textcircled{i}) \neq f(\textcircled{j})$, which implies the equality $\Omega_{j,i} = 1 - \Omega_{i,j}$, which is the reason that we only need one binary variable per conflict. We can obtain the inequality $f(\textcircled{i}) \neq f(\textcircled{j})$ for each conflict $\{i, j\} \in \Psi_S$ by combining well-posedness constraint (3.8j) with circuitual constraint (3.8h).

The binary-valued design variables $\Omega_{i,j}$ are not uniquely defined for each signal group diagram. For example, with the signal group diagram in Figure 3.2 we can associate four different values for the binary-valued design variables $\Omega_{i,j}$, see Table 3.4. If we may shift the signal group diagram in time, e.g., let the effective green time of signal group 3 start at time zero, then even more such combinations would exist. Of course, we can assume w.l.o.g. that $f(\textcircled{1}) = 0$ and, as a result, only two combinations remain for the example. In general, this ambiguity then only occurs when $f(\textcircled{i}) = 0$ (or $f(\textcircled{i}) = 1$) for some signal group $i \in \mathcal{S} \setminus \{1\}$; for such a signal group it does not matter if its switch to effective green is scheduled at fraction zero or at fraction one, i.e., the signal group diagram with $\Omega_{i,j} = 0$ for all conflicting signal groups j is then the same as the signal group diagram with $\Omega_{i,j} = 1$ for all conflicting signal groups j . However, if some fraction $f(\textcircled{i})$ is close to zero or close to one, these two signal group diagrams are not the same, yet they may be very similar and therefore have a very similar value for the objective function. Since the group-based approaches are usually solved with a branch and bound method, this is not a desirable property; multiple nodes in the search tree may be related to very similar

solutions in terms of the real-valued design variables and the objective value, which may result in a large search tree, and as a consequence, large computation times.

	$f(\textcircled{1})$	$f(\textcircled{3})$	$f(\textcircled{4})$	$f(\textcircled{5})$	$f(\textcircled{11})$	$f(\textcircled{12})$	$\Omega_{1,5}$	$\Omega_{3,5}$	$\Omega_{3,11}$	$\Omega_{3,12}$	$\Omega_{4,12}$	$\Omega_{5,12}$
1	0	0	$\frac{3835}{9487}$	$\frac{3635}{9487}$	$\frac{2243}{9487}$	$\frac{2243}{9487}$	0	0	0	0	1	1
2	1	0	$\frac{3835}{9487}$	$\frac{3635}{9487}$	$\frac{2243}{9487}$	$\frac{2243}{9487}$	1	0	0	0	1	1
3	0	1	$\frac{3835}{9487}$	$\frac{3635}{9487}$	$\frac{2243}{9487}$	$\frac{2243}{9487}$	0	1	1	1	1	1
4	1	1	$\frac{3835}{9487}$	$\frac{3635}{9487}$	$\frac{2243}{9487}$	$\frac{2243}{9487}$	1	1	1	1	1	1

Table 3.4: Four different combinations of the binary-valued design variables $\Omega_{i,j}$ that can be associated with the signal group diagram in Figure 3.2

Remark 3.11. Not all cited group-based approaches use the real-valued design variables $f(\textcircled{i}) \in [0, 1]$, $i \in \mathcal{S}$ and the binary-valued design variables $\Omega_{i,j}$, $\{i, j\} \in \Psi_{\mathcal{S}}$, $i < j$ (in combination with the durations of the effective green intervals) directly. However, by applying a coordinate transformation, each of these approaches can be written in these design variables (amongst other design variables) or in the design variables of the symmetric variant. Instead of the variables $f(\textcircled{i})$, $i \in \mathcal{S}$, this symmetric variant uses the variables $f(\textcircled{i}) \in [0, 1]$, $i \in \mathcal{S}$; for this symmetric variant the integral-valued design variable $\Omega_{i,j}$ equals zero whenever $f(\textcircled{i}) \leq f(\textcircled{j})$ and it equals one otherwise.

3.3.2 Novel approach

Where the currently existing group-based approaches need 6 binary variables for the example of Figure 3.1, the MIP problem of Section 3.2 requires only one integral-valued design variable. The integral-valued design variables are needed to formulate the cycle periodicity constraints (3.8g). As mentioned before, in general, a cycle basis of a graph G with vertices V and arcs A consists of $d = |A| - |V| + \nu(G)$ cycles, where $\nu(G)$ is the number of connected components of the graph G . Hence, the integral cycle basis required to formulate the cycle periodicity constraints (3.8g) consists of $2|\Psi_{\mathcal{S}}| + \nu(G)$ cycles. However, $|\Psi_{\mathcal{S}}| + |\mathcal{S}|$ of these cycles are zero-width cycles, which are the cycles associated with circutal constraints (3.8h)–(3.8i); for these cycles we know the integral multiplicity $z_{\mathcal{C}}$. Hence, $|\Psi_{\mathcal{S}}| - |\mathcal{S}| + \nu(G)$ integral-valued design variables remain. For the example in Figure 3.1a only one integral-valued design variable $z_{\mathcal{C}}$ with unknown value remains, which is the integral variable $z_{\mathcal{C}}$ associated with cycle (3.7); this is the cycle:

$$\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{5}, \textcircled{5}), (\textcircled{5}, \textcircled{3}), (\textcircled{3}, \textcircled{3}), (\textcircled{3}, \textcircled{12}), (\textcircled{12}, \textcircled{12}), (\textcircled{12}, \textcircled{5})\}.$$

Since the number of connected components $\nu(G)$ cannot exceed the number of signal groups $|\mathcal{S}|$, the number of integral-valued design variables $z_{\mathcal{C}}$ of the proposed method, which is $|\Psi_{\mathcal{S}}| - |\mathcal{S}| + \nu(G)$, does not exceed the number of binary-valued design variables $\Omega_{i,j}$ of the currently existing group-based approaches, which is $|\Psi_{\mathcal{S}}|$.

3.3.3 Relating the two approaches

The integral-valued design variable z_C is related to the binary-valued design variables $\Omega_{i,j}$ as follows:

Lemma 3.2. *Consider a cycle basis $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ obtained by using Lemma 3.1. Consider a cycle $\mathcal{C} \in \mathcal{B}$ that does not equal the zero-width cycle associated with the circuital constraint (3.8i). For each such cycle, we can relate its integral-valued design variable z_C to the binary design variables $\Omega_{i,j}$ as follows:*

$$z_C = \sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}^+} \Omega_{i,j} - \sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}^-} \Omega_{i,j}.$$

Proof. See Appendix C.5.2. □

For the example of Figure 3.1, we have only one cycle for which the value of z_C is not known before optimization; this is the cycle (3.7). Applying Lemma 3.2 to this cycle gives: $z_C = \Omega_{5,3} + \Omega_{3,12} + \Omega_{12,5} = 2 - \Omega_{3,5} + \Omega_{3,12} - \Omega_{5,12}$. From Table 3.4 we can see that all four combinations of $\Omega_{i,j}$ are captured by $z_C = 1$.

Remark 3.12. *The currently existing group-based approaches and the novel approach given in this chapter show resemblance with the periodic event scheduling problem (PESP). The currently existing group-based approaches have much in common with the original formulation of the PESP, see for example (Serafini and Ukovich, 1989a), whereas the novel formulation resembles the cycle periodicity formulation of the PESP, see for example (Nachtigall, 1994).*

Also some differences exist between the optimization problem proposed in this chapter and the cycle periodicity formulation of the PESP. First, our constraint graph $G = (V, A)$ contains some cycles \mathcal{C} for which the integral-valued design variable z_C must be fixed. Second, we do not assume the period duration T to be fixed. Third, the formulation proposed in this chapter has additional constraints, e.g., stability constraints (3.8e) and well-posedness constraints (3.8j). Fourth, the objective functions that we consider cannot be written as the weighted sum assumed by the cycle periodicity formulation of the PESP.

3.3.4 Numerical comparison of the two approaches

In this section we perform an extensive numerical comparison of the currently existing group-based approaches with the novel approach proposed in this chapter. To this end, we use a total of thirteen real-life examples, which correspond to intersections that are positioned in the Netherlands. All relevant information about these intersections can be found in (Fleuren and Lefebvre, 2016a). We have grouped these intersections according to their sizes, see also Table 3.5. The first five intersections (S1, S2, S3, S4 and S5) concern T-junctions with only six signal groups and are considered to be small. Intersections M1,

M2, M3 and M4 concern intersections with 9-15 signal groups and are considered to have a medium size. For the last 4 intersections (L1, L2, L3 and L4) the number of signal groups is between 27 and 29, which we consider to be large.

For each of the thirteen intersections we consider twelve different optimization problems, which are depicted in Table 3.5. For six of these optimization problems we minimize the period duration; for each of these optimization problems we scale the arrival rates differently. We consider the scalings: 1.00, 1.05, 1.10, 1.15, 1.20 and 1.25. Furthermore, one of these twelve optimization problems maximizes the capacity of the intersection, i.e., maximizes the growth factor β with which the arrival rates can be scaled; this optimization problem can be obtained from MIP problem (3.8) by multiplying the left-hand side of stability constraint (3.8e) with β and maximizing β . The remaining five optimization problems minimize the average weighted delay that road users experience at the intersection; for each of these optimization problems we fix the period duration and minimize the average delay at that period duration. We consider 5 different period durations that are all scalings of the minimum period duration; we obtain this minimum period duration via optimization. We consider the scalings: 1.1, 1.2, 1.3, 1.4 and 1.5. To formulate these mixed-integer programming problems we use the delay equation of Van den Broek (3.2b). We approximate this delay with a piecewise linear approximation. In the upcoming section we elaborate on such piecewise linear approximations.

Piecewise linear approximation

Suppose that we have an optimization problem with as goal to minimize the objective function $J(\mathbf{x})$, which depends on a single variable \mathbf{x} , subject to linear constraints. Let this objective function $J(\mathbf{x})$ be non-linear function. This section elaborates on how to approximate $J(\mathbf{x})$ with a piecewise linear function by using only linear constraints; the result is a linear optimization problem. First, we consider the generic case: $J(\mathbf{x})$ is some, possibly non-convex, continuous function of the variable \mathbf{x} . Subsequently, we consider the case that $J(\mathbf{x})$ is a convex function. Finally, we use these piecewise linear approximations to approximate the average (weighted) delay that road users experience at the intersection.

Generic function Consider some continuous function $J(\mathbf{x})$ of a single variable \mathbf{x} that is restricted to the interval $[x_0, x_m]$. The function $J(\mathbf{x})$ can be approximated by a piecewise linear approximation as follows, see also (Lin et al., 2013). Divide the interval $[x_0, x_m]$ into m smaller intervals: $[x_0, x_1], [x_1, x_2], \dots, [x_{m-1}, x_m]$, where $x_0 < x_1 < \dots < x_m$. In each of these intervals we approximate the function $J(\mathbf{x})$ with a linear function, see also Figure 3.6; the values x_0, x_1, \dots, x_m are also referred to as the break points of the piecewise linear approximation of $J(\mathbf{x})$. In the interval $[x_k, x_{k+1}]$, $k = 0, \dots, m-1$ we approximate the function $J(\mathbf{x})$ by the linear function:

$$f_k(\mathbf{x}) := J(x_k) + s_k(\mathbf{x} - x_k),$$

where,

$$s_k := \frac{J(x_{k+1}) - J(x_k)}{x_{k+1} - x_k}.$$

To this end, we use the binary-valued design variables \mathbf{b}_k , $k = 0, 1, \dots, m-1$. These binary-valued design variables are related via the following constraint:

$$\sum_{k=1}^{m-1} \mathbf{b}_k = 1,$$

The following constraint restricts \mathbf{x} to the interval $[x_k, x_{k+1}]$ whenever $\mathbf{b}_k = 1$:

$$x_k - L(1 - \mathbf{b}_k) \leq \mathbf{x} \leq x_{k+1} + L(1 - \mathbf{b}_k), \quad k = 0, 1, \dots, m-1,$$

where L is some large positive number. Let \mathbf{J}^{pwl} be a real-valued design variable. We use this real-valued design variable to model the piecewise linear approximation of $J(\mathbf{x})$ by using constraints that are linear in \mathbf{x} ; the value of \mathbf{J}^{pwl} depends on \mathbf{x} and equals the value $f_k(\mathbf{x})$ when $\mathbf{x} \in [x_k, x_{k+1}]$. We force the equality $\mathbf{J}^{\text{pwl}} = f_k(\mathbf{x})$ when $\mathbf{x} \in [x_k, x_{k+1}]$, $k = 0, \dots, m-1$ by including the following linear constraints:

$$-L(1 - \mathbf{b}_k) \leq \mathbf{J}^{\text{pwl}} - f_k(\mathbf{x}) \leq L(1 - \mathbf{b}_k), \quad k = 0, 1, \dots, m-1,$$

where L is some large positive number. Note that by substituting the definition $f_k(\mathbf{x}) := J(x_k) + s_k(\mathbf{x} - x_k)$, the above constraints indeed become linear in the real-valued design variables \mathbf{J}^{pwl} and \mathbf{x} . The goal of the resulting optimization problem is to minimize the value of \mathbf{J}^{pwl}

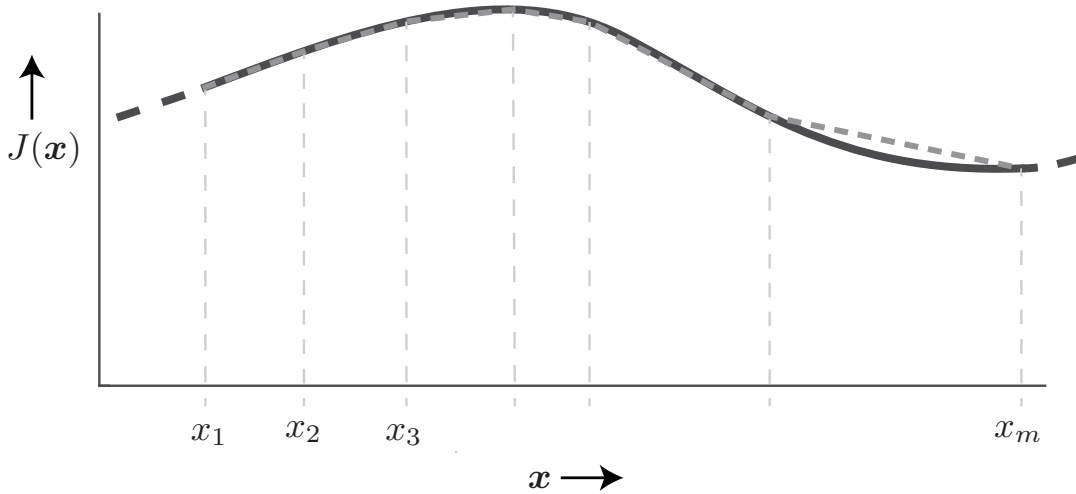


Figure 3.6: Piecewise linear approximation (dashed gray) of a function (black).

Convex function In the generic case we need the additional binary-valued design variables \mathbf{b}_k , $k = 0, 1, \dots, m-1$. These additional binary variables are not needed whenever $J(\mathbf{x})$ is a continuous and convex function. The value for \mathbf{J}^{pwl} then satisfies:

$$\mathbf{J}^{\text{pwl}} := \max_{k=0,1,\dots,m-1} f_k(\mathbf{x}), \quad (3.10)$$

see also Figure 3.7. As we would like to minimize the value of \mathbf{J}^{pwl} it suffices to include only the following linear constraints:

$$\mathbf{J}^{\text{pwl}} \geq f_k(\mathbf{x}), \quad k = 0, \dots, m-1.$$

Note that for this alternative formulation it is not necessary that \mathbf{x} is restricted to the interval $[x_0, x_m]$; this was necessary in the generic case.

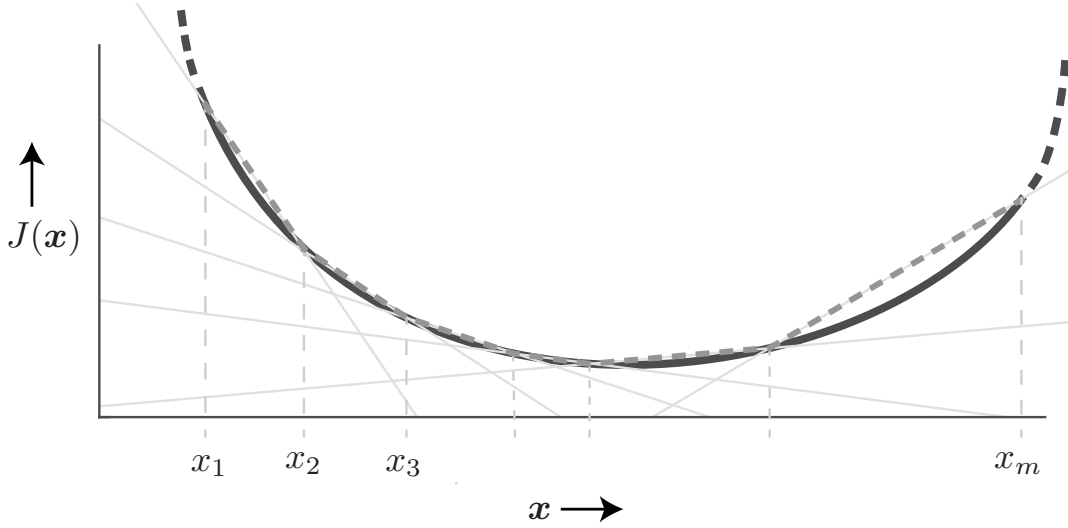


Figure 3.7: Piecewise linear approximation (dashed gray) of a convex function (black). The light gray lines give the functions $f_k(\mathbf{x}) := J(x_k) + s_k(\mathbf{x} - x_k)$, $k = 0, \dots, m-1$.

Piecewise linear approximation of the delay The average weighted delay D can be written as:

$$D = \sum_{i \in \mathcal{S}} d_i,$$

where,

$$d_i = \sum_{q \in \mathcal{Q}_i} w_q d_q.$$

We approximate the delay d_q that road users experience at queue q with the equation of Van den Broek (3.2b). As we fix the period duration T , the term d_i is a convex function of only the effective red fraction $\gamma(\textcircled{i}, \textcircled{i})$. We can formulate this piecewise

linear approximation as was shown in the previous section. Note that the effective red time $\gamma(\textcircled{i}, \textcircled{i})T$ is included in the following interval:

$$\left[\lfloor \max\{\underline{r}_i, T - \bar{g}_i\} \rfloor, \lceil \min\{\bar{r}_i, T - \underline{g}_i\} \rceil \right] \cup \left[0, T - \max_{q \in \mathcal{Q}_i} \rho_q T \right). \quad (3.11)$$

To approximate $d_i(\gamma(\textcircled{i}, \textcircled{i}))$ piecewise linearly we define the breakpoints $x_0, x_0 + 1, x_0 + 2, \dots, x_m$ to correspond to the integral effective red times $\gamma(\textcircled{i}, \textcircled{i})T$ that are included in the above interval. Let $\mathbf{d}_i^{\text{pw1}}$ be the real-valued design variable that is used to approximate the function d_i piecewise linearly. The goal is to minimize $\sum_{i \in \mathcal{S}} \mathbf{d}_i^{\text{pw1}}$.

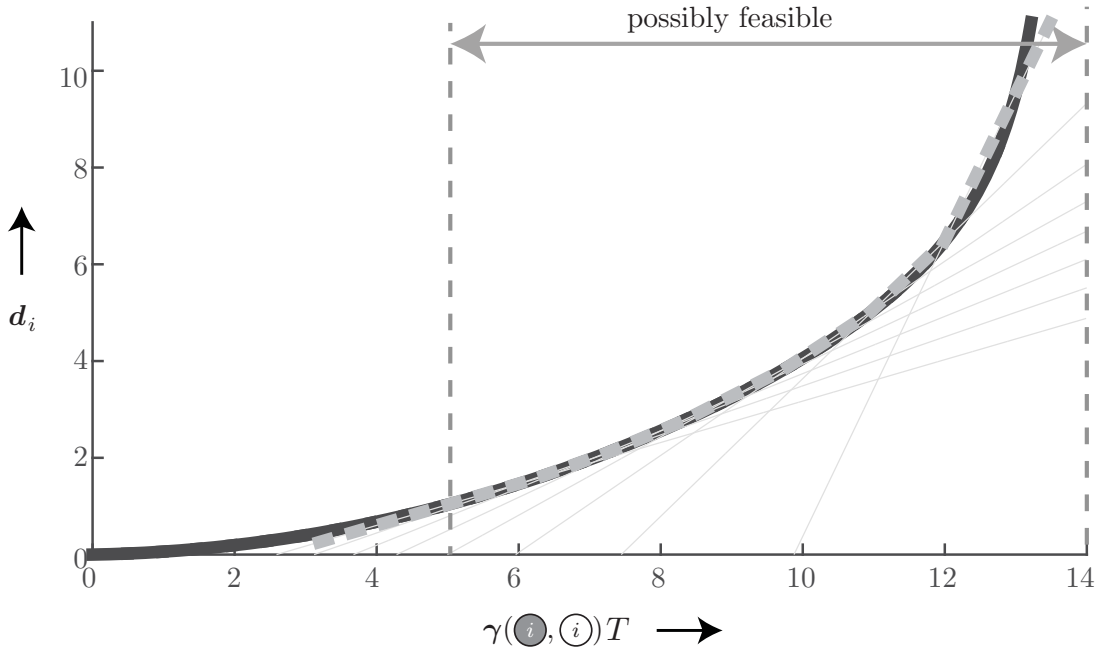


Figure 3.8: Piecewise linear approximation (dashed grey) of the delay equation \mathbf{d}_i of Van den Broek (3.2b) (black) when $\mathcal{Q}_i = \{1\}$, $\rho_i = 0.3$, $w_i = 1$, $\mu_i = 0.5$, $T = 20$, $\underline{g}_i = 5$, $\underline{r}_i = 5$, and $\bar{g}_i = \bar{r}_i = \infty$ seconds. The interval of possibly feasible values for the effective red time is then $[5, 14)$. The piecewise linear function has the following break points: 5, 6, 7, ..., 13. The light gray lines give the functions $f_k(\mathbf{x}) := J(x_k) + s_k(\mathbf{x} - x_k)$, $k = 0, \dots, m - 1$.

Numerical results

In Table 3.5 we give the results (objective values) of all 13×12 optimization problems and in Table 3.6 we give the sizes of these optimization problems. In Table 3.7 we give the computation times for three different solvers, which are two of the (currently) fastest commercial solvers and one of the (currently) fastest non-commercial solvers: CPLEX version 12.6.1.0 (International Business Machines Corp, 2015), GUROBI version 6.0.5 (Gurobi Optimization, Inc., 2015) and SCIP version 3.2.0 (Achterberg, 2009). The computation times are subject to fluctuations. These fluctuations are caused by background processes that are running on the computer and they are typically in the order of 10 percent. To obtain a good estimate for these computation times, we have solved each optimization problem at least 100 times. All results in this thesis are obtained on a computer with

specifications: Intel i5-4300U CPU @1.90GHZ with 16.0GB of RAM. From Table 3.7 we conclude that the novel approach performs better for all three objective functions. The difference in computation times is the smallest when maximizing the capacity of the intersection, and it is the largest when minimizing the delay that road users experience at the intersection. The computation times are especially large when using SCIP to minimize the delay at large intersections. The improvement is also large for these test cases; the ratio of the computation times is then 0.055, which implies a reduction in computation times of $(1 - 0.055)100 = 94.5$ percent and a speed up of a factor $1/0.055 \approx 18$. Note that for some of these test cases the computation time is very short and therefore, for these cases, the difference in computation time is not noticeable in practice. However, for some cases this difference is very much noticeable, e.g., when minimizing the delay at large intersections.

Recall that a signal group diagram can be used to obtain a phase diagram, which forms the basis of many (semi)actuated controllers. Such a phase diagram can be derived from a pre-timed controller, e.g., with the method from Appendix A.2, and is usually based on forecasted (or historical) arrival rates. Fast optimization of pre-timed control may also allow us to compute such a phase diagram and an associated actuated controller in an online manner, i.e., depending on the current traffic situation. We briefly touch upon such online algorithms in Section 8.2 of this thesis.

Remark 3.13. *In Table 3.7 we have given aggregated computation times. To obtain these computation times, we have computed the geometric average computation time over the different test cases instead of the (arithmetic) average computation time. The geometric average of the numbers x_1, x_2, \dots, x_N is defined as:*

$$\sqrt[N]{x_1 x_2 \dots x_N}.$$

We motivate the usage of this geometric average with an example. Consider a test set of two test cases: test case 1 and test case 2. For this test set, we compare two methods: method 1 and method 2. The average computation times are as follows:

	Test case 1	Test case 2
Method 1	9 minutes	1 minute
Method 2	3 minutes	3 minutes

If we define the speedup ratio as the computation time of method 1 divided by the computation time of method 2, we obtain the speedup ratios: 3 and $1/3$. If we take the average of this, we obtain $1^{2/3}$, which would suggest that method 1 is on average ~ 67 percent slower than method 2. However, if we define the speedup ratio the other way around, i.e., as the ratio of the computation time of method 2 to the computation time of method 1, we also obtain an average of $1^{2/3}$. This would suggest that method 2 is on average ~ 67 percent slower than method 1. Therefore, we can skew the results either in favor of

Objective	min T					max β		min D					
	Scaling	1.00	1.05	1.10	1.15	1.20	1.25	-	1.1	1.2	1.3	1.4	1.5
S1		35.86	39.32	43.51	48.71	55.32	64.00	1.39	22.71	17.13	15.16	14.20	13.75
S2		30.00	30.00	30.00	30.00	30.84	34.30	1.56	14.98	14.61	14.38	14.31	14.50
S3		39.11	40.72	43.06	45.69	48.66	52.05	1.66	47.78	31.72	26.58	24.55	23.62
S4		65.75	69.00	73.02	76.77	81.21	86.20	1.48	37.60	27.26	24.94	24.33	24.45
S5		64.14	67.53	71.30	75.52	82.95	92.41	1.35	44.33	34.09	31.80	30.91	30.89
M1		43.70	45.56	47.69	50.03	56.26	67.90	1.35	34.17	25.57	23.70	22.79	22.26
M2		45.55	48.08	52.24	57.18	63.16	70.53	1.36	41.67	29.73	26.70	25.55	25.18
M3		141.05	167.40	205.86	∞	∞	∞	1.11	86.37	64.43	58.19	56.62	56.74
M4		80.01	85.81	92.53	100.38	109.69	120.91	1.34	47.71	36.82	33.24	31.95	31.71
L1		74.71	78.10	81.81	86.37	121.75	∞	1.22	40.48	38.12	38.56	39.46	40.58
L2		83.59	89.83	97.07	105.58	115.73	∞	1.23	61.97	50.15	48.51	48.61	49.60
L3		74.57	80.35	84.62	92.26	103.76	118.53	1.25	37.61	33.05	32.33	32.67	33.46
L4		71.55	74.67	80.50	88.97	99.44	112.71	1.27	32.61	29.14	28.66	29.07	29.86

Table 3.5: The objective value of each of the 13×12 test cases. The first column indicates which intersection is considered and the first row indicates which objective function is considered: minimizing the period duration (min T), maximizing the capacity (max β) or minimizing the average delay (min D). In case that the average delay is minimized, the period duration is fixed to some scaling (> 1) of the minimum period duration; the second row indicates how much the minimum period duration, which can be found in the second column, is scaled. In case that the period duration is minimized, the second row indicates the scaling of the arrival rates that is used. This table contains infinite values; these infinite values indicate that the corresponding MILP problems are infeasible.

$\min T$						$\max \beta$						$\min D$					
# Variables			# Constraints			# Variables			# Constraints			# Variables			# Constraints		
Int.	Cont.	Ineq.	Int.	Eq.	Ineq.	Int.	Cont.	Ineq.	Int.	Eq.	Ineq.	Int.	Cont.	Ineq.	Int.	Eq.	Ineq.
S1	1 (6)	25 (13)	50 (52)	13 (0)	1 (6)	1 (6)	26 (14)	50 (52)	13 (0)	1 (6)	1 (6)	30 (18)	183-246 (185-248)	13 (0)			
S2	1 (6)	25 (13)	50 (52)	13 (0)	1 (6)	1 (6)	26 (14)	50 (52)	13 (0)	1 (6)	1 (6)	30 (18)	163-222 (165-224)	13 (0)			
S3	1 (6)	25 (13)	50 (52)	13 (0)	1 (6)	1 (6)	26 (14)	50 (52)	13 (0)	1 (6)	1 (6)	30 (18)	235-316 (237-318)	13 (0)			
S4	4 (9)	31 (13)	56 (58)	19 (0)	4 (9)	4 (9)	32 (14)	56 (58)	19 (0)	4 (9)	4 (9)	36 (18)	393-530 (395-532)	19 (0)			
S5	4 (9)	31 (13)	56 (58)	19 (0)	4 (9)	4 (9)	32 (14)	56 (58)	19 (0)	4 (9)	4 (9)	36 (18)	380-514 (382-516)	19 (0)			
M1	10 (18)	55 (19)	92 (94)	37 (0)	10 (18)	10 (18)	56 (20)	92 (94)	37 (0)	10 (18)	10 (18)	63 (27)	392-526 (394-528)	37 (0)			
M2	10 (18)	55 (19)	92 (94)	37 (0)	10 (18)	10 (18)	56 (20)	92 (94)	37 (0)	10 (18)	10 (18)	63 (27)	403-544 (405-546)	37 (0)			
M3	22 (35)	99 (29)	156 (158)	71 (0)	22 (35)	22 (35)	100 (30)	156 (158)	71 (0)	22 (35)	22 (35)	112 (42)	1994-2697 (1996-2699)	71 (0)			
M4	22 (35)	99 (29)	156 (158)	71 (0)	22 (35)	22 (35)	100 (30)	156 (158)	71 (0)	22 (35)	22 (35)	112 (42)	1185-1597 (1187-1599)	71 (0)			
L1	44 (70)	195 (55)	304 (306)	141 (0)	44 (70)	44 (70)	196 (56)	304 (306)	141 (0)	44 (70)	44 (70)	221 (81)	2181-2936 (2183-2938)	141 (0)			
L2	44 (70)	195 (55)	304 (306)	141 (0)	44 (70)	44 (70)	196 (56)	304 (306)	141 (0)	44 (70)	44 (70)	221 (81)	2406-3255 (2408-3257)	141 (0)			
L3	45 (71)	201 (59)	318 (320)	145 (0)	45 (71)	45 (71)	202 (60)	318 (320)	145 (0)	45 (71)	45 (71)	229 (87)	2308-3085 (2310-3087)	145 (0)			
L4	45 (71)	201 (59)	318 (320)	145 (0)	45 (71)	45 (71)	202 (60)	318 (320)	145 (0)	45 (71)	45 (71)	229 (87)	2195-2955 (2197-2957)	145 (0)			

Table 3.6: The sizes of all the MILP problems in the following format: number of integers (Int.), number of continuous variables (Cont.), number of inequality constraints (Ineq.) and the number of equality constraints (Eq.). All information concerning the currently existing group-based approaches is displayed between brackets; all other values concern the novel approach given in this chapter. When minimizing the delay ($\min D$), the number of inequality constraints that is needed for the piecewise approximation of the delay depends on the period duration at which we want to minimize the delay; the larger the period duration the more inequality constraints are needed. Therefore, we give the smallest and the largest number of inequality constraints that is needed when minimizing the period duration.

Solver	Intersections	$\min T$			$\max \beta$			$\min D$		
		GB (s)	Novel (s)	Novel/GB (s)	GB (s)	Novel (s)	Novel/GB (s)	GB (s)	Novel (s)	Novel/GB (s)
CPLEX	Small	0.087	0.058	0.666	0.081	0.062	0.764	0.099	0.080	0.810
	Medium	0.094	0.078	0.828	0.096	0.094	0.981	0.429	0.188	0.439
	Large	0.370	0.203	0.549	0.276	0.202	0.733	8.296	1.313	0.158
GUROBI	Small	0.005	0.004	0.895	0.005	0.004	0.850	0.017	0.011	0.636
	Medium	0.021	0.018	0.842	0.015	0.022	1.535	0.314	0.102	0.325
	Large	0.699	0.203	0.290	0.375	0.215	0.574	7.822	1.231	0.157
SCIP	Small	0.026	0.020	0.771	0.037	0.022	0.599	0.074	0.042	0.566
	Medium	0.116	0.064	0.556	0.120	0.106	0.883	1.373	0.234	0.170
	Large	1.038	0.568	0.547	0.693	0.659	0.951	54.624	2.997	0.055
Column average		-	-	0.630	-	-	0.839	-	-	0.277

Table 3.7: The (geometric) average computation times (in seconds) needed by the currently existing group-based approaches (GB) and the novel approach that is proposed in this chapter (Novel). We distinguish between three types of optimization problems ($\min T$, $\max \beta$ and $\min D$), three types of intersections (small, medium and large), and three types of solvers (CPLEX 12.6.1.0, GUROBI 6.0.5. and SCIP 3.2.0). Moreover, we give the (geometric) average for some columns.

method 1 or in favor of method 2 when using arithmetic averages. Furthermore, when using the arithmetic average, it also matters in which order the calculations are performed; first calculating the average computation time and subsequently calculating the speedup ratio from these averages, gives a different answer than first calculating the speed up ratio and then calculating the average of these ratios. The geometric average does not possess these undesirable properties. For the example, the result is then always a speedup ratio of 1 (independent of the definition of the speedup ratio and independent of the order in which the calculations are performed), which indicates that both methods have the same performance.

Remark 3.14. *In this chapter we consider the period duration T to be fixed when minimizing the average delay that road users experience. Question is how to find a good value for the period duration T . We answer this question in Chapter 5. In that chapter we give a heuristic method to find a good period duration T when minimizing the average delay that road users experience at the intersection. This heuristic method iteratively searches for a better period duration. In each iteration of this heuristic method a MILP problem is solved that minimizes the average delay at some fixed period duration.*

3.4 Summary

This chapter presents a novel group-based approach to optimize signal group diagrams for isolated intersections. With this approach, the structure of the signal group diagram and the duration of these green intervals can be optimized simultaneously. We have used the mathematical framework of (Serafini and Ukovich, 1989a) to formulate the linear constraints of this optimization problem. The design variables are both integral-valued and real-valued. The real-valued design variables represent periods in time; the inverse of the period duration is used as a design variable and all other real-valued design variables express the duration of a specific interval in time as a fraction of the period duration. The integral-valued design variables are used to model the periodicity of the signal group diagram.

Possible objective functions of the optimization problem are: minimizing the period duration of the signal group diagram, maximizing the capacity of the intersection, and minimizing the average delay that road users experience at the intersection. The objective functions are linear for the first two problems. For the third problem the objective function may be convex depending on the formula that is used to approximate the average delay that road users experience at a queue under pre-timed control. Examples of approximations that are convex are the ones from (Miller, 1963; van den Broek et al., 2006; Webster, 1958).

For the currently existing group-based approaches, multiple solutions may be associated with the same signal group diagram; all these solutions have different values for the binary-valued design variables. The formulation proposed in this chapter, which ag-

gregates the binary variables of the currently existing group-based approaches, does not possess this undesirable property. With an extensive computational study we have compared the currently existing group-based approaches with the novel proposed method; the results show superiority of the novel approach.

In the next chapter we extend this framework to also optimize over the number of realizations of each signal group, i.e., in the next chapter we also optimize the number of effective green intervals that each of the signal groups receive during a period; the number of realizations was assumed to be one in this chapter.

Optimizing pre-timed control: variable number of realizations

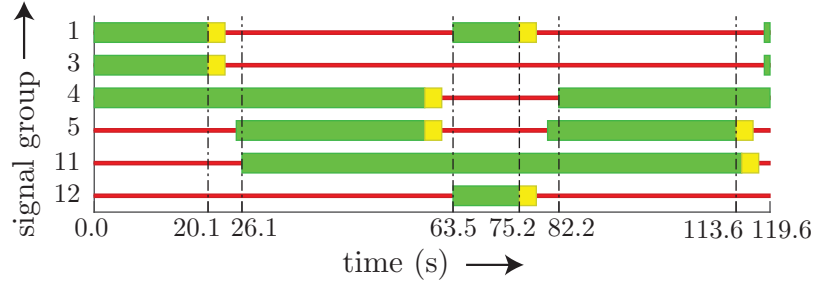
4.1 Introduction

In this chapter we extend the mixed-integer programming problem proposed in the previous chapter to also optimize the number of realizations (distinct green intervals) of each traffic light. To our knowledge, this is the first mixed-integer programming problem allowing traffic lights to have multiple realizations and, therefore, also the first one considering the number of realizations of each traffic light to be a design variable.

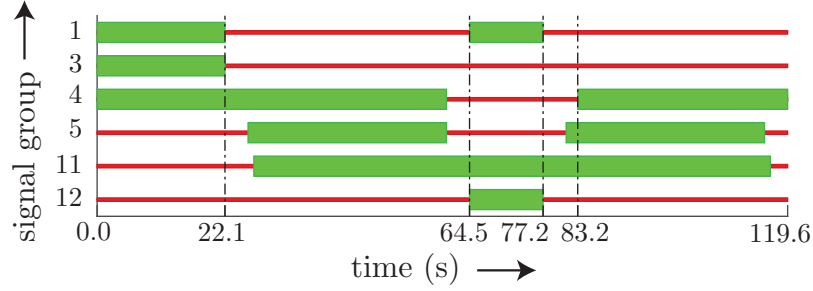
To show the effect of multiple realizations, we consider again the example introduced in Section 3.2.1. In that section we considered the minimization of the (average) delay that road users experience at the T-junction introduced in the same section. The result was the optimal signal group diagram given in Figure 3.1b. For this signal group diagram road users experience an average (approximated) delay of 26.416 seconds. When we allow each signal group to have an additional realization, we can obtain the signal group diagram in Figure 4.1; this signal group diagram can be obtained by solving a mixed-integer (convex) programming problem and its switching times are given in Table 4.1. For this signal group diagram, signal groups 1 and 5 have two realizations and the average delay that road users experience is 25.106 seconds; this is an improvement of ~ 5 percent. For the intersections from (Fleuren and Lefebber, 2016a), the average delay that road users experience can also substantially decrease by allowing several signal groups to have multiple realizations, see also Section 4.4; for some intersections we have seen a decrease of more than 10 percent.

This chapter has the following structure. In Section 4.2, we fix the number of realizations of each signal group. In that section we formulate a mixed-integer programming problem to find the optimal signal group diagram; in contrast to Chapter 3, this num-

This chapter is based on the following paper: Fleuren, S. and Lefebber, E. (2016d). Optimizing fixed-time control at isolated intersections: Part II: Optimizing the number of green intervals. Technical Report DC 2016.068, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven, The Netherlands. Available at http://mn.wtb.tue.nl/~lefeber/do_download_pdf.php?id=171. Submitted to Transportation Science



(a) Signal group diagram that visualizes the indications.



(b) Signal group diagram that visualizes the effective green and effective red modes.

Figure 4.1: A signal group diagram for the T-junction introduced in Section 3.2.1 for which signal groups 1 and 5 have two realizations and all other signal groups have one realization. The average delay that road users experience for this signal group diagram is 25.106 seconds and the period duration is 119.58 seconds.

Signal group (i)	Realization (k)	$t_{i,k}^G$	$t_{i,k}^Y$	$t_{i,k}^R$	$t_{i,k}^g$	$t_{i,k}^r$
1	1	118.58	20.14	23.14	0	22.14
	2	63.49	75.23	78.23	64.49	77.23
3	1	118.58	20.14	23.14	0	22.14
4	1	82.23	58.49	61.49	83.23	60.49
5	1	25.14	58.49	61.49	26.14	60.49
	2	80.23	113.58	116.58	81.23	115.58
11	1	26.14	114.58	117.58	27.14	116.58
12	1	63.49	75.23	78.23	64.49	77.23

Table 4.1: The times (rounded to hundreds of a second) at which each signal group $i = 1, 3, 4, 5, 11, 12$ switches to green ($t_{i,k}^G$), yellow ($t_{i,k}^Y$), red ($t_{i,k}^R$), effective green ($t_{i,k}^g$), and effective red ($t_{i,k}^r$) for the signal group diagram in Figure 4.1.

ber of realizations is allowed to exceed one. Subsequently, in Section 4.3 we adjust this formulation to also optimize the number of realizations that each signal group has. In Section 4.4 we perform a numerical study and in Section 4.5 we give a summary.

4.2 Fixed number of realizations

In this section we consider the number of realizations of each signal group to be fixed. To formulate the corresponding optimization problem we require the number of realizations K_i of signal group $i \in \mathcal{S}$. Furthermore, we require the same inputs as those required in the previous chapter, see Section 3.2.1. In this chapter we use the following notation. We

define $\mathcal{K}_i := \{1, \dots, K_i\}$ as the set of realizations of signal group $i \in \mathcal{S}$. Furthermore, we define Ψ_R as the set of conflicting realizations:

$$\Psi_R := \{\{(i, k), (j, k')\} \mid \{i, j\} \in \Psi_S, k \in \mathcal{K}_i, k' \in \mathcal{K}_j\}.$$

For each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ we have to satisfy minimum clearance times.

4.2.1 Real-valued design variables

Let the realizations of each signal group $i \in \mathcal{S}$ be numbered according to the periodic order in which they occur, i.e., the realizations of signal group $i \in \mathcal{S}$ are scheduled in the periodically repeating order $1, 2, \dots, K_i$. Let \textcircled{i}_k (\textcircled{i}_k) denote the start (end) of realization $k \in \mathcal{K}_i$ of signal group $i \in \mathcal{S}$. For ease of notation we define $\textcircled{i}_0 := \textcircled{i}_{K_i}$. We use the following terminology in this thesis. We refer to the interval between the event \textcircled{i}_k and the event \textcircled{i}_k as *effective green interval* k of signal group $i \in \mathcal{S}$; we may also refer to this interval as *realization* k of signal group i . Similarly we refer to the interval between the event \textcircled{i}_{k-1} and the event \textcircled{i}_k as *effective red interval* k of signal group $i \in \mathcal{S}$; note that effective red interval k of signal group i precedes effective green interval k of signal group i . Define the set of periodic events \mathcal{E} as follows:

$$\mathcal{E} = \{\textcircled{i}_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\} \cup \{\textcircled{i}_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}.$$

We use the same definition for the fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ and the fractions $\gamma(\varepsilon_1, \varepsilon_2)$ as used in Chapter 3; fraction $f(\varepsilon)$ denotes the time (as a fraction of the period duration) at which event $\varepsilon \in \mathcal{E}$ is scheduled and the fraction $\gamma(\varepsilon_1, \varepsilon_2)$ denotes the time (expressed as a fraction of the period duration) between an occurrence of periodic event ε_1 and (the previous or the next occurrence of) periodic event ε_2 , i.e.,

$$\gamma(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)$$

for some integral value $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. In Table 4.2 we give the values of $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ for the signal group diagram in Figure 4.1. The real-valued design variables of the optimization problem are the variables $\gamma(\varepsilon, \varepsilon)$ that are subject to some safety constraint together with the reciprocal of the period duration $\mathbf{T}' := 1/T$; later in this section we prove that each real-valued design variable $\gamma(\varepsilon_1, \varepsilon_2)$ and its associated integer $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously. From the real-valued design variables we can obtain the switching times $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ with the method given in Section 3.2.1.

4.2.2 Linear constraints

When the number of realizations of each signal group is fixed, the optimization problem is very similar to the one from Chapter 3. The period duration T is bounded from below

Signal group (i)	Realization (k)	$f(\textcircled{i}_k)$	$f(\textcircled{i}_k)$
1	1	0	2214
		11958	11958
		6449	7723
3	1	0	2214
		11958	11958
		8323	6049
4	1	11958	11958
		2614	6049
		11958	11958
5	1	11958	11958
		8123	11558
		11958	11958
11	1	2714	11658
		11958	11958
		6449	7723
12	1	11958	11958
		6449	7723
		11958	11958

Table 4.2: The fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ associated with the signal group diagram in Figure 4.1.

by \underline{T} and from above by \overline{T} . Thus, the reciprocal \mathbf{T}' must satisfy the following constraint:

$$1/\overline{T} \leq \mathbf{T}' \leq 1/\underline{T}. \quad (4.1a)$$

Each effective green time of signal group i is bounded from below and from above:

$$0 \leq \underline{g}_i \mathbf{T}' \leq \gamma(\textcircled{i}_k, \textcircled{i}_k) \leq \overline{g}_i \mathbf{T}', \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i. \quad (4.1b)$$

Also each effective red time of signal group i is bounded from below and from above:

$$0 < \underline{r}_i \mathbf{T}' \leq \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) \leq \overline{r}_i \mathbf{T}'. \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i, \quad (4.1c)$$

Recall that $\textcircled{i}_0 := \textcircled{i}_{K_i}$, which we use in the above constraint. A signal group controls the access to the intersection for the queues $q \in \mathcal{Q}_i$. Queue $q \in \mathcal{Q}$ must be effective green for at least a fraction $\rho_q := \lambda_q/\mu_q$ of the period duration as, otherwise, its queue length would grow indefinitely. To ensure stability for each queue $q \in \mathcal{Q}_i$, signal group i must be effective green for at least a fraction $\rho_i^{\text{SG}} := \max_{q \in \mathcal{Q}_i} \rho_q > 0$ of the period duration:

$$0 < \rho_i^{\text{SG}} \leq \sum_{k \in \mathcal{K}_i} \gamma(\textcircled{i}_k, \textcircled{i}_k), \quad i \in \mathcal{S}. \quad (4.1d)$$

Minimum clearance times have to be satisfied for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$:

$$\underline{c}_{i,j} \mathbf{T}' \leq \gamma(\textcircled{i}_k, \textcircled{j}_{k'}), \quad \{(i, k), (j, k')\} \in \Psi_R. \quad (4.1e)$$

We allow such a minimum clearance time $\underline{c}_{i,j}$ to be negative; in Appendix C.1 the use of such negative clearance times is motivated. To have a well-posed optimization problem we restrict the duration of a negative clearance time:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) \geq \epsilon \mathbf{T}', \quad \{(i, k), (j, k')\} \in \Psi_R, \quad (4.1f)$$

which restricts effective green interval k of signal group i plus the clearance time to effective green interval k' of signal group j to be at least $\epsilon > 0$ seconds. This constraint ensures the inclusion $\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) \in (-1, 1)$. A clearance time $\gamma(\textcircled{i}_k, \textcircled{j}_{k'})T$ then, as desired, refers to the time between an occurrence of the event \textcircled{i}_k and the next or the previous occurrence of the event $\textcircled{j}_{k'}$ depending on the sign of $\gamma(\textcircled{i}_k, \textcircled{j}_{k'})$. Moreover, the constraints (4.1f) ensure that each variable $\gamma(\textcircled{i}_k, \textcircled{i}_k)$ and its associated integer $z(\textcircled{i}_k, \textcircled{i}_k)$ are defined unambiguously. We prove both statements in Section 4.2.2.

The following constraints reduce the symmetry of the proposed mixed-integer programming problem; for each signal group we assume w.l.o.g. that its first effective red time is the largest:

$$\gamma(\textcircled{i}_{K_i}, \textcircled{i}_1) \geq \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k), \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i \setminus \{1\}. \quad (4.1g)$$

This constraint reduces the solution space and reduces the symmetry of the MIP problem. Therefore, including this last constraint is expected to reduce the computation time needed to solve the optimization problem.

Circuitual constraints

The variables $\gamma(\varepsilon_1, \varepsilon_2)$ are also related via cycle periodicity constraints. For more information on these cycle periodicity constraints than is given in this section, we refer to Section 3.3. The cycle periodicity constraints model the periodicity of the signal group diagram. To formulate these constraints we have to extend the definition of the constraint graph $G = (V, A)$ to allow signal groups to have multiple realizations:

$$V = \{\textcircled{i}_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\} \cup \{\textcircled{i}_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\},$$

$$A = A_g \cup A_r \cup A_c,$$

where,

$$A_g := \{(\textcircled{i}_k, \textcircled{i}_k) \mid i \in \mathcal{S}, k \in \mathcal{K}_i\},$$

$$A_r := \{(\textcircled{i}_{k-1}, \textcircled{i}_k) \mid i \in \mathcal{S}, k \in \mathcal{K}_i\},$$

$$A_c := \{(\textcircled{i}_k, \textcircled{j}_{k'}) \mid \{(i, k), (j, k')\} \in \Psi_R\}.$$

The set of vertices V equals the set of events \mathcal{E} and, therefore, each vertex represents either a switch to effective green or a switch to effective red. With each vertex $\varepsilon \in V$ we can associate the fraction $f(\varepsilon)$. Furthermore, constraint graph G has a directed arc $(\varepsilon_1, \varepsilon_2) \in A$ for each of the real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$; the arcs in A_g represent effective green intervals, the arcs in A_r represent effective red intervals, and the arcs in A_c represent clearance intervals. See Figure 4.2 for the constraint graph of the T-junction introduced in Section 3.2.1 when $K_1 = K_5 = 2$ and $K_3 = K_4 = K_{11} = K_{12} = 1$.

Consider a cycle \mathcal{C} in this constraint graph G . From the periodicity of a signal group diagram it follows that the following cycle periodicity constraint should be satisfied, see

also Section 3.3:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2) = z_{\mathcal{C}}, \quad (4.1h)$$

where $z_{\mathcal{C}} \in \mathbb{Z}$. Fortunately, it suffices to formulate this constraint only for the cycles in some integral cycle basis of the constraint graph G ; this constraint is then automatically satisfied for all the cycles in the constraint graph G .

For some cycles in the constraint graph G we must fix the value of $z_{\mathcal{C}}$. Each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ must occur within the same period. Therefore, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ it holds that:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) + \gamma(\textcircled{j}_{k'}, \textcircled{j}_{k'}) + \gamma(\textcircled{j}_{k'}, \textcircled{i}_k) = 1, \quad (4.1i)$$

which implies that each period consists of realization k of signal group i , a clearance interval from this realization to realization k' of signal group j , the k' th realization of signal group j itself, and a clearance interval back to realization k of signal group i . Furthermore, the effective green intervals of signal group $i \in \mathcal{S}$ together with the effective red intervals of signal group i constitute one period, which implies the following circuital constraint:

$$\sum_{k \in \mathcal{K}_i} (\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k)) = 1, \quad i \in \mathcal{S}. \quad (4.1j)$$

The circuital constraints (4.1h)–(4.1j) model the periodicity of the signal group diagram. In the next section we show how to find an integral cycle basis of the constraint graph; the method that we use to find an integral cycle basis extends the method proposed in the previous chapter.

Obtaining an integral cycle basis of the constraint graph

To formulate the linear constraints (4.1) we require an integral cycle basis of the constraint graph G ; this integral cycle basis is needed to formulate the cycle periodicity constraints (4.1h). We find this integral cycle basis similar to the approach used in Chapter 3. The resulting integral cycle basis includes all the cycles associated with circuital constraints (4.1i)–(4.1j), and is constructed from a strictly fundamental cycle basis.

Recall that a strictly fundamental cycle basis (SFCB) is defined by a spanning forest \mathcal{F} . We find the spanning forest \mathcal{F} from a spanning forest \mathcal{F}' of the smaller (undirected) conflict graph $G' = (V', A')$; we bring to mind its definition (3.5):

$$\begin{aligned} V' &:= \mathcal{S}, \\ A' &:= \Psi_{\mathcal{S}}. \end{aligned}$$

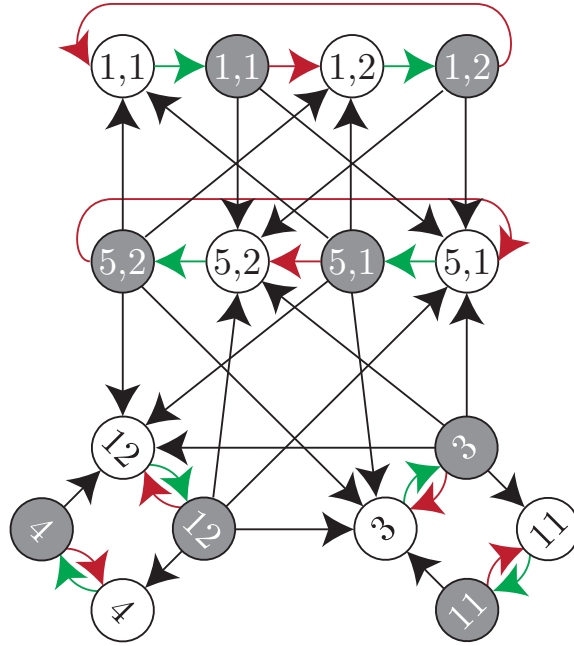


Figure 4.2: The constraint graph $G = (V, A)$ of the intersection introduced in Section 3.2.1 when $K_1 = K_5 = 2$ and $K_3 = K_4 = K_{11} = K_{12} = 1$. The white (grey) vertex with the text i, k denotes the event \odot_k (\odot_k). The white (grey) vertex with the text i denotes the event \odot_1 (\odot_1). The effective green intervals, effective red intervals and clearance intervals are visualized in green, red respectively black.

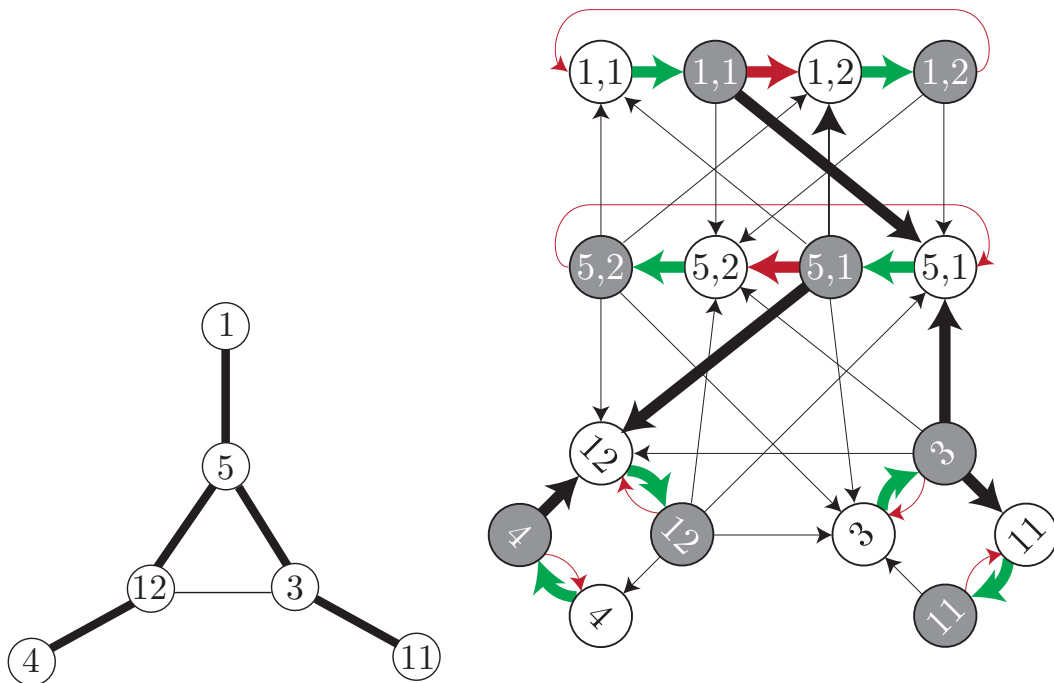


Figure 4.3: Visualization of the spanning forest of the constraint graph G (right) that is obtained from a spanning forest of the conflict graph G' (left); each bold arc is included in the spanning forest. The conflict graph corresponds to the intersection introduced in Section 3.2.1 and the constraint graph corresponds to $K_1 = K_5 = 2$ and $K_3 = K_4 = K_{11} = K_{12} = 1$.

The conflict graph associated with the T-junction introduced in Section 3.2.1 is depicted in Figure 4.3 (left). We can obtain a spanning forest \mathcal{F} of the constraint graph G from a spanning forest \mathcal{F}' of the smaller (undirected) conflict graph $G' = (V', A')$ with the following equation, see also Figure 4.3:

$$\begin{aligned} \mathcal{F} := & \{((\textcircled{i}_k, \textcircled{j}_{k'})) \mid \{i, j\} \in \mathcal{F}', i < j\} \cup A_g \\ & \cup \{((\textcircled{i}_{k-1}, \textcircled{i}_k)) \mid i \in \mathcal{S}, k \in \mathcal{K}_i \setminus \{1\}\}. \end{aligned} \quad (4.2)$$

Thus, the spanning forest \mathcal{F} includes all the arcs that represent effective green intervals, for each signal group $i \in \mathcal{S}$ it includes $K_i - 1$ of the K_i arcs that represent red intervals, and it includes an arc that represents a clearance interval for each arc in \mathcal{F}' . From spanning forest \mathcal{F} we can obtain a strictly fundamental cycle basis. This SFCB does not necessarily contain all the cycles that are associated with circuital constraints (4.1i)–(4.1j); it contains the zero-width cycle associated with the conflict $\{(i, k), (j, k')\} \in \Psi_{\mathcal{S}}, i < j$ if and only if $k = k' = 1$ and $\{i, j\} \in \mathcal{F}'$. We can however use this SFCB to construct an integral cycle basis that does include all these zero-width cycles, see Lemma 4.1; the resulting integral cycle basis has replaced some of the cycles in the SFCB by cycles with a width of zero, i.e., we replace some cycles by cycles for which the integer $z_{\mathcal{C}}$ is fixed to one. This implies that this integral cycle basis has a smaller width than the SFCB from which it is built.

Lemma 4.1. *Let \mathcal{F}' be a spanning forest of the conflict graph G' and let \mathcal{F} be the spanning forest of the constraint graph G calculated with (4.2). Define $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ to be the SFCB of graph G defined by spanning forest \mathcal{F} , and let \mathcal{B}' be the set of cycles obtained from \mathcal{B} when, for each arc $((\textcircled{i}_k, \textcircled{j}_{k'})) \notin \mathcal{F}, i < j$, we replace the cycle $\mathcal{C}_{\mathcal{F}}((\textcircled{j}_{k'}, \textcircled{i}_k))$ by the cycle:*

$$\mathcal{C} = \mathcal{C}^+ = \{((\textcircled{i}_k, \textcircled{i}_k), (\textcircled{i}_k, \textcircled{j}_{k'})), ((\textcircled{i}_k, \textcircled{j}_{k'}), (\textcircled{j}_{k'}, \textcircled{j}_{k'})), ((\textcircled{j}_{k'}, \textcircled{j}_{k'}), (\textcircled{j}_{k'}, \textcircled{i}_k)), ((\textcircled{j}_{k'}, \textcircled{i}_k))\}.$$

The set \mathcal{B}' is an integral cycle basis of the constraint graph G that includes all the cycles associated with circuital constraints (4.1i)–(4.1j).

Proof. See Appendix D.3.1 for the proof. □

With Lemma 4.1 we can construct an integral cycle basis of the constraint graph G . To this end, we require a spanning forest \mathcal{F}' of the conflict graph G' ; we calculate this spanning forest with the algorithm of (Amaldi et al., 2004) and we take all arc weights to be the same.

For the constraint graph G visualized in Figure 4.2, the resulting integral cycle basis consists of 23 cycles. Six of these cycles correspond to circuital constraints (4.1i) and eleven of them correspond to circuital constraints (4.1j). Therefore, six cycles remain for which the value of $z_{\mathcal{C}}$ is unknown (before optimization). Thus, the optimization problem has six integral-valued design variables for this example.

Well-posedness

Each variable $\gamma(\varepsilon_1, \varepsilon_2)$ can be written as $\gamma(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)$. We can prove that each real-valued design variable $\gamma(\varepsilon_1, \varepsilon_2)$ and its corresponding integer $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously with a proof that is similar to the one from Section 3.2.1.

4.2.3 Objective function

In this section we elaborate on three different objective functions: minimizing the period duration T , maximizing the capacity of the intersection, and minimizing the average weighted delay that road users experience at the intersection.

We can minimize the period duration by maximizing its reciprocal \mathbf{T}' . The resulting problem is a mixed-integer linear programming (MILP) problem. When maximizing the capacity of the intersection, we search for the signal group diagram for which the largest increase in the arrival rates λ_q , $q \in \mathcal{Q}$ is sustainable. To this end, we multiply the left-hand sides (ρ_i^{SG}) of stability constraints (4.1d) by a growth factor β . The objective is to maximize this growth factor. The resulting problem is also a MILP problem. Whenever the maximum growth factor β^{\max} is less than one, this implies that the intersection is overloaded by $(1 - \beta^{\max})100$ percent. On the other hand, when this growth factor is greater than one, the intersection has $(\beta^{\max} - 1)100$ percent of overcapacity.

The last objective is to minimize the average weighted delay that road users experience at the intersection:

$$D = \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q,$$

where d_q is the average delay at queue $q \in \mathcal{Q}$ and w_q is the weight factor associated with this queue. We can use the approximations of for example (Miller, 1963; van den Broek et al., 2006; Webster, 1958) to approximate the delay d_q . However, all these approximations assume that a signal group has a single realization, i.e., these formulae assume $K_i = 1$. As no better alternative is available at the time of writing, we extend these approximations, in a straightforward manner, to allow a signal group to have multiple realizations. For the formulae of (Miller, 1963; van den Broek et al., 2006; Webster, 1958) the extended approximation is a convex function of the design variables; in Appendix D.1 we prove this convexity for the approximation of (van den Broek et al., 2006), which is the approximation that we use in this thesis. As a consequence of this convexity, the resulting problem is a mixed-integer convex programming problem when minimizing the average delay that road users experience.

For ease of notation we define $r'_{i,k} := \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k)$ and $r_{i,k} := r'_{i,k}/\mathbf{T}'$. First assume that signal group i receives a single realization, i.e., $K_i = 1$, and consider one of its queues $q \in \mathcal{Q}_i$. The aforementioned approximations for the delay that road users experience at the queue $q \in \mathcal{Q}_i$ can be split into a deterministic part and a stochastic part: $d_q =$

$d_q^{\text{det}} + d_q^{\text{stoch}}$, see Chapter 2. For the approximation of (van den Broek et al., 2006) we have:

$$\begin{aligned}
 d_q^{\text{det}} &:= \frac{r_{i,1}^2}{2T(1-\rho_q)}, \\
 &= \frac{r'_{i,1}{}^2}{2\mathbf{T}'(1-\rho_q)}, \\
 d_q^{\text{stoch}} &= \frac{r_{i,1}}{2\lambda_q(1-\rho_q)T} \left(\frac{\sigma_q^2}{1-\rho_q} + \frac{r_{i,1}\rho_q^2\sigma_q^2T^2}{(1-\rho_q)(T-r_{i,1})^2((1-\rho_q)T-r_{i,1})} \right), \\
 &= \frac{r'_{i,1}}{2\lambda_q(1-\rho_q)\mathbf{T}'} \left(\frac{\sigma_q^2}{1-\rho_q} + \frac{r'_{i,1}\rho_q^2\sigma_q^2\mathbf{T}'^2}{(1-\rho_q)(\mathbf{T}'-r'_{i,1})^2((1-\rho_q)\mathbf{T}'-r'_{i,1})} \right).
 \end{aligned} \tag{4.3}$$

In the following sections we extend this deterministic delay term and this stochastic delay term to also allow $K_i > 1$.

Extending the deterministic delay term The deterministic delay term describes the delay whenever the arrival process and the departure process would be purely deterministic and fluid-like, see also Figure 2.2; the amount of traffic waiting at queue q increases with a rate of λ_q during an effective red interval. During an effective green interval this amount of traffic decreases with a rate of $\mu_q - \lambda_q$ as long as the queue is not emptied. When the queue is emptied, the queue remains empty until the next effective red interval starts. The deterministic delay term (4.3) assumes that the queue is emptied during its effective green interval, i.e., it assumes stability. We can extend this deterministic delay term, in a straightforward manner, to the case of multiple realizations, i.e., to the case $K_i \geq 1$. Consider again the deterministic and fluid-like arrival and departure process. Whenever, for this deterministic system, the queue is emptied during each effective green interval, we find the following expression for the deterministic delay term:

$$\begin{aligned}
 d_q^{\text{det}} &= \sum_{k \in \mathcal{K}_i} \frac{r'_{i,k}{}^2}{2\mathbf{T}'(1-\rho_q)} = \sum_{k \in \mathcal{K}_i} d_{q,k}^{\text{det}}. \\
 d_{q,k}^{\text{det}} &:= \frac{r'_{i,k}{}^2}{2\mathbf{T}'(1-\rho_q)}
 \end{aligned} \tag{4.4}$$

When minimizing the average weighted delay, we can force the queue to be emptied during each effective green interval (for this deterministic arrival and departure process) by adding the following constraints to the mixed-integer programming problem:

$$(1 - \rho_i^{\text{SG}}) \gamma(\textcircled{i}_k, \textcircled{i}_k) \geq \rho_i^{\text{SG}} \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k), \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i. \tag{4.5}$$

We only have to include this constraint for the signal groups $i \in \mathcal{S}$ for which $K_i > 1$; this inequality is already implied by stability constraint (4.1d) for each signal group $i \in \mathcal{S}$ with $K_i = 1$.

Remark 4.1. *Whenever we do not force the queue q to be emptied during each effective green interval for the deterministic and fluid-like queueing process, the deterministic delay term d_q^{det} cannot be written as the sum (4.4); in Appendix D.2 we show that the deterministic delay term d_q^{det} and also the total delay d_q are then no longer convex.*

Extending the stochastic delay term The stochastic delay term d_q^{stoch} corresponds to the stochastic contributions in the delay. For example, when the arrivals are stochastic, the queue might not be empty at the end of an effective green interval. When $K_i = 1$, this stochastic delay term is a function of the fraction $r'_{i,1}$, which is the total red fraction of signal group i , and the (reciprocal of the) period duration \mathbf{T}' . In case that a signal group has multiple realizations, we replace $r'_{i,1}$ by $r'_{i,1} + \dots + r'_{i,K_i}$, which is the total red fraction of signal group i when $K_i \geq 1$.

Remark 4.2. *We do not claim that this straightforward extension of the approximate formulae for the delay results in a good approximation for the case $K_i > 1$; we merely want to show different objective functions that can be considered when signal groups are allowed to receive multiple realizations. However, this extended approximation does have some desirable properties. Consider the extended approximation of (van den Broek et al., 2006). It satisfies the following properties:*

- *Consider a queue $q \in \mathcal{Q}_i$. The delay d_q returned by the extended approximation of (van den Broek et al., 2006) grows to infinity as $g'_i \downarrow \rho_q$, where $g'_i := \sum_{k \in \mathcal{K}_i} \gamma(\textcircled{i}_k, \textcircled{i}_k)$ is the total green fraction of signal group i , i.e., the delay at queue $q \in \mathcal{Q}_i$ grows to infinity as the total green fraction of signal group i approaches the load ρ_q .*
- *Consider the case that signal group i has a single realization, i.e., $K_i = 1$. This extended approximation then reduces to the original approximation of (van den Broek et al., 2006).*
- *Consider the case that signal group i receives multiple realizations, i.e., $K_i > 1$. Let $g_{i,k}$ be the duration of effective green interval k of signal group i . Assume that $g_{i,k} = 0$ for each realizations $k \in \mathcal{K}_i \setminus \{1\}$. From equation (4.5) it then follows that $r_{i,k} = 0$ for each realizations $k \in \mathcal{K}_i \setminus \{1\}$. Therefore, effectively, signal group i has only one effective green interval ($g_{i,1}$) and one effective red interval ($r_{i,1}$). The extended approximation then reduces to the original approximation of Van den Broek (3.2b) with an effective red time of $r_{i,1}$ seconds and a period duration of $g_{i,1} + r_{i,1}$ seconds.*
- *Consider a signal group diagram for which signal group i receives one effective green time of $g_{i,1}$ seconds and one effective red time of $r_{i,1}$ seconds. The period duration of this signal group diagram is $T := g_{i,1} + r_{i,1}$ seconds. Moreover, consider a signal group diagram with a period duration of $K_i T$ seconds for which signal group i has K_i*

realizations. For this second signal group diagram, signal group i alternates between an effective green time of $g_{i,1}$ seconds and an effective red time of $r_{i,1}$ seconds. The signal timings of signal group i are identical for both signal group diagrams and, as a consequence, also the delay that road users experience is identical for both signal group diagrams. For the second signal group diagram we can find the average delay that road users experience at signal group i by applying the extended approximation of (van den Broek et al., 2006) with K_i effective red times of $r_{i,1}$ seconds and a period duration of $K_i T$ seconds. As desired, the result is the same as the delay that road users experience for the first signal group diagram, which can be obtained by using the (original) approximation of (van den Broek et al., 2006) with a single effective red time of $r_{i,1}$ seconds and a period duration of T seconds.

4.3 Variable number of realizations

In the previous section we have formulated an optimization problem that assumes the number of realizations K_i to be fixed for each signal group $i \in \mathcal{S}$. In this section we consider the number of realizations K_i of each signal group $i \in \mathcal{S}$ to be a design variable. To this end, for each signal group $i \in \mathcal{S}$ we require a minimum number of realizations $\underline{K}_i \geq 1$ and a maximum number of realizations \overline{K}_i . We define:

$$\begin{aligned}\underline{\mathcal{K}}_i &:= \{1, \dots, \underline{K}_i\}, \\ \overline{\mathcal{K}}_i &:= \{1, \dots, \overline{K}_i\}, \\ \mathcal{K}_i^d &:= \overline{\mathcal{K}}_i \setminus \underline{\mathcal{K}}_i.\end{aligned}$$

Note that signal group $i \in \mathcal{S}$ is certain to have the realizations $k \in \underline{\mathcal{K}}_i$ and, via optimization, it is decided whether signal group $i \in \mathcal{S}$ has the realization $k \in \mathcal{K}_i^d$. Moreover, in this section we use the following constraint graph $G = (V, A)$:

$$\begin{aligned}V &= \{(\textcircled{i})_k \mid i \in \mathcal{S}, k \in \overline{\mathcal{K}}_i\} \cup \{(\textcircled{i})_k \mid i \in \mathcal{S}, k \in \overline{\mathcal{K}}_i\}, \\ A &= A_g \cup A_r \cup A_c,\end{aligned}$$

where,

$$\begin{aligned}A_g &:= \{((\textcircled{i})_k, \textcircled{i})_k \mid i \in \mathcal{S}, k \in \overline{\mathcal{K}}_i\}, \\ A_r &:= \{((\textcircled{i})_{k-1}, (\textcircled{i})_k) \mid i \in \mathcal{S}, k \in \overline{\mathcal{K}}_i\}, \\ A_c &:= \{((\textcircled{i})_k, (\textcircled{j})_{k'}) \mid \{(i, k), (j, k')\} \in \Psi_R\}, \\ \Psi_R &:= \{((i, k), (j, k')) \mid \{i, j\} \in \Psi_S, k \in \overline{\mathcal{K}}_i, k' \in \overline{\mathcal{K}}_j\}.\end{aligned} \tag{4.6}$$

where $(\textcircled{i})_0$ is defined to equal $(\textcircled{i})_{\overline{K}_i}$ and Ψ_R is a set of conflicting realizations. The above constraint graph equals the constraint graph defined in Section 4.2.2 with \mathcal{K}_i replaced by $\overline{\mathcal{K}}_i$:

In this section we adjust the optimization problem formulated in Section 4.2 to also optimize the number of realizations of each signal group $i \in \mathcal{S}$. First, we elaborate on the

differences with respect to that optimization problem. Thereupon, we give the complete mixed-integer programming formulation.

4.3.1 Additional design variables

With respect to the optimization problem from the previous section, the optimization problem that is proposed in this section has one additional binary-valued design variable $\mathbf{b}_{i,k}$ for each of the realizations $k \in \mathcal{K}_i^d$ of signal group $i \in \mathcal{S}$; the binary-valued design variable $\mathbf{b}_{i,k}$, $k \in \mathcal{K}_i^d$ equals one whenever signal group i has a k th realization and it equals zero otherwise. These binary-valued design variables are related according to the following constraint:

$$\mathbf{b}_{i,k+1} \leq \mathbf{b}_{i,k}, \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i^d \setminus \{\bar{K}_i\}, \quad (4.7a)$$

which implies that signal group i has no $k+1$ st realization if it has no k th realization. We use the binary variable $\mathbf{b}_{i,k}$ to 'switch' realization k of signal group $i \in \mathcal{S}$ on ($\mathbf{b}_{i,k} = 1$) or off ($\mathbf{b}_{i,k} = 0$). In other words, we force effective green interval k of signal group i and its preceding effective red interval to have a duration of zero seconds whenever signal group i has no k th realization, i.e., we force $\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) = 0$ and $\gamma(\textcircled{i}_k, \textcircled{i}_k) = 0$ when $\mathbf{b}_{i,k} = 0$. As a consequence, signal group $i \in \mathcal{S}$ practically has no k th realization when $\mathbf{b}_{i,k} = 0$; from constraint (4.7a) it then follows that signal group $i \in \mathcal{S}$ also has no $k+1$ st realization, et cetera.

4.3.2 Modified constraints

Some constraints of optimization problem (4.1) may obstruct $\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k)$ and $\gamma(\textcircled{i}_k, \textcircled{i}_k)$ from becoming zero when $\mathbf{b}_{i,k} = 0$, e.g., the lower bound on each effective green time may prevent $\gamma(\textcircled{i}_k, \textcircled{i}_k)$ from becoming zero. Therefore, we modify such constraints so that they allow these variables to become zero when $\mathbf{b}_{i,k} = 0$. We replace the bounds on the effective green times (4.1b) as follows; we distinguish between the (certain) realizations $k \in \underline{\mathcal{K}}_i$ and the (uncertain) realizations $k \in \mathcal{K}_i^d$:

$$\underline{g}_i \mathbf{T}' \leq \gamma(\textcircled{i}_k, \textcircled{i}_k) \leq \bar{g}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \underline{\mathcal{K}}_i, \quad (4.7b)$$

$$\underline{g}_i \mathbf{T}' - (1 - \mathbf{b}_{i,k})L \leq \gamma(\textcircled{i}_k, \textcircled{i}_k) \leq \bar{g}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \mathcal{K}_i^d, \quad (4.7c)$$

where L is some large number; in this case $L = \underline{g}_i / \underline{T}$ suffices. The latter constraint becomes redundant when $\mathbf{b}_{i,k} = 0$. Similarly, we replace the bounds on the effective red times (4.1c):

$$\underline{r}_i \mathbf{T}' \leq \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) \leq \bar{r}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \underline{\mathcal{K}}_i, \quad (4.7d)$$

$$\underline{r}_i \mathbf{T}' - (1 - \mathbf{b}_{i,k})L \leq \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) \leq \bar{r}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \mathcal{K}_i^d, \quad (4.7e)$$

where $L = \underline{r}_i/\underline{T}$ is sufficiently large. Furthermore, we replace well-posedness constraint (4.1f). For each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \underline{\mathcal{K}}_i$ we have the original constraint:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) \geq \epsilon \mathbf{T}'. \quad (4.7f)$$

However, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i^d$ we have the following modified constraint:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) + (1 - \mathbf{b}_{i,k})L \geq \epsilon \mathbf{T}', \quad (4.7g)$$

where $L = 1 + \epsilon/\underline{T}$ is sufficiently large.

4.3.3 Additional constraints

We have modified the constraints that may obstruct the k th realization of signal group $i \in \mathcal{S}$ and its preceding effective red interval from having a duration of zero when $\mathbf{b}_{i,k} = 0$. With the following constraints we 'switch off' realization $k \in \mathcal{K}_i^d$ when $\mathbf{b}_{i,k} = 0$:

$$\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k) \leq \mathbf{b}_{i,k}L, \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i^d, \quad (4.7h)$$

$$0 \leq \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k), \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i^d, \quad (4.7i)$$

$$0 \leq \gamma(\textcircled{i}_k, \textcircled{i}_k), \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i^d, \quad (4.7j)$$

where $L = 1$ is sufficiently large. Consider the case that $\mathbf{b}_{i,k} = 0$. The above constraints then force the following events to occur simultaneously: the end of effective green interval $k-1$ (\textcircled{i}_{k-1}), the start of effective green interval k (\textcircled{i}_k), and the end of effective green interval k (\textcircled{i}_k). Since the events \textcircled{i}_{k-1} and \textcircled{i}_k occur simultaneously, this implies for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ that:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) = \gamma(\textcircled{i}_{k-1}, \textcircled{j}_{k'}),$$

when $\mathbf{b}_{i,k} = 0$. Therefore, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i^d$ we include the following constraint:

$$-\mathbf{b}_{i,k}L \leq \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) - \gamma(\textcircled{i}_{k-1}, \textcircled{j}_{k'}) \leq \mathbf{b}_{i,k}L, \quad (4.7k)$$

where $L = 2$ is sufficiently large. Moreover, we have the following equality for each conflict $\{(i, k), (j, k')\} \in \Psi_R$ when $\mathbf{b}_{i,k} = 0$, see Figure 4.4:

$$\gamma(\textcircled{j}_{k'}, \textcircled{i}_k) = \gamma(\textcircled{j}_{k'}, \textcircled{i}_{k-1}) + \gamma(\textcircled{i}_{k-1}, \textcircled{i}_{k-1}),$$

Therefore, for each conflict $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i^d$ we include the following constraint:

$$-\mathbf{b}_{i,k}L \leq \gamma(\textcircled{j}_{k'}, \textcircled{i}_k) - (\gamma(\textcircled{j}_{k'}, \textcircled{i}_{k-1}) + \gamma(\textcircled{i}_{k-1}, \textcircled{i}_{k-1})) \leq \mathbf{b}_{i,k}L. \quad (4.7l)$$

where $L = 2$ is sufficiently large.

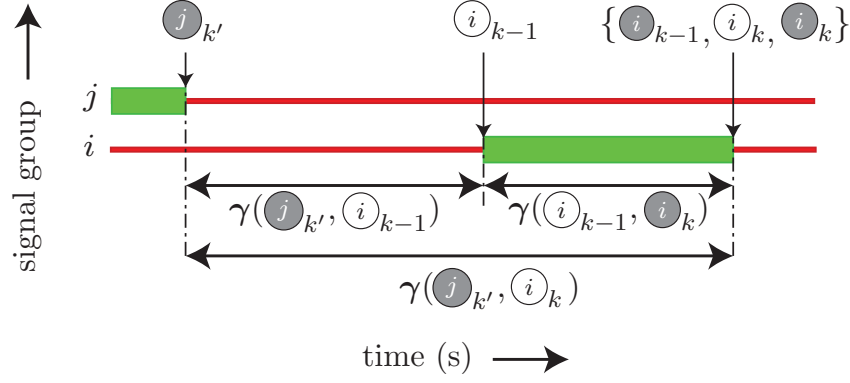


Figure 4.4: Relation between the clearance times when the optimization has decided that signal group i should not have a k th realization. The events i_{k-1} , i_k , and i_k then occur simultaneously.

Remark 4.3. The constraints (4.7k)–(4.7l) are not crucial to the formulated MIP problem; when we omit these constraints then we would find the same signal group diagram as when these constraints are included. However, we include these constraints for the following two reasons. First, to speed up the computation times. With these additional constraints we improve the quality of the LP relaxations. Such an LP relaxation relaxes the integral-valued design variables (and the binary-valued design variables) of a MILP problem to be real-valued design variables. Solving such an LP relaxation results in a lower bound on the corresponding MILP problem. These LP relaxations are important in for example the solvers: CPLEX (International Business Machines Corp, 2015), GUROBI (Gurobi Optimization, Inc., 2015) and SCIP (Achterberg, 2009). A tighter optimization problem (better LP relaxations) is expected to reduce computation times (Maranas and Zomorodi, 2016). Second, these constraints ensure that each variable $\gamma(i_k, j_{k'})$ is defined unambiguously. The well-posedness constraints (4.7f)–(4.7g) ensure this unambiguous definition when $b_{i,k} = 1$. However, they do not when $b_{i,k} = 0$; in that case this unambiguous definition is ensured by constraints (4.7k)–(4.7l).

4.3.4 Complete MIP problem

Below we summarize the complete MIP problem.

Objective function

We can optimize any of the objective functions that have been introduced in Section 4.2.3, i.e., we can minimize the period duration T , maximize the capacity of the intersection, or minimize the average weighted delay that road users experience. The objective function can be written as follows:

$$\underset{T', \gamma, z, b}{\text{minimize}} \quad J(T', \gamma, z, b), \quad (4.8a)$$

where the vector γ contains all arc lengths $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$, the vector z contains all integral-valued design variables z_C , $C \in \mathcal{B}$, and the vector b contains all binary variables

$\mathbf{b}_{i,k}$, $k \in \mathcal{K}_i^d$, $i \in \mathcal{S}$.

Remark 4.4. When maximizing the capacity of the intersection, we have an additional (real-valued) design variable β , which is the growth factor of the arrival rates.

Linear constraints

The MIP problem has the following constraints.

Bounds on the period duration The period duration is bounded from below and from above:

$$1/\bar{T} \leq \mathbf{T}' \leq 1/\underline{T}. \quad (4.8b)$$

Bounds on effective green times and effective red times Each effective green time is bounded from below and bounded from above. The lower bound on the k th effective green time of signal group $i \in \mathcal{S}$ becomes redundant whenever $\mathbf{b}_{i,k} = 0$.

$$\underline{g}_i \mathbf{T}' \leq \gamma(\odot_k, \odot_k) \leq \bar{g}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \underline{\mathcal{K}}_i, \quad (4.8c)$$

$$\underline{g}_i \mathbf{T}' - (1 - \mathbf{b}_{i,k})L \leq \gamma(\odot_k, \odot_k) \leq \bar{g}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \mathcal{K}_i^d. \quad (4.8d)$$

Each effective red time is bounded from below and bounded from above. The lower bounds on the k th effective red time of signal group $i \in \mathcal{S}$ becomes redundant whenever $\mathbf{b}_{i,k} = 0$.

$$\underline{r}_i \mathbf{T}' \leq \gamma(\odot_{k-1}, \odot_k) \leq \bar{r}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \underline{\mathcal{K}}_i, \quad (4.8e)$$

$$\underline{r}_i \mathbf{T}' - (1 - \mathbf{b}_{i,k})L \leq \gamma(\odot_{k-1}, \odot_k) \leq \bar{r}_i \mathbf{T}', \quad i \in \mathcal{S}, k \in \mathcal{K}_i^d. \quad (4.8f)$$

Stability Each signal group needs to be stable:

$$0 < \rho_i^{\text{SG}} \leq \sum_{k \in \mathcal{K}_i} \gamma(\odot_k, \odot_k), \quad i \in \mathcal{S}. \quad (4.8g)$$

Minimum clearance times Minimum clearance times need to be satisfied for each pair of conflicting realizations:

$$\underline{c}_{i,j} \mathbf{T}' \leq \gamma(\odot_k, \odot_{k'}), \quad \{(i, k), (j, k')\} \in \Psi_R. \quad (4.8h)$$

Number of realizations The binary variables $\mathbf{b}_{i,k}$, $k \in \mathcal{K}_i^d$ are used to optimize the number of realizations of signal group i . These binary variables are related according to the following constraints:

$$\mathbf{b}_{i,k+1} \leq \mathbf{b}_{i,k}, \quad i \in \mathcal{S}, k \in \mathcal{K}_i^d \setminus \{\bar{K}_i\}. \quad (4.8i)$$

This binary variable is used to force effective green interval k and effective red interval k of signal group $i \in \mathcal{S}$ to practically not exists whenever $\mathbf{b}_{i,k} = 0$:

$$\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k) \leq \mathbf{b}_{i,k}L, \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i^d, \quad (4.8j)$$

$$0 \leq \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k), \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i^d, \quad (4.8k)$$

$$0 \leq \gamma(\textcircled{i}_k, \textcircled{i}_k), \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i^d. \quad (4.8l)$$

Reducing symmetry We reduce the symmetry of the MIP problem with the following constraints:

$$\gamma(\textcircled{i}_{\bar{K}_i}, \textcircled{i}_1) \geq \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k), \quad i \in \mathcal{S}, \quad k \in \bar{\mathcal{K}}_i \setminus \{1\}. \quad (4.8m)$$

Cycle periodicity constraints The periodicity of the signal group diagram is forced with the cycle periodicity constraints:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2) = \mathbf{z}_{\mathcal{C}}, \quad \forall \mathcal{C} \in \mathcal{B}, \quad (4.8n)$$

where \mathcal{B} is an integral cycle basis of the constraint graph G . For some cycles we know the multiplicity $\mathbf{z}_{\mathcal{C}}$ (and must fix this multiplicity). For each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ it holds that:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) + \gamma(\textcircled{j}_{k'}, \textcircled{j}_{k'}) + \gamma(\textcircled{j}_{k'}, \textcircled{i}_k) = 1, \quad (4.8o)$$

Furthermore, for each signal group $i \in \mathcal{S}$ we have:

$$\sum_{k \in \bar{\mathcal{K}}_i} (\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k)) = 1, \quad (4.8p)$$

Well-posedness constraints The following constraints ensure that the real-valued design variables $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$ are defined unambiguously. For each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \underline{\mathcal{K}}_i$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) \geq \epsilon \mathbf{T}', \quad (4.8q)$$

and for each pair conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i^d$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) + (1 - \mathbf{b}_{i,k})L \geq \epsilon \mathbf{T}', \quad (4.8r)$$

where ϵ is some small positive value.

Relating clearance times If signal group $i \in \mathcal{S}$ does not have a k th realization ($\mathbf{b}_{i,k} = 0$) then the following events are scheduled at the same time: \textcircled{i}_{k-1} , \textcircled{i}_k and \textcircled{i}_k . As a result, the clearance times associated with the k th realization of signal group i are then related to the clearance times associated with the $k - 1$ st realization of this signal group. This relation is expressed in the following constraints. For each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i^d$ we have:

$$-\mathbf{b}_{i,k}L \leq \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) - \gamma(\textcircled{i}_{k-1}, \textcircled{j}_{k'}) \leq \mathbf{b}_{i,k}L. \quad (4.8s)$$

Furthermore, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i^d$ we have:

$$-\mathbf{b}_{i,k}L \leq \gamma(\textcircled{j}_{k'}, \textcircled{i}_k) - (\gamma(\textcircled{j}_{k'}, \textcircled{i}_{k-1}) + \gamma(\textcircled{i}_{k-1}, \textcircled{i}_{k-1})) \leq \mathbf{b}_{i,k}L. \quad (4.8t)$$

This concludes the formulation of the MIP problem. Below we relate the MIP problem that is formulated in this section to the one that is formulated in the previous section.

Relation to a fixed number of realizations

Consider a signal group diagram for which the number of realizations K_i of each signal group $i \in \mathcal{S}$ satisfies $\underline{K}_i \leq K_i \leq \overline{K}_i$. In the following lemma we prove that each such signal group diagram satisfies the linear constraints (4.1) if and only if it satisfies the constraints of MIP problem (4.8). This implies that when we set the binary variables $\mathbf{b}_{i,k}$ as follows:

$$\begin{aligned} \mathbf{b}_{i,k} &= 0, & i \in \mathcal{S}, \quad k &= \underline{K}_i + 1, \dots, K_i, \\ \mathbf{b}_{i,k} &= 1, & i \in \mathcal{S}, \quad k &= K_i + 1, \dots, \overline{K}_i, \end{aligned}$$

then, as desired, the resulting constraints (4.8) permit the same signal group diagrams as the constraints (4.1).

Lemma 4.2. *Let MIP_{fix} be the optimization problem with linear constraints (4.1) and let MIP_{var} be the optimization problem (4.8). Consider a signal group diagram for which the number of realizations K_i of each signal group $i \in \mathcal{S}$ satisfies $\underline{K}_i \leq K_i \leq \overline{K}_i$. This signal group diagram satisfies the constraints of MIP_{fix} if and only if it satisfies the constraints of MIP_{var} , i.e., MIP_{fix} has a solution that results in this signal group diagram if and only if MIP_{var} has a solution that results in this signal group diagram.*

Proof. See Appendix D.3.2 for the proof. □

Well-posedness

Consider a signal group diagram for which signal group $i \in \mathcal{S}$ has K_i realizations and define $\mathcal{K}_i := \{1, \dots, K_i\}$. We prove that each real-valued design variable $\gamma(\varepsilon_1, \varepsilon_2)$ and

its associated integer $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously, i.e., only one value for $\gamma(\varepsilon_1, \varepsilon_2)$ (and only one value for $z(\varepsilon_1, \varepsilon_2)$) can be associated with this signal group diagram. In Section 4.2.2 we have already proved that the variables $\gamma(\varepsilon_1, \varepsilon_2)$ and the associated integers $z(\varepsilon_1, \varepsilon_2)$ are uniquely defined for the variables that are associated with effective green interval $k \in \mathcal{K}_i$ of signal group $i \in \mathcal{S}$, effective red interval $k \in \mathcal{K}_i$ of signal group $i \in \mathcal{S}$, and the clearance time between realization $k \in \mathcal{K}_i$ of signal group $i \in \mathcal{S}$ and conflicting realization $k' \in \mathcal{K}_j$ of signal group $j \in \mathcal{S}$. All other variables $\gamma(\varepsilon_1, \varepsilon_2)$ can only attain one value for this signal group diagram and are therefore also defined unambiguously. We give these values below. For each signal group $i \in \mathcal{S}$ and each realization $k \in \overline{\mathcal{K}}_i \setminus \mathcal{K}_i$, the effective green time and the preceding effective red time are forced to be zero by constraints (4.8j)–(4.8l).

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) := 0,$$

and

$$\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) := 0.$$

We use the definition of Ψ_R as defined in Section 4.3, i.e.,

$$\Psi_R := \{(i, k), (j, k')\} \mid \{i, j\} \in \Psi_S, k \in \overline{\mathcal{K}}_i, k' \in \overline{\mathcal{K}}_j\}.$$

For each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \notin \mathcal{K}_i$ and $k' \in \mathcal{K}_j$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) := \gamma(\textcircled{i}_{K_i}, \textcircled{j}_{k'}),$$

which is forced by (4.8s). Furthermore, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i$ and $k' \notin \mathcal{K}_j$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) := \gamma(\textcircled{i}_k, \textcircled{j}_{K_j}) + \gamma(\textcircled{j}_{K_j}, \textcircled{j}_{K_j}),$$

which is forced by (4.8t), and for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \notin \mathcal{K}_i$ and $k' \notin \mathcal{K}_j$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) := \gamma(\textcircled{i}_{K_i}, \textcircled{j}_{K_j}) + \gamma(\textcircled{j}_{K_j}, \textcircled{j}_{K_j})$$

which is forced by (4.8s) and (4.8t). Therefore, all variables $\gamma(\varepsilon_1, \varepsilon_2)$ and their associated integers $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously.

4.4 Numerical results

In this section we perform an extensive numerical study. To this end, we use the thirteen real-life intersections from (Fleuren and Lefebvre, 2016a), which are categorized by size: small (S), medium (M) and large (L).

For each of these real-life intersections we consider 36 different optimization problems. For all these optimization problems we fix the minimum number of realizations \underline{K}_i , $i \in \mathcal{S}$ to one, i.e., each signal group must have at least one realization. We do however vary the maximum number of realizations \overline{K}_i , $i \in \mathcal{S}$; we consider three variants: we allow the zero, two or four most heavily loaded signal groups to have an additional realization.

We also distinguish between three objective functions: minimizing the period duration, maximizing the capacity of the intersection and minimizing the average delay that road users experience at the intersection. When minimizing the period duration, we consider six different scalings of the arrival rates λ_q : 1.00, 1.05, 1.10, 1.15, 1.20 and 1.25. Together with the three different values for \overline{K}_i , $i \in \mathcal{S}$ this constitutes 18 optimization problems that minimize the period duration. When maximizing the capacity of the intersection, we vary only the values for \overline{K}_i , $i \in \mathcal{S}$ and, therefore, three optimization problems maximize the capacity of the intersection. The remaining 15 optimization problems minimize the average delay that road users experience at the intersection. For each of these optimization problems we fix the period duration. Let T^{\min} be the minimum period duration of any feasible signal group diagram with a single realization per signal group; we can obtain T^{\min} by minimizing the period duration with $\underline{K}_i = \overline{K}_i = 1$ for each signal group $i \in \mathcal{S}$. We consider five different period durations, which are all scalings of the minimum period duration T^{\min} : $1.1T^{\min}$, $1.2T^{\min}$, $1.3T^{\min}$, $1.4T^{\min}$, $1.5T^{\min}$. Together with the three variants for the maximum number of realizations this constitutes 15 optimization problems for each of the thirteen intersection. To obtain good estimates for the computation time that is needed to solve each of these optimization problems, we have solved each optimization problem ten times.

We formulate each of the optimization problems that minimizes the delay as a mixed-integer linear programming problem. We do so as follows. We approximate the delay d_q , $q \in \mathcal{Q}$ with the (extended version of the) formula of Van den Broek, see Section 4.2.3; in Section 4.2.3 we have extended this formula in a straightforward manner to allow signal groups to have multiple realizations, i.e., $\overline{K}_i > 1$. Recall that the average delay equals:

$$D = \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q = \sum_{i \in \mathcal{S}} d_i,$$

where d_i is the contribution of signal group i to the average delay, which equals:

$$d_i = \sum_{q \in \mathcal{Q}_i} w_q d_q.$$

Since we minimize the average delay that road users experience at the intersection, we take w_q proportional to the arrival rate λ_q , i.e., $w_q = \lambda_q/\Lambda$, where $\Lambda := \sum_{q \in \mathcal{Q}} \lambda_q$ is the total arrival rate at the intersection.

We approximate d_i with piecewise linear functions, see also Section 3.3.4. We distinguish between two cases: $\overline{K}_i = 1$ and $\overline{K}_i > 1$. Consider the case that $\overline{K}_i = 1$. We can then approximate d_i with a single linear function as done in Section 3.3.4. When signal

group $i \in \mathcal{S}$ possibly receives multiple realizations, i.e., $\bar{K}_i > 1$, we break d_i into $\bar{K}_i + 1$ different term, see also Section 4.2.3:

$$d_i = d_{i,1}^{\det} + \dots + d_{i,\bar{K}_i}^{\det} + d_i^{\text{stoch}},$$

where,

$$d_{i,k}^{\det} = \sum_{q \in \mathcal{Q}_i} \frac{w_q \gamma(\textcircled{i}_k, \textcircled{i}_k)^2}{2T'(1 - \rho_q)}, \quad (4.9)$$

$$d_i^{\text{stoch}} = \sum_{q \in \mathcal{Q}_i} w_q d_q^{\text{stoch}}.$$

Each of these $\bar{K}_i + 1$ terms is convex (see Appendix D.1) and is approximated by its own piecewise linear function. Let the period duration be fixed to T seconds. The deterministic delay term $d_{i,k}^{\det}$, $k = 1, \dots, \bar{K}_i$ is only a function of the effective red fraction $\gamma(\textcircled{i}_k, \textcircled{i}_k)$. The effective red time $\gamma(\textcircled{i}_k, \textcircled{i}_k)T$ is included in the following interval:

$$\begin{aligned} & \left[\lfloor \max\{\underline{r}_i, T - \bar{K}_i \bar{g}_i - (\bar{K}_i - 1) \bar{r}_i\} \rfloor, \lceil \min\{\bar{r}_i, T - k' \underline{g}_i - (k' - 1) \underline{r}_i\} \rceil \right] \\ & \cup \left[0, T - \max_{q \in \mathcal{Q}_i} \rho_q T - (k' - 1) \underline{r}_i \right), \end{aligned} \quad (4.10)$$

where $k' := \max\{k, \underline{K}_i\}$. We approximate the deterministic delay term $d_{i,k}^{\det}$ by a piecewise linear function with break points $x_0, x_0 + 1, \dots, x_m$ that correspond to the integral effective red times $\gamma(\textcircled{i}_k, \textcircled{i}_k)T$ included in the above interval. Furthermore, we include a lower bound of zero on this deterministic delay term, which prevents this deterministic delay term from becoming negative when $\mathbf{b}_{i,k} = 0$.

The stochastic delay term d_i^{stoch} is a function of the total effective red fraction: $r'_i = \gamma(\textcircled{i}_1, \textcircled{i}_1) + \dots + \gamma(\textcircled{i}_{\bar{K}_i}, \textcircled{i}_{\bar{K}_i})$, see Section 4.2.3. The total effective red time $r_i = r'_i T$ is included in the following interval:

$$\left[\lfloor \max\{\underline{K}_i \underline{r}_i, T - \bar{K}_i \bar{g}_i\} \rfloor, \lceil \min\{\bar{K}_i \bar{r}_i, T - \underline{K}_i \underline{g}_i\} \rceil \right] \cup \left[0, T - \max_{q \in \mathcal{Q}_i} \rho_i T \right). \quad (4.11)$$

We approximate this stochastic delay term with a piecewise linear function with break points $x_0, x_0 + 1, \dots, x_m$ that correspond to the integral total effective red times included in the above interval.

In Table 4.3 and Table 4.4 we give the results for the test cases that minimize the period duration. In Table 4.3 we give the objective values of these test cases, and in Table 4.4 we give the improvement in the objective value when allowing some of the most heavily loaded signal groups to have an additional realization. It appears that for our test cases, allowing signal groups to have multiple realizations did not decrease the minimum period duration. An exception is the intersection S4; for this intersection the minimum period duration decreased (up to) 2.9 percent.

In Table 4.5 we give the objective values for the test cases that maximize the growth factor of the arrival rates. Furthermore, in this table we give the improvement in the

objective value when allowing some of the most heavily loaded signal groups to have an additional realization; an improvement is observed for the intersection S4 (1.16 percent) and for the intersection L2 (0.72 percent).

In Table 4.6 and Table 4.7 we give the results for the test cases that minimize the average delay that road users experience. In Table 4.6 we give the objective values of these test cases, and in Table 4.4 we give the improvement in the objective value when allowing some of the most heavily loaded signal groups to have an additional realization. When minimizing the delay that road users experience, a substantial improvement (of often several percent) can be made by allowing some of the most heavily loaded signal groups to have an additional realization. Note that the delays are relatively small at the larger period durations (of for example a period duration that equals $1.5T^{\min}$). For these larger period durations, even smaller delays can be achieved by allowing several signal groups to have multiple realizations; at these larger period duration, more freedom may be available to schedule the realizations and, as a consequence, it may for these larger period durations be easier to fit in additional realizations in the signal group diagram. For this test set we allow at most four of the most heavily loaded signal groups to have an additional realization. This already results in substantial improvements. By using either expertise or a trial-and-error approach to determine which signal groups should receive multiple realizations, an even larger improvement may be possible.

In Table 4.8 we have shown the computation times for the numerical study. We have obtained these computation times for three different solvers: CPLEX version 12.6.1.0 (International Business Machines Corp, 2015), GUROBI version 6.0.5 (Gurobi Optimization, Inc., 2015) and SCIP version 3.2.0 (Achterberg, 2009). The computation times are increasing in the number of signal groups that is allowed to have an additional realization. For this numerical study, it seems that the solver CPLEX is best able to handle an increase in the number of additional realizations

Scaling 'Ad. real.'	1.00			1.05			1.10			1.15			1.20			1.25		
	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4
S1	35.86	35.86	35.86	39.32	39.32	39.32	43.51	43.51	43.51	48.71	48.71	48.71	55.32	55.32	55.32	64.00	64.00	64.00
S2	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.84	30.84	30.84	34.30	34.30	34.30
S3	39.11	39.11	39.11	40.72	40.72	40.72	43.06	43.06	43.06	45.69	45.69	45.69	48.66	48.66	48.66	52.05	52.05	52.05
S4	65.75	65.75	65.75	69.00	69.00	69.00	73.02	73.02	73.02	76.77	76.77	76.77	81.21	81.21	81.21	86.20	86.20	86.20
S5	64.14	64.14	64.14	67.53	67.53	67.53	71.30	71.30	71.30	75.52	75.52	75.52	82.95	82.95	82.95	92.41	92.41	92.41
M1	43.70	43.70	43.70	45.56	45.56	45.56	47.69	47.69	47.69	50.03	50.03	50.03	56.26	56.26	56.26	67.90	67.90	67.90
M2	45.55	45.55	45.55	48.08	48.08	48.08	52.24	52.24	52.24	57.18	57.18	57.18	63.16	63.16	63.16	70.53	70.53	70.53
M3	141.05	141.05	141.05	167.40	167.40	167.40	205.86	205.86	205.86	∞	∞	∞	∞	∞	∞	∞	∞	∞
M4	80.01	80.01	80.01	85.81	85.81	85.81	92.53	92.53	92.53	100.38	100.38	100.38	109.69	109.69	109.69	120.91	120.91	120.91
L1	74.71	74.71	74.71	78.10	78.10	78.10	81.81	81.81	81.81	86.37	86.37	86.37	121.75	121.75	121.75	∞	∞	∞
L2	83.59	83.59	83.59	89.83	89.83	89.83	97.07	97.07	97.07	105.58	105.58	105.58	115.73	115.73	115.73	∞	∞	∞
L3	74.57	74.57	74.57	80.35	80.35	80.35	84.62	84.62	84.62	92.26	92.26	92.26	103.76	103.76	103.76	118.53	118.53	118.53
L4	71.55	71.55	71.55	74.67	74.67	74.67	80.50	80.50	80.50	88.97	88.97	88.97	99.44	99.44	99.44	112.71	112.71	112.71

Table 4.3: The objective values for each of the $13 \times 6 \times 3$ test cases in the numerical study that minimizes the period duration. The first column indicates which intersection is considered, the first row indicates the scaling of the arrival rates is considered, and the second row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations. This table contains infinite values; these infinite values indicate that the corresponding MILP problems are infeasible.

Scaling	1.00		1.05		1.10		1.15		1.20		1.25	
Ad. real.	2	4	2	4	2	4	2	4	2	4	2	4
S1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
S2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
S3	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
S4	0.0%	0.0%	0.0%	0.0%	0.0%	-1.3%	0.0%	-2.8%	0.0%	-2.9%	0.0%	-2.9%
S5	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
M1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
M2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
M3	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	-	-	-	-	-	-
M4	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
L1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	-	-
L2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	-	-
L3	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
L4	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Table 4.4: Decrease in the objective values when minimizing the period duration and allowing some of the most heavily loaded signal groups to have an additional realization; these improvements are defined with respect to the case that each signal group has only one realization. The first column indicates which intersection is considered, the first row indicates which scaling of the arrival rates is considered, and the second row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Ad. real.	0	2	4
S1	1.39	1.39 (0.00%)	1.39 (0.00%)
S2	1.56	1.56 (0.00%)	1.56 (0.00%)
S3	1.66	1.66 (0.00%)	1.66 (0.00%)
S4	1.48	1.48 (0.00%)	1.50 (1.16%)
S5	1.35	1.35 (0.00%)	1.35 (0.00%)
M1	1.35	1.35 (0.00%)	1.35 (0.00%)
M2	1.36	1.36 (0.00%)	1.36 (0.00%)
M3	1.11	1.11 (0.00%)	1.11 (0.00%)
M4	1.34	1.34 (0.00%)	1.34 (0.00%)
L1	1.22	1.22 (0.00%)	1.22 (0.00%)
L2	1.23	1.23 (0.00%)	1.24 (0.72%)
L3	1.25	1.25 (0.00%)	1.25 (0.00%)
L4	1.27	1.27 (0.00%)	1.27 (0.00%)

Table 4.5: The objective values for each of the 13×3 test cases in the numerical study that maximizes the growth factor of the arrival rates that is sustainable. In this table, we also visualize (between brackets) the increase in the objective value with respect to the case that each signal group has only one realization. The first column indicates which intersection is considered, and the first row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Scaling	1.1			1.2			1.3			1.4			1.5		
	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4
Ad. real.															
S1	22.71	22.71	22.71	17.13	17.13	17.13	15.16	15.16	15.16	14.20	14.20	14.20	13.75	13.75	13.72
S2	14.98	14.98	14.98	14.61	14.61	14.61	14.38	14.38	14.38	14.31	14.31	14.31	14.50	14.50	14.50
S3	47.78	47.78	47.79	31.72	31.72	31.72	26.58	26.58	26.58	24.55	24.55	24.55	23.62	23.62	23.62
S4	37.60	37.60	30.80	27.26	27.26	24.80	24.94	24.94	23.35	24.33	24.33	22.36	24.45	23.74	21.36
S5	44.33	44.33	44.33	34.09	34.09	34.09	31.80	31.80	31.80	30.91	30.91	30.91	30.89	30.89	30.89
M1	34.17	34.17	33.53	25.57	25.57	24.85	23.70	23.70	22.98	22.79	22.79	22.08	22.26	22.26	21.59
M2	41.67	41.67	40.70	29.73	29.73	28.85	26.70	26.70	25.88	25.55	25.55	24.78	25.18	25.18	24.46
M3	86.37	86.37	79.53	64.43	64.43	61.64	58.19	58.19	57.47	56.62	56.62	56.50	56.74	54.35	54.14
M4	47.71	47.71	45.70	36.82	36.82	35.19	33.24	33.24	31.93	31.95	31.95	30.79	31.71	30.38	29.74
L1	40.48	40.48	39.79	38.12	37.90	36.45	38.56	38.27	36.68	39.46	39.17	37.39	40.58	40.12	37.95
L2	61.97	61.97	61.97	50.15	50.15	50.15	48.51	48.51	47.30	48.61	48.61	44.82	49.60	48.09	44.30
L3	37.61	37.61	37.61	33.05	33.05	33.05	32.33	32.33	32.33	32.67	32.40	31.49	33.46	33.09	31.53
L4	32.61	31.74	31.74	29.14	28.19	28.19	28.66	27.62	27.42	29.07	27.96	27.65	29.86	28.68	28.32

Table 4.6: The objective values for each of the $13 \times 5 \times 3$ test cases in the numerical study that minimizes the average delay that road users experience. For each of these test cases, the period duration is fixed to some scaling (> 1) of the minimum period duration; the minimum period duration can be found in column 2 of Table 4.3. The first column indicates how many signal groups are allowed, the first row indicates which scaling of the minimum period duration is considered, and the second row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Scaling	1.1		1.2		1.3		1.4		1.5	
Ad. real.	2	4	2	4	2	4	2	4	2	4
S1	0.0%	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	-0.3%
S2	0.0%	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0%
S3	0.0%	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0%
S4	0.0%	-0.2%	0.0 %	-9.0%	0.0 %	-6.4%	0.0 %	-8.1%	-2.9%	-12.7%
S5	0.0%	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0%
M1	0.0%	0.0 %	0.0 %	-2.8%	0.0 %	-3.1%	0.0 %	-3.1%	0.0 %	-3.0%
M2	0.0%	0.0 %	0.0 %	-3.0%	0.0 %	-3.1%	0.0 %	-3.0%	0.0 %	-2.9%
M3	0.0%	-0.1%	0.0 %	-4.3%	0.0 %	-1.2%	0.0 %	-0.2%	-4.2%	-4.6%
M4	0.0%	0.0 %	0.0 %	-4.4%	0.0 %	-4.0%	0.0 %	-3.6%	-4.2%	-6.2%
L1	0.0%	0.0 %	-0.6%	-4.4%	-0.7%	-4.9%	-0.7%	-5.3%	-1.1%	-6.5%
L2	0.0%	0.0 %	0.0 %	0.0 %	0.0 %	-2.5%	0.0 %	-7.8%	-3.0%	-10.7%
L3	0.0%	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	-0.8%	-3.6%	-1.1%	-5.8%
L4	0.0%	0.0 %	-3.3%	-3.3%	-3.6%	-4.3%	-3.8%	-4.9%	-3.9%	-5.2%

Table 4.7: Decrease in the objective values when minimizing the average delay that road users experience and allowing some signal groups to have an additional realization; these improvements are defined with respect to the case that each signal group has only one realization. For each of these test cases, the period duration is fixed to some scaling (> 1) of the minimum period duration; the minimum period duration can be found in column 2 of Table 4.3. The first column indicates which intersection is considered, the first row indicates which scaling of the minimum period duration is considered, and the second row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Solver	Intersections	min T			max β			min D		
		0	2	4	0	2	4	0	2	4
CPLEX	Small	0.06	0.09	0.11	0.06	0.10	0.14	0.08	0.13	0.29
	Medium	0.08	0.13	0.22	0.09	0.14	0.25	0.19	0.48	1.17
	Large	0.20	0.42	1.08	0.20	0.37	1.09	1.31	4.04	13.21
GUROBI	Small	0.00	0.01	0.04	0.00	0.01	0.06	0.01	0.03	0.16
	Medium	0.02	0.11	0.78	0.02	0.07	0.42	0.10	0.47	1.35
	Large	0.20	3.30	41.39	0.22	0.60	2.13	1.23	5.52	17.79
SCIP	Small	0.02	0.04	0.10	0.02	0.07	0.31	0.04	0.14	0.48
	Medium	0.06	0.23	0.74	0.11	0.30	1.38	0.23	0.78	2.77
	Large	0.57	2.19	10.92	0.66	6.25	18.88	3.00	13.33	57.07

Table 4.8: The (geometric) average computation times (in seconds) needed by the approach that is proposed in this chapter. We distinguish between three types of optimization problems (min T , max β and min D), three types of intersections (small, medium and large), and three types of solvers (CPLEX 12.6.1.0, GUROBI 6.0.5. and SCIP 3.2.0). Furthermore, we vary the number of signal groups that is allowed to have an additional realization, see the second row of this table. If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Remark 4.5. *In (Fleuren, 2016) we elaborate on several techniques to reduce the computation time needed to solve optimization problem (4.8). In the upcoming paragraphs we briefly elaborate on these techniques.*

Note that the computation times are especially large when minimizing the delay that road users experience at the intersection. The convex objective function is then approximated with piecewise linear functions. To reduce the number of linear segments that is needed (while keeping the same accuracy) and, as a result, speed up the computation time, we obtain tighter bounds on some of the intervals in time, e.g., each effective red interval. Furthermore, we obtain a tighter optimization formulation by adding several constraints (cuts) to the optimization problem. This results in a tighter formulation of the optimization problem, which is expected to reduce the computation times (Maranas and Zomorodi, 2016).

Moreover, to formulate optimization problem (4.8), we require an integral cycle basis of the constraint graph G . In (Fleuren, 2016) we attempt to find 'good' integral cycle bases, i.e., we try to find a small-width integral cycle basis. To this end, a novel method is proposed. This approach constructs an integral cycle basis of the constraint graph from an integral cycle basis of a much smaller graph, which is called the directed conflict graph. This directed conflict graph has one vertex for each signal group and one (directed) arc for each pair of conflicting signal groups. Much care is devoted to finding an integral cycle basis of the directed conflict graph that results in a small-width integral cycle basis of the constraint graph.

The techniques described in (Fleuren, 2016) result (on average) in a reduction in computation times of approximately 40%.

4.5 Summary

In this chapter we have extended the optimization framework introduced in Chapter 3. This extension allows the optimization over the number of realizations of each signal group. First, in Section 4.2 we have considered the number of realizations of each signal group to be a fixed and given value. In that section we have formulated an optimization problem to simultaneously optimize: the period duration of the signal group diagram, when each of the realizations start, and when these realization end. The proposed optimization formulation closely resembles the optimization problem proposed in Chapter 3; however, in contrast to that chapter, a signal group is allowed to have multiple realizations. Possible objective functions of the optimization framework are: minimizing the period duration of the signal group diagram, maximizing the capacity of the intersection, and minimizing the average delay that road users experience at the intersection. One of the differences with Chapter 3 is the objective function when minimizing the delay that road users experience at the intersection. In the previous chapter, we could compute the delay d_q that road users experience at a queue $q \in \mathcal{Q}$ under pre-timed control with, for example, the formulae

of (Miller, 1963; van den Broek et al., 2006; Webster, 1958). However, all these formulae assume that a signal group receives only one realization. Therefore, we have extended these approximations, in a straightforward manner, to allow for multiple realizations.

Subsequently, in Section 4.3 we have considered the number of realizations of each signal group to be a design variable and formulated an optimization problem to simultaneously optimize: the period duration of the signal group diagram, the number of realizations of each signal group, when each of these realizations start, and when these realizations end. This optimization formulation uses binary variables to optimize the number of realizations of each signal group. Each such binary variable is used to switch on (or off) a specific realization; when this binary equals zero, then this realization (and its preceding effective red time) is forced to have a duration of zero seconds and, as a consequence, this realization then practically does not exist.

Finally, in Section 4.4 we have performed an extensive numerical case study. For this numerical study, we have concluded that allowing several signal groups to have multiple realizations has little (or no) effect when minimizing the period duration and also has little (to no) effect when maximizing the capacity of the intersection. However, the average delay that road users experience can decrease substantially by allowing signal groups to have multiple realizations. For our test case, this decrease was often several percent. For these test cases we optimize the number of realizations for at most four signal groups (which are the four most heavily loaded signal groups); each of these signal groups is allowed to have an additional realization. All other signal groups have a single realization. An even larger decrease is probably possible when we use either expertise or a trial-and-error approach to determine which signal groups should receive multiple realizations.

In the next chapter we consider the problem of finding integral signal group diagrams. For such an integral signal group diagram, the period duration is integral and all switches to green, yellow and red are scheduled at an integral second. Thereupon in Chapter 6 we consider the problem of optimizing the layout of the intersection. We can, for example, find the layout that maximizes the capacity at the intersection. To this end, we simultaneously have to optimize the layout of the intersection and a signal group diagram that specifies when each of the traffic lights switch to green, yellow and red.

Integral signal group diagrams

5.1 Introduction

In Chapter 4 we have formulated an optimization framework to optimize signal group diagrams. With this optimization framework we can for example find the optimal signal group diagram with respect to the following objective functions: minimization of the period duration, maximization of the capacity of the intersection, and minimization of the average (weighted) delay that road users experience at the intersection. In this chapter we consider the optimization of *integral signal group diagrams*. A integral signal group diagram is defined as follows:

Definition 5.1 (Integral signal group diagram). *A signal group diagram is integral whenever its period duration T is integral and all switches to green, yellow and red are scheduled at an integral second.*

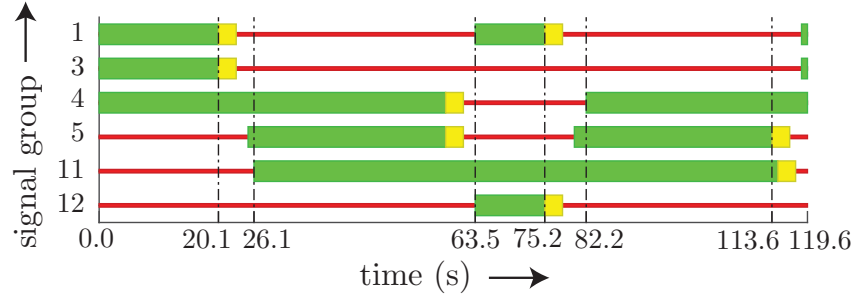
These integral signal group diagrams are desired in practice because they are clear, presentable and easy to work with. In this chapter we consider the optimization of such integral signal group diagrams. We simultaneously optimize: the period duration, the number of realizations that each signal group has, when these realizations start, and when these realizations end. Throughout this chapter we assume:

Assumption 5.1. *For each signal group $i \in \mathcal{S}$ for which the number of realizations is optimized, we assume the starting lost time l_i^s plus the ending lost time l_i^e to be integral, i.e., we assume $l_i^s + l_i^e \in \mathbb{Z}$ for each signal group $i \in \mathcal{S}$ that satisfies $\underline{K}_i < \overline{K}_i$.*

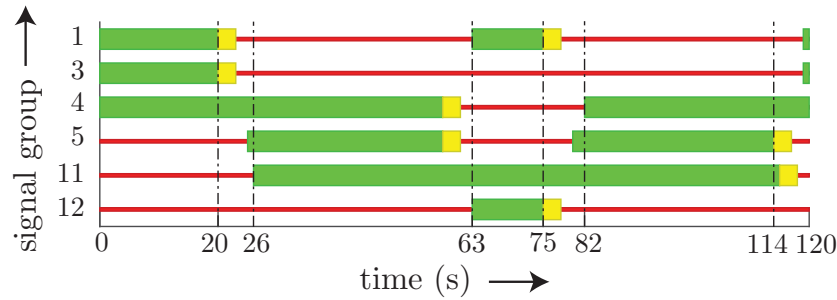
Consider again the example introduced in Section 3.2.1. In Chapter 4 we have already obtained the optimal signal group diagram for this example. We again visualize this signal group diagram in Figure 5.1a. Road users experience an average delay of 25.106 seconds for this signal group diagram. In Figure 5.1b we depict the integral signal group diagram obtained with the optimization procedure proposed in this chapter. The average delay

This chapter is based on the following paper: Fleuren, S. and Lefebvre, E. (2016b). Integral fixed-times schedules for traffic light control

that road users experience for this integral signal group diagram equals 25.133 seconds, which is only slightly larger than the average delay that road users experience for the optimal signal group diagram obtained in Chapter 4.



(a) The signal group diagram obtained in Chapter 4.



(b) An integral signal group diagram.

Figure 5.1: A signal group diagram that is not integral and a signal group diagram that is integral for the T-junction introduced in Section 3.2.1. The average delay that road users experience for the integral signal group diagram is 25.133 seconds and the average delay that road users experience for the signal group diagram that is not integral is 25.106 seconds. For esthetic reasons we only give the first digit after the decimal point for the signal group diagram that is not integral

It is possible to find an integral signal group diagram by first optimizing over all signal group diagrams (also the ones that are not integral). Subsequently, from this signal group diagram we can obtain an integral signal group diagram by, for example, rounding the period duration and all switches to green, yellow and red to the nearest integer. Unfortunately, it is very likely that the resulting integral signal group diagram does not satisfy all the constraints, e.g., it possibly does not satisfy the minimum clearance times that need to be satisfied. Nevertheless, from a practical point of view such an approach might be appropriate; the result is an integral signal group diagram for which each constraint is violated only marginally at worst. However, in this chapter we take a more rigorous approach; we propose a fast approach that optimizes over only the integral signal group diagrams that are feasible.

The approach proposed in this chapter consists of two steps. In the first step of this two-step approach we search for the optimal signal group diagram amongst the signal group diagrams that satisfy some structural property; this structural property is also satisfied by any integral signal group diagram. The result of this first step is not necessarily an integral signal group diagram. However, forcing the resulting signal group diagram to

satisfy the structural property makes it easier to obtain a feasible integral signal group diagram during the second step.

In the second step we solve a *rounding MILP problem*. This relatively simple MILP problem attempts to (quickly) obtain an integral signal group diagram from the not necessarily integral signal group diagram obtained in the first step; this optimization problem searches for a feasible integral signal group diagram for which each green time and each red time (of the not necessarily integral signal group diagram) is rounded upwards or downwards.

We are able to prove the following statements concerning the optimality of the two-step approach. Consider the problem of minimizing the period duration of the signal group diagram or the problem of maximizing the capacity at the intersection. For these two objective functions the two-step approach returns the optimal integral signal group diagram when $\bar{K}_i > 1$ for at most one signal group $i \in \mathcal{S}$. Consider the minimization of the average (weighed) delay that road users experience at the intersection. We can prove that the two-step approach then returns the optimal integral signal group diagram when $K_i = 1$ for all signal groups $i \in \mathcal{S}$.

We test the two-step approach for a large test set; for all of the test cases in this large test set, we are able to find a feasible integral signal group diagram with the two-step approach. For each test case for which the period duration is minimized and for each test case for which the capacity of the intersection is maximized we are able to find the optimal integral signal group diagram. For some of the test cases for which the delay that road users experience is minimized, the returned integral signal group diagram is (possibly) not optimal; however, these integral signal group diagrams are then close to optimality.

This chapter is organized as follows. First in Section 5.2 we give some preliminaries. Subsequently, in Section 5.3 we introduce some notation. Thereupon, in Section 5.4 we elaborate on the input data required for the optimization of integral signal group diagrams. In Section 5.5 we give a structural property that is satisfied by any feasible integral signal group diagram. Subsequently, we consider the optimization of integral signal group diagrams with a two-step approach, which is split in two parts; in Section 5.6 we consider the first step of the two-step approach and in Section 5.7 we consider the second step. Finally, in Section 5.8 we perform a numerical study to test these optimization procedures and in Section 5.9 we give a summary.

5.2 Preliminaries

In this section we give all preliminaries required in this chapter. First, in Section 5.2.1, we introduce some new definitions regarding cycles and paths in the constraint graph G . The solutions to a set of linear constraints span a polyhedron. Such polyhedra play an important role in this chapter. In Section 5.2.2 we elaborate on these polyhedra. The

final section, which is Section 5.2.3, concerns the integrality of the optimal solution to a (mixed-integer) linear programming problem.

5.2.1 Cycles and paths

In this section we introduce some definitions regarding cycles and paths. Moreover, we present different characterizations of integral cycle bases. One of these characterizations is in terms of the determinant of a cycle basis, as given in (Liebchen, 2003).

Cycles

Consider a graph $G = (V, A)$. First, we recall the definition of a cycle-arc incidence vector:

Definition 5.2 (Cycle-arc incidence vector). *Associated with each cycle \mathcal{C} in the graph G is a row vector $C \in \{-1, 0, +1\}^{|A|}$ s.t. for each arc $a \in A$:*

$$C(a) = \begin{cases} +1 & \text{if } a \in C^+, \\ 0 & \text{if } a \notin C, \\ -1 & \text{if } a \in C^-. \end{cases}$$

This vector is called the cycle-arc incidence vector of the cycle \mathcal{C} .

The cycle-arc incidence vectors of the cycles in the graph G span a space, which we call the cycle space. This cycle space has a dimension $d = |A| - |V| + \nu(G)$, where $\nu(G)$ is the number of connected components of the graph G . A cycle basis is a set of cycles that span the cycle space:

Definition 5.3 (Cycle basis). *A cycle basis $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ is a set of independent cycles, i.e., the cycle-arc incidence vectors C_i , $i = 1, \dots, d$ are independent, such that for each cycle \mathcal{C} in the graph G we can find $\alpha \in \mathbb{R}^d$ such that:*

$$C = \alpha_1 C_1 + \dots + \alpha_d C_d.$$

An integral cycle basis is a special type of cycle basis:

Definition 5.4 (Integral cycle basis). *An integral cycles basis $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ is a set of independent cycles such that for each cycle \mathcal{C} in the graph G we can find $\alpha \in \mathbb{Z}^d$ such that:*

$$C = \alpha_1 C_1 + \dots + \alpha_d C_d.$$

With each cycle basis we can associate a cycle-arc incidence matrix:

Definition 5.5 (Cycle-arc incidence matrix). *Let $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ be a cycle basis of the graph G . The $d \times |A|$ cycle-arc incidence matrix M is comprised of the cycle-arc incidence vectors C_i , $i = 1, \dots, d$; the i th row of M equals C_i .*

Lemma 5.1 (Liebchen, 2003). *Let $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ be a cycle basis of the graph G and let X_1 and X_2 be two non-singular $d \times d$ submatrices of the cycle-arc incidence matrix M associated with cycle basis \mathcal{B} . Then $\det(X_1) = \pm \det(X_2)$.*

This lemma implies that the determinant of a cycle basis can be defined as follows:

Definition 5.6 (Determinant of a cycle basis). *The determinant of a cycle basis $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ is $\text{abs}(\det(X))$, where X is a non-singular $d \times d$ submatrix of the associated cycle-arc incidence matrix M and where $\text{abs}(x)$ is the absolute value of x .*

An integral cycle basis can be characterized by this determinant:

Theorem 5.2 (Liebchen, 2003). *A cycle basis is integral if and only if its determinant is 1.*

In this chapter, we need the definition of a totally unimodular cycle basis, which is a special type of integral cycle basis. Before we introduce its formal definition, we first have to define a totally unimodular matrix:

Definition 5.7 (Totally unimodular (TU) matrix). *A matrix A is called totally unimodular (TU) if and only if every square submatrix of A has a determinant in $\{-1, 0, +1\}$.*

Definition 5.8 (Totally unimodular cycle basis (TUCB)). *A cycle basis $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ is a totally unimodular cycle basis if its cycle-arc incidence matrix M is totally unimodular.*

For an integral cycle basis it holds that every $d \times d$ submatrix of its cycle-arc incidence matrix has a determinant in $\{-1, 0, +1\}$. However, for a totally unimodular cycle basis also every other (square) submatrix of this cycle-arc incidence matrix has a determinant in $\{-1, 0, +1\}$. Therefore, the class of totally unimodular cycle bases is included in the class of integral cycle bases. Furthermore, such a totally unimodular cycle basis exists for each graph, see for example (Kavitha et al., 2009).

In Figure 5.2 we give the inclusions of the different types of cycle bases used in this thesis. We have obtained these inclusions from (Kavitha et al., 2009). In this figure we visualize the inclusions amongst the following types of cycle bases: integral cycle bases (ICB), totally unimodular cycle bases (TUCB), and strictly fundamental cycle bases (SFCB). We have already introduced the set of integral cycle bases and the set of strictly fundamental cycle bases in Chapter 3. The set of totally unimodular cycle bases plays an important role in this chapter.

Paths

Similar to a cycle-arc incidence vector we also define a path-arc incidence vector:

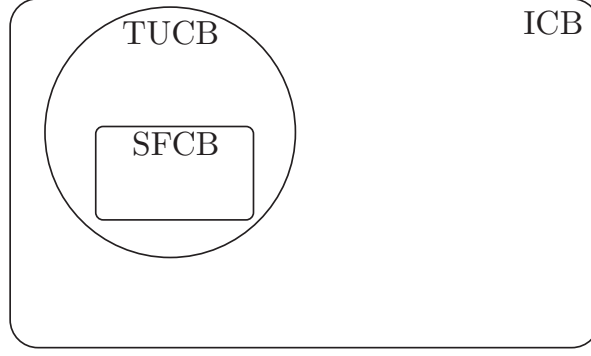


Figure 5.2: Inclusions of the different types of cycle bases that are used in this thesis. In this thesis we use the following cycle bases: integral cycle bases (ICB), totally unimodular cycle bases (TUCB), and strictly fundamental cycle bases (SFCB).

Definition 5.9 (Path-arc incidence vector). Associated with each path \mathcal{P} in graph $G = (V, A)$ is a row vector $P \in \{-1, 0, +1\}^{|A|}$ s.t. for each $a \in A$:

$$P(a) = \begin{cases} +1 & \text{if } a \in \mathcal{P}^+, \\ 0 & \text{if } a \notin \mathcal{P}, \\ -1 & \text{if } a \in \mathcal{P}^-. \end{cases}$$

This vector is called the path-arc incidence vector of path \mathcal{P} .

Furthermore, we define a path-arc incidence matrix:

Definition 5.10 (Path-arc incidence matrix). Let $\{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ be a set of paths in $G = (V, A)$. The $n \times |A|$ path-arc incidence matrix M_P associated with the paths $\{\mathcal{P}_1, \dots, \mathcal{P}_n\}$ is comprised of the path-arc incidence vectors P_i ; the i th row of M_P equals P_i , which is the path-arc incidence vector of path \mathcal{P}_i .

5.2.2 Polyhedra

Polyhedra play an important role in this chapter. A \mathcal{H} -polyhedron is defined as follows.

Definition 5.11 (\mathcal{H} -Polyhedron). A \mathcal{H} -polyhedron P is a set of the form $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

In this thesis we may also refer to such a \mathcal{H} -polyhedron with 'polyhedron'.

Definition 5.12 (Vertex of a polyhedron). Consider some \mathcal{H} -Polyhedron P . A vector $x \in P$ is called a vertex of P if and only if x cannot be written as the convex combination of two other vectors $x_1 \in P$ and $x_2 \in P$, i.e., $x \in P$ is a vertex of P if and only if $\nexists x_1, x_2 \in P, x_1, x_2 \neq x$ such that $x = \alpha x_1 + (1 - \alpha)x_2$ for some $\alpha \in (0, 1)$.

Definition 5.13 (Integral polyhedron). A \mathcal{H} -polyhedron is called integral if and only if all its vertices are integral valued.

Consider a \mathcal{H} -polyhedron of the form $P := \{x \in \mathbb{R}^n \mid \underline{b} \leq Ax \leq \bar{b}, \underline{x} \leq x \leq \bar{x}\}$. Note that this polyhedron can be written in the original form as follows:

$$P := \{x \in \mathbb{R}^n \mid A'x \leq b'\},$$

where,

$$A' := \begin{bmatrix} A \\ -A \\ I \\ -I \end{bmatrix}, \quad b' := \begin{bmatrix} \bar{b} \\ -\underline{b} \\ \underline{x} \\ -\bar{x} \end{bmatrix}.$$

The following theorem states that this polyhedron P is integral if $\underline{b}, \bar{b}, \underline{x}$, and \bar{x} are integral and A is TU.

Theorem 5.3 (Hoffman and Kruskal, 1956). *Let $\underline{b}, \bar{b}, \underline{x}$ and \bar{x} be integral vectors and A be a totally unimodular matrix. Then the polyhedron $\{x \mid \underline{b} \leq Ax \leq \bar{b}, \underline{x} \leq x \leq \bar{x}\}$ is integral.*

5.2.3 Integrality of the optimal solution

In this section we first elaborate on the integrality of the optimal solution to a linear programming problem. Subsequently, we elaborate on the integrality of the optimal solution to a mixed-integer linear programming problem.

Linear programming problem

Consider the following linear programming problem:

$$\begin{aligned} & \min_{\mathbf{x}} \quad c\mathbf{x}, \\ & \text{subject to:} \\ & \quad \underline{b} \leq A\mathbf{x} \leq \bar{b}, \\ & \quad \underline{x} \leq \mathbf{x} \leq \bar{x}, \\ & \text{where,} \\ & \quad \mathbf{x} \in \mathbb{R}^n, \\ & \quad \underline{b}, \bar{b} \in \mathbb{Z}^m, \\ & \quad \underline{x}, \bar{x} \in \mathbb{Z}^n, \\ & \quad A \text{ is TU} \end{aligned} \tag{5.1}$$

Let P be the polyhedron spanned by all feasible solutions to the constraints of linear programming problem (5.1):

$$P := \{\mathbf{x} \mid \underline{b} \leq A\mathbf{x} \leq \bar{b}, \underline{x} \leq \mathbf{x} \leq \bar{x}\}.$$

Let x_1, x_2, \dots, x_n be the vertices of P . The optimal solution to the linear programming problem (5.1) is positioned at one of these vertices x_1, x_2, \dots, x_n . This follows from the fact that any feasible point $x \in P$ can be written as a convex combination of the vertices of P , i.e., for each vector $x \in P$ it holds that:

$$x = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n,$$

for some α such that $\alpha_i \in [0, 1]$, $i = 1, \dots, n$ and $\sum_{i=1}^n \alpha_i = 1$. As a consequence, the objective value associated with any feasible point $x \in P$ can also be written as a convex combination of objective values:

$$cx = \alpha_1 cx_1 + \alpha_2 cx_2 + \dots + \alpha_n cx_n,$$

where $\alpha_i \in [0, 1]$, $i = 1, \dots, n$ and $\sum_{i=1}^n \alpha_i = 1$. This implies that for each such feasible point $x \in P$, at least one vertex exists for which the objective value is at most (and also at least) that of x .

Since an optimal solution is positioned at one of the vertices of P , from Theorem 5.3 it follows that linear programming problem (5.1) has an optimal solution that is integral. A linear programming problem can be solved with for example a simplex algorithm. Such an algorithm only searches for the optimal solution amongst the vertices of P . As a result, this simplex algorithm returns an integral solution when solving linear programming problem (5.1).

Mixed-integer linear programming problem

Consider a mixed-integer linear programming (MILP) problem. For such a problem, some of the design variables are restricted to be integral-valued. Fixing the value of these integral-valued design variables results in a linear programming problem. Let (\mathbf{x}, \mathbf{z}) be a solution to the mixed-integer programming problem, where \mathbf{x} is a vector that contains the values of the real-valued design variables and \mathbf{z} is a vector that contains the values of the integral-valued design variables. Define $\text{LP}(\mathbf{z})$ to be the LP problem that results from setting the integral-valued design variables to the values in \mathbf{z} . Let $(\mathbf{z}^*, \mathbf{x}^*)$ be the optimal solution to the MILP problem. The vector \mathbf{x}^* , by definition, satisfies the constraints of $\text{LP}(\mathbf{z}^*)$. Furthermore, if $\text{LP}(\mathbf{z}^*)$ is of the form (5.1), then $\text{LP}(\mathbf{z}^*)$ has an optimal integral solution. This implies that when $\text{LP}(\mathbf{z}^*)$ is of the form (5.1), the MILP problem must have an optimal integral solution. Note that in particular, the MILP problem must have an optimal integral solution if for each feasible vector \mathbf{z} (each vector \mathbf{z} for which $\text{LP}(\mathbf{z})$ has a feasible solution) the corresponding LP problem $\text{LP}(\mathbf{z})$ can be written in the form (5.1).

5.3 Notation

In Chapter 4 we optimized the following real-valued design variables: the reciprocal of the period duration \mathbf{T}' and the fractions $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$, where A is the set of

arcs of the constraint graph $G = (V, A)$. Recall that such a fraction $\gamma(\varepsilon_1, \varepsilon_2)$ equals the time (as a fraction of the period duration) between an occurrence of the periodically recurring event ε_1 and an occurrence of the periodic event ε_2 , where the event ε_1 (ε_2) is either a switch to effective green or a switch to effective red. In this chapter we would like to optimize these variables $\gamma(\varepsilon_1, \varepsilon_2)$ such that the switches to green, yellow and red are scheduled at an integral second. To distinguish between the effective (green and red) modes of a signal group and the indication of a signal group, we introduce some notation (see also Figure 5.3).

Switching times

Recall that $\overline{\mathcal{K}}_i$ is the set of realizations of signal group $i \in \mathcal{S}$ when signal group i receives the maximum number of \overline{K}_i realizations. Define the set of periodic events \mathcal{E} as done in Chapter 4:

$$\mathcal{E} = \{\textcircled{i}_k \mid i \in \mathcal{S}, k \in \overline{\mathcal{K}}_i\} \cup \{\textcircled{\textcircled{i}}_k \mid i \in \mathcal{S}, k \in \overline{\mathcal{K}}_i\}.$$

Let $t^{\text{eff}}(\textcircled{i}_k) \in [0, T)$ and $t^{\text{eff}}(\textcircled{\textcircled{i}}_k) \in [0, T)$ denote the time at which the k th effective green time of signal group i starts respectively ends; note that in Chapter 4, we have denoted the time $t^{\text{eff}}(\varepsilon)$ by $t(\varepsilon)$. Signal group i switches to green l_i^s seconds before it becomes effective green, i.e., green interval $k \in \overline{\mathcal{K}}_i$ of signal group i starts at:

$$t^{\text{ind}}(\textcircled{i}_k) := t^{\text{eff}}(\textcircled{i}_k) - l_i^s \in [-l_i^s, T - l_i^s),$$

where l_i^s is the starting lost time of signal group i . Signal group i switches to red l_i^e seconds after it becomes effective red, i.e., the k th switch to red is scheduled at the time:

$$t^{\text{ind}}(\textcircled{\textcircled{i}}_k) := t^{\text{eff}}(\textcircled{\textcircled{i}}_k) + l_i^e \in [l_i^e, T + l_i^e),$$

where l_i^e is the ending lost time of signal group i . We define $\omega_t(\varepsilon)$ such that $t^{\text{ind}}(\varepsilon) := t^{\text{eff}}(\varepsilon) + \omega_t(\varepsilon)$, i.e.,

$$\begin{aligned} \omega_t(\textcircled{i}_k) &:= -l_i^s, & i \in \mathcal{S}, \quad k \in \overline{\mathcal{K}}_i, \\ \omega_t(\textcircled{\textcircled{i}}_k) &:= l_i^e, & i \in \mathcal{S}, \quad k \in \overline{\mathcal{K}}_i, \end{aligned}$$

Note that, depending on the context, \textcircled{i}_k ($\textcircled{\textcircled{i}}_k$) either refers to a switch to an effective green (effective red) mode or to a switch to a green (red) indication.

Times between events

Define $\chi^{\text{eff}}(\varepsilon_1, \varepsilon_2)$ to equal:

$$\chi^{\text{eff}}(\varepsilon_1, \varepsilon_2) := t^{\text{eff}}(\varepsilon_2) - t^{\text{eff}}(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)/\mathbf{T}',$$

for some integer $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$; in Section 4.2.1 we have used the notation $f(\varepsilon) := t^{\text{eff}}(\varepsilon)\mathbf{T}'$. Note that $\chi^{\text{eff}}(\varepsilon_1, \varepsilon_2)$ is a time between an occurrence of the periodic event ε_1 and an occurrence of the periodic event ε_2 , where ε_1 and ε_2 each denote a switch to effective green or a switch to effective red. Moreover, we define $\gamma^{\text{eff}}(\varepsilon_1, \varepsilon_2) := \chi^{\text{eff}}(\varepsilon_1, \varepsilon_2)\mathbf{T}'$; in Chapter 4, we have denoted this fraction by $\gamma(\varepsilon_1, \varepsilon_2)$. Similarly, we define $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)$ as:

$$\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) := t^{\text{ind}}(\varepsilon_2) - t^{\text{ind}}(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)/\mathbf{T}',$$

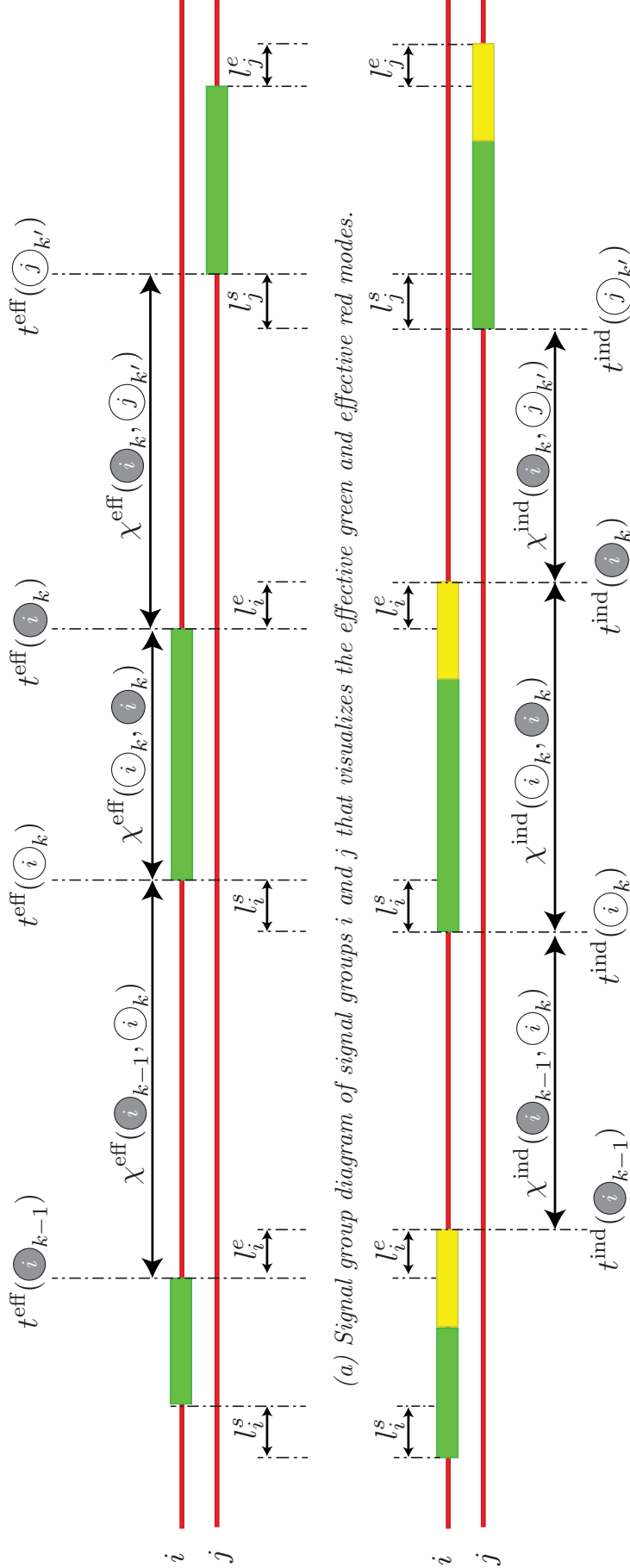
and $\gamma^{\text{ind}}(\varepsilon_1, \varepsilon_2) := \chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)\mathbf{T}'$. Note that $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)$ is a time between an occurrence of the periodic event ε_1 and an occurrence of the periodic event ε_2 , where ε_1 and ε_2 each denote a switch to a green indication or a switch to a red indication. We have the following relation:

$$\begin{aligned} \chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) &:= t^{\text{ind}}(\varepsilon_2) - t^{\text{ind}}(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)/\mathbf{T}', \\ &:= t^{\text{eff}}(\varepsilon_2) + \omega_t(\varepsilon_2) - t^{\text{eff}}(\varepsilon_1) - \omega_t(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)/\mathbf{T}', \\ &:= \chi^{\text{eff}}(\varepsilon_1, \varepsilon_2) + \omega_t(\varepsilon_2) - \omega_t(\varepsilon_1), \\ &:= \chi^{\text{eff}}(\varepsilon_1, \varepsilon_2) + \omega_\chi(\varepsilon_1, \varepsilon_2), \end{aligned}$$

where,

$$\omega_\chi(\varepsilon_1, \varepsilon_2) := \omega_t(\varepsilon_2) - \omega_t(\varepsilon_1).$$

In vector notation, we have $\chi^{\text{ind}} = \chi^{\text{eff}} + \omega_\chi = \gamma^{\text{eff}}/\mathbf{T}' + \omega_\chi$.



(b) Signal group diagram of signal groups i and j that visualizes the indication green, yellow and red indications.

Figure 5.3: Relation of the effective modes of a signal group to the indication of this signal group.

5.4 Input data

In this chapter, we require the same input data as in Chapter 4. As we desire an integral signal group diagram, we assume that the following data is all integral:

- the yellow time Y_i of each signal group $i \in \mathcal{S}$,
- the lower bound \underline{T} and the upper bound \bar{T} on the period duration,
- the minimum green time $\underline{G}_i := \underline{g}_i - Y_i + l_i^s + l_i^e$ and the maximum green time $\bar{G}_i := \bar{g}_i - Y_i + l_i^s + l_i^e$ of each signal group $i \in \mathcal{S}$,
- the minimum red time $\underline{R}_i := \underline{r}_i - l_i^s - l_i^e$ and the maximum red time $\bar{R}_i := \bar{r}_i - l_i^s - l_i^e$ of each signal group $i \in \mathcal{S}$, and
- the minimum clearance time $\underline{C}_{i,j} := \underline{c}_{i,j} - l_i^e - l_j^s$ of each pair of conflicting signal groups $\{i, j\} \in \Psi_{\mathcal{S}}$,

Furthermore, like in Chapter 4, we ensure a well-posed optimization problem, with well-posedness constraints. Such a well-posedness constraint forces the strict positivity of $\chi^{\text{eff}}(\textcircled{i}_k, \textcircled{i}_k) + \chi^{\text{eff}}(\textcircled{i}_k, \textcircled{j}_{k'})$ by forcing it to exceed some small positive value ϵ , which is equivalently to forcing the inequality $\chi^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) + \chi^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_{k'}) \geq \epsilon + l_j^s - l_i^s$. We replace ϵ by $\epsilon_{i,j} > 0$, whose value may be different for each conflict $\{i, j\} \in \Psi_{\mathcal{S}}$. We assume w.l.o.g. that $\epsilon_{i,j} + l_j^s - l_i^s$ is integral.

5.5 Structural property

In the previous chapters of this thesis, we have considered the following stability constraint for each signal group $i \in \mathcal{S}$:

$$0 < \rho_i^{\text{SG}} \leq \sum_{k \in \mathcal{K}_i} \gamma(\textcircled{i}_k, \textcircled{i}_k).$$

This inequality forces the total effective green time g_i of signal group $i \in \mathcal{S}$ to be at least $\rho_i^{\text{SG}}T$ seconds. In this chapter, we force this total effective green time to be strictly larger than $\rho_i^{\text{SG}}T$. We do so, because (assuming a stochastic arrival process) the average delay that road users experience at signal group $i \in \mathcal{S}$ becomes infinite for $g_i = \rho_i^{\text{SG}}T$, see for example the extended version of the approximation of (van den Broek et al., 2006) given in Section 4.2.3.

Consider some integral signal group diagram and let K_i be the number of realizations that signal group $i \in \mathcal{S}$ has for this integral signal group diagram. Furthermore, let G_i denote the total green time of signal group i during one period of this signal group diagram. This total green time G_i is related to the total effective green time g_i of signal group $i \in \mathcal{S}$ as follows:

$$G_i = g_i + K_i(l_i^s + l_i^e - Y_i).$$

For reasons of stability it must hold that:

$$g_i > \rho_i^{\text{SG}} T,$$

where T is the period duration of the integral signal group diagram. Note that the total green time of signal group $i \in \mathcal{S}$ is integral for any integral signal group diagram. Therefore, by combining the above equations, we obtain the following inequality that is satisfied for any integral signal group diagram:

$$G_i \geq \lfloor \rho_i^{\text{SG}} T + K_i(l_i^s + l_i^e - Y_i) \rfloor + 1.$$

The above inequality implies that for each integral signal group diagram, the total green time of signal group i exceeds (or equals) the smallest integral duration for which signal group i is stable. Let $\text{Prop}(\beta)$ denote the following structural property:

Prop(β). *Consider a signal group diagram with a period duration of T seconds. Let G_i be the total green time of signal group $i \in \mathcal{S}$ during one period of T seconds. This signal group diagram satisfies $\text{Prop}(\beta)$ if its period duration T is integral and for each signal group $i \in \mathcal{S}$ it holds that:*

$$G_i \geq \lfloor \beta \rho_i^{\text{SG}} T + K_i(l_i^s + l_i^e - Y_i) \rfloor + 1.$$

Note that when an integral signal group diagram satisfies $\text{Prop}(\beta)$, this signal group diagram can handle a growth factor β of the arrival rates. Moreover, note that any feasible integral signal group diagram satisfies $\text{Prop}(1)$. We use this property $\text{Prop}(\beta)$ in the next section. That section concerns the first step of the two-step approach.

5.6 First step of the two-step approach

In this section we consider the first step of the two-step approach that searches for the optimal integral signal group diagram. The goal of this first step is not to find an integral signal group diagram yet, but to find a signal group diagram that can easily be rounded in the second step. We consider three different objective functions: minimization of the period duration (Section 5.6.1), maximization of the capacity of the intersection (Section 5.6.2), and minimization of the average (weighted) delay that road users experience (Section 5.6.3). For each of these objective functions we give a two-step approach to optimize integral signal group diagrams. In this section we elaborate on the first step of this two-step approach. When minimizing the period duration, during the first step we search for the optimal signal group diagram amongst the signal group diagrams that satisfy $\text{Prop}(1)$. When maximizing the capacity of the intersection, we search for the

largest growth factor β for which some signal group diagram exists that satisfies $\text{Prop}(\beta)$. Moreover, when minimizing the average (weighted) delay that road users experience at the intersection, we search for the optimal signal group diagram amongst the signal group diagrams that satisfy $\text{Prop}(1)$. Subsequently, during the second step, we solve a (relatively simple) MILP problem to obtain an integral signal group diagram from the signal group diagram returned by the first step. We consider this second step in Section 5.7. Thereupon, in Section 5.8 we perform a large numerical study. For all its test cases we indeed find an integral signal group diagram with the proposed two-step approach.

5.6.1 Minimizing the period duration

In this section we consider the minimization of the period duration. First, we elaborate on an algorithm to find the signal group diagram with the smallest period duration amongst the signal group diagrams that satisfy the structural property $\text{Prop}(1)$. Subsequently, we prove that this algorithm either finds the signal group diagram with the smallest period duration amongst the signal group diagrams that satisfy $\text{Prop}(1)$ or it detects that no signal group diagram exists that satisfies $\text{Prop}(1)$.

Optimization procedure

The first step is to minimize the period duration T by solving MILP problem (4.8) with the objective function:

$$J(\mathbf{T}', \boldsymbol{\gamma}^{\text{eff}}, \mathbf{z}, \mathbf{b}) := -\mathbf{T}'. \quad (5.2)$$

This MILP problem optimizes over all the signal group diagrams (also the signal group diagrams that do not satisfy $\text{Prop}(1)$). Therefore, the resulting signal group diagram might not satisfy $\text{Prop}(1)$. However, this MILP problem gives a lower bound on the period duration of any signal group diagram satisfying $\text{Prop}(1)$. In the next step we modify the MILP problem to obtain a better lower bound on the period duration. We repeat that process to iteratively obtain a better lower bound. Eventually, this lower bound equals the smallest period duration of any signal group diagram satisfying $\text{Prop}(1)$.

Updating the lower bound on the period duration Let the period duration of the returned signal group diagram be T seconds. Any signal group diagram satisfying $\text{Prop}(1)$ has an integral period duration. Therefore, each such signal group diagram must have a period duration of at least $\lceil T \rceil$ seconds. Hence, we can update the lower bound on the period duration as follows:

$$\underline{T} := \lceil T \rceil. \quad (5.3)$$

A lower bound on the total effective green time Consider any signal group diagram satisfying $\text{Prop}(1)$. Let T be its period duration and let K_i be the number of

realizations that signal group $i \in \mathcal{S}$ has for this signal group diagram. The total green time G_i of signal group $i \in \mathcal{S}$ exceeds or equals $\lfloor \rho_i^{\text{SG}} T + K_i(l_i^s + l_i^e - Y_i) \rfloor + 1$ seconds for this signal group diagram (see the definition of Prop(1)). Therefore, the total effective green time $g_i := G_i - K_i(l_i^s + l_i^e - Y_i)$ of signal group $i \in \mathcal{S}$ satisfies the following inequality:

$$g_i \geq \lfloor \rho_i^{\text{SG}} T + K_i(l_i^s + l_i^e - Y_i) \rfloor + 1 - K_i(l_i^s + l_i^e - Y_i).$$

Since the period duration of this signal group diagram must exceed (or equal) the newly obtained lower bound \underline{T} , we have the following inequality for each signal group $i \in \mathcal{S}$:

$$g_i \geq \lfloor \rho_i^{\text{SG}} \underline{T} + K_i(l_i^s + l_i^e - Y_i) \rfloor + 1 - K_i(l_i^s + l_i^e - Y_i).$$

For each signal group $i \in \mathcal{S}$ we include a linear constraint to force the latter inequality. The total effective green time g_i , $i \in \mathcal{S}$ satisfies:

$$g_i := \sum_{k=1}^{\bar{K}_i} \gamma^{\text{eff}}(\odot_k, \bullet_k) / \mathbf{T}',$$

where \bar{K}_i is the maximum number of realizations that signal group i is allowed to have. As a result, the desired inequality can be written as:

$$\sum_{k=1}^{\bar{K}_i} \gamma^{\text{eff}}(\odot_k, \bullet_k) \geq (\lfloor \rho_i^{\text{SG}} \underline{T} + K_i(l_i^s + l_i^e - Y_i) \rfloor + 1 - K_i(l_i^s + l_i^e - Y_i)) \mathbf{T}'. \quad (5.4)$$

For each signal group $i \in \mathcal{S}$ with $\underline{K}_i = \bar{K}_i$ we have $K_i := \underline{K}_i$ and, as a consequence, the above inequality is linear for these signal groups; we include this linear constraint for each signal group $i \in \mathcal{S}$ with $\underline{K}_i = \bar{K}_i$. Consider a signal group $i \in \mathcal{S}$ for which the number of realizations is optimized, i.e., a signal group $i \in \mathcal{S}$ for which $\underline{K}_i < \bar{K}_i$. As we also optimize K_i , the above constraint is not linear for such a signal group. However, using Assumption 5.1, this constraint reduces to the following linear constraint:

$$\sum_{k=1}^{\bar{K}_i} \gamma^{\text{eff}}(\odot_k, \bullet_k) \geq (\lfloor \rho_i^{\text{SG}} \underline{T} \rfloor + 1) \mathbf{T}'. \quad (5.5)$$

We include this constraint for each signal group $i \in \mathcal{S}$ with $\underline{K}_i < \bar{K}_i$.

Obtaining a new lower bound on the period duration With the formulated MILP problem we can obtain a new lower bound on the period duration of any signal group diagram satisfying Prop(1). To this end, we solve the formulated MILP problem with the objective of minimizing (5.2). The algorithm terminates if this MILP problem is infeasible; then no signal group diagram exists that satisfies Prop(1). If the resulting signal group diagram satisfies Prop(1), the algorithm terminates as well; we have then found the optimal signal group diagram amongst the signal group diagrams satisfying Prop(1). Assume that the algorithm did not (yet) terminate. We then perform another

iteration. We again update the lower bound on the period duration (5.3), and use this new lower bound \underline{T} to update linear constraint (5.4) for each signal group $i \in \mathcal{S}$ with $\underline{K}_i = \overline{K}_i$, and update linear constraint (5.5) for each signal group $i \in \mathcal{S}$ with $\underline{K}_i < \overline{K}_i$. Subsequently, we solve the resulting MILP problem with the objective of minimizing (5.2). This process repeats until either infeasibility is detected or the optimal signal group diagram is found.

Assume that the algorithm terminates because the optimal signal group diagram is found. Let the returned optimal signal group diagram have a period duration of T seconds. If this signal group diagram is not already integral, we solve the rounding MILP problem that we will propose in Section 5.7. If this rounding MILP problem is feasible, it returns an integral signal group diagram with a period duration of T seconds. This integral signal group diagram then indeed minimizes the period duration amongst all integral signal group diagrams.

We have formulated the proposed optimization procedure in Algorithm 1. We prove that this algorithm either detects infeasibility or it finds the optimal signal group diagram amongst the ones that satisfy Prop(1). Furthermore, in Section 5.8.1 we test this algorithm for a large numerical test set; for all test cases we are able to find the optimal integral signal group diagram that minimizes the period duration.

Algorithm 1 The first step of the two-step approach that searches for the integral signal group diagram with the smallest period duration.

- 1: **Initialize** For each signal group $i \in \mathcal{S}$ that satisfies $\underline{K}_i = \overline{K}_i$, add (5.4) with $K_i := \underline{K}_i$ to MILP problem (4.8). For all other signal groups add (5.5) to MILP problem (4.8).
 - 2: **while** true **do**
 - 3: Solve MILP problem (4.8) with $J(\mathbf{T}', \boldsymbol{\gamma}, \mathbf{z}, \mathbf{b}) := -\mathbf{T}'$.
 - 4: **if** the MILP problem is infeasible **then**
 - 5: **Terminate:** no signal group diagram exists that satisfies Prop(1).
 - 6: **else if** the obtained signal group diagram satisfies Prop(1) **then**
 - 7: **Terminate:** we have found the optimal signal group diagram amongst the signal group diagrams satisfying Prop(1).
 - 8: **else**
 - 9: Set $\underline{T} := \lceil T \rceil$.
 - 10: Update the linear constraints (5.4)–(5.5) using the newly obtained lower bound \underline{T} .
-

Remark 5.1. For an integral signal group diagram it holds that $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) := \gamma^{\text{ind}}(\varepsilon_1, \varepsilon_2)/\mathbf{T}'$ is integral for each arc $(\varepsilon_1, \varepsilon_2) \in A$. Therefore, it might seem natural to make $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$ integral-valued design variables. However, this is not possible in a linear optimization problem as we simultaneously optimize the period duration and the signal timings and, therefore, need to formulate the optimization problem by using the real-valued design variables \mathbf{T}' and $\boldsymbol{\gamma}^{\text{eff}}(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$; otherwise the optimization problem becomes non-linear.

Note that the linear optimization problem can also not be formulated using the real-valued design variables \mathbf{T}' and $\gamma^{\text{ind}}(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$ as (5.5) is not linear in these variables.

Refinement of Algorithm 1

During each iteration of Algorithm 1 we need to solve a MILP problem, which is relatively time consuming. To speed up Algorithm 1 we refine this algorithm as follows. Consider the signal group diagram found during iteration k . Let $\mathbf{z}(\mathbf{b})$ be the vector that contains the values of the integral-valued (binary-valued) design variables associated with this signal group diagram. Before we solve the MILP problem (on line 3) during iteration $k + 1$, we can first set the integral-valued (binary-valued) design variables to the values in $\mathbf{z}(\mathbf{b})$ and, subsequently, solve a linear programming problem (instead of a MILP problem) minimizing the period duration. If the result is a signal group diagram with a period duration that equals the current lower bound \underline{T} , the algorithm terminates; we have then found a signal group diagram that minimizes the period duration amongst the ones that satisfy Prop(1). Otherwise, we solve the MILP problem.

Optimality

In this section we prove that Algorithm 1 either finds the optimal signal group diagram or it detects infeasibility:

Lemma 5.4. *Algorithm 1 terminates in a finite number of iterations and either returns the signal group diagram with the smallest period duration amongst the signal group diagrams satisfying Prop(1) or it detects that no signal group diagram exists that satisfies Prop(1).*

Proof. See Appendix E.1.1 for the proof. □

This concludes the first step of the two-step approach that searches for the integral signal group diagram with the smallest period duration. In the next section we consider the first step of the two-step approach that searches for the integral signal group diagram that can handle the largest growth factor β of the arrival rates.

5.6.2 Maximizing the capacity

In this section we consider the maximization of the capacity of the intersection. We use a two-step approach to search for the integral signal group diagram that can handle the largest growth factor of the arrival rates (loads). During the first step of this two-step approach, we search for the maximum growth factor β^{\max} for which a signal group diagram exists that satisfies Prop(β^{\max}).

A bisection algorithm

Consider MIP problem (4.8). We can obtain the largest growth factor of the arrival rates that is sustainable for some signal group diagram by multiplying the left-hand side of stability constraints (4.8g) by β and, subsequently, maximizing the growth factor β .

Let $\beta_{\mathbb{R}}^{\max}$ be the resulting growth factor. Define $\beta_{\mathbb{Z}}^{\max}$ to be the largest growth factor for which a signal group diagram exists that satisfies $\text{Prop}(\beta_{\mathbb{Z}}^{\max})$. Trivially $\beta_{\mathbb{Z}}^{\max} \leq \beta_{\mathbb{R}}^{\max}$ and, therefore, we can obtain an upper bound ($\beta_{\mathbb{R}}^{\max}$) on the value of $\beta_{\mathbb{Z}}^{\max}$ by solving a MILP problem. This MILP problem can also be used to detect infeasibility. When this MILP problem is infeasible, for all growth factors β no feasible signal group diagram exists for which a growth factor β is sustainable and, therefore, for all growth factors β also no feasible signal group diagram exists that satisfies $\text{Prop}(\beta)$.

Assume that for some growth factor β a signal group diagram exists that satisfies $\text{Prop}(\beta)$. Note that any signal group diagram that satisfies $\text{Prop}(\beta_1)$ also satisfies $\text{Prop}(\beta_2)$, $\beta_2 \leq \beta_1$. Therefore, we can find the maximum growth factor $\beta_{\mathbb{Z}}^{\max}$ via bisection. First we obtain an upper bound $ub := \beta_{\mathbb{R}}^{\max}$ by solving the aforementioned MILP problem. Furthermore, we set the lower bound $lb := 0$. With Algorithm 1 we can check if a signal group diagram exists that satisfies $\text{Prop}(m)$, where $m := (ub + lb)/2$ (we then simply scale all loads ρ_q , $q \in \mathcal{Q}$ with a factor m before running Algorithm 1); Algorithm 1 either detects infeasibility (no signal group diagram exists that satisfies $\text{Prop}(m)$) or it returns the signal group diagram with the smallest period duration amongst the signal group diagrams that satisfy $\text{Prop}(m)$. If infeasibility is detected, we update $ub := m$ and otherwise we update $lb := m$. This process repeats until $ub - lb$ is small enough, i.e., until $ub - lb \leq \delta$; the lower bound lb is then the largest growth factor (up to arbitrary precision δ) for which a signal group diagram exists that satisfies $\text{Prop}(\beta)$. We visualize this algorithm in Figure 5.4a.

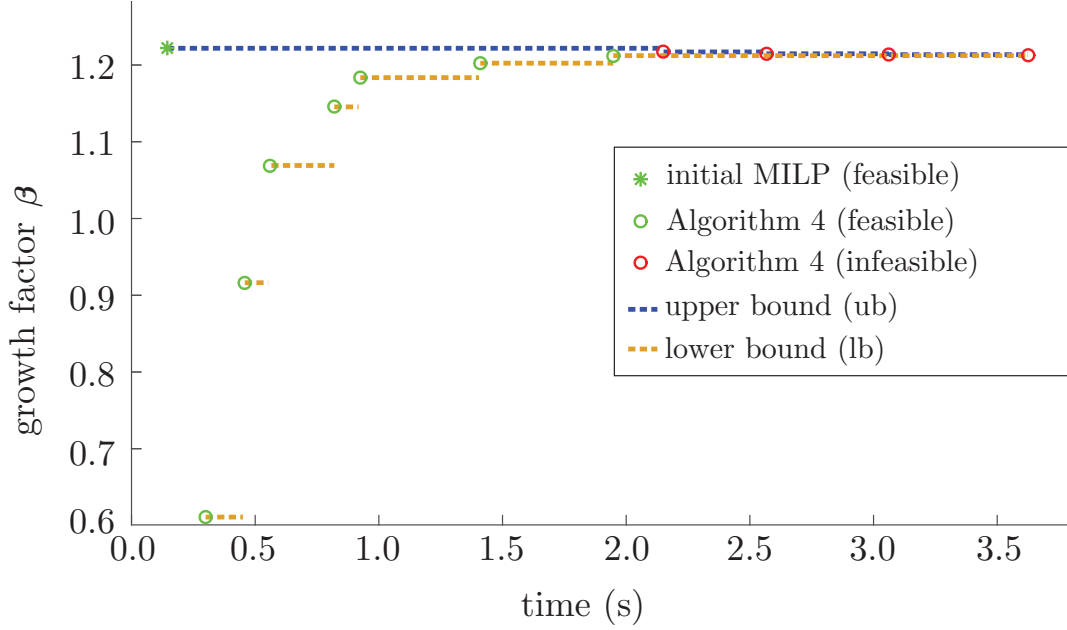
Refinement of the bisection algorithm

During each iteration of the bisection algorithm we need to solve at least one MILP problem, which might be time consuming. Therefore, we refine this algorithm as follows, see also Figure 5.4b. By solving the initial MILP problem, which gave us the upper bound $\beta_{\mathbb{R}}^{\max}$, we have also found a signal group diagram that can handle a growth factor of $\beta_{\mathbb{R}}^{\max}$. Let \mathbf{z} (\mathbf{b}) be the associated vector containing the values of the integral-valued (binary-valued) design variables. We can set the integral-valued (binary-valued) design variables to the values in \mathbf{z} (\mathbf{b}) and, subsequently, perform the bisection algorithm that is proposed in the previous section. The result is the signal group diagram that can handle the largest growth factor β (up to arbitrary precision) amongst the signal group diagrams satisfying $\text{Prop}(\beta)$ and for which the integral-valued design variables (binary-valued design variables) are equal to the values in \mathbf{z} (\mathbf{b}). As a result, we have obtained a better lower bound lb by solving several linear programming problems. Subsequently, we can check if a signal group diagram exists that satisfies $\text{Prop}(lb + \delta)$ by running Algorithm 1. If such a signal group diagram does not exist, we can update the upper bound to $ub := lb + \delta$ and the algorithm terminates. If such a signal group diagram does exist, we can again find a better lower bound lb via bisection by fixing the integral-valued and binary-valued design variables (to the values associated with the newly found signal

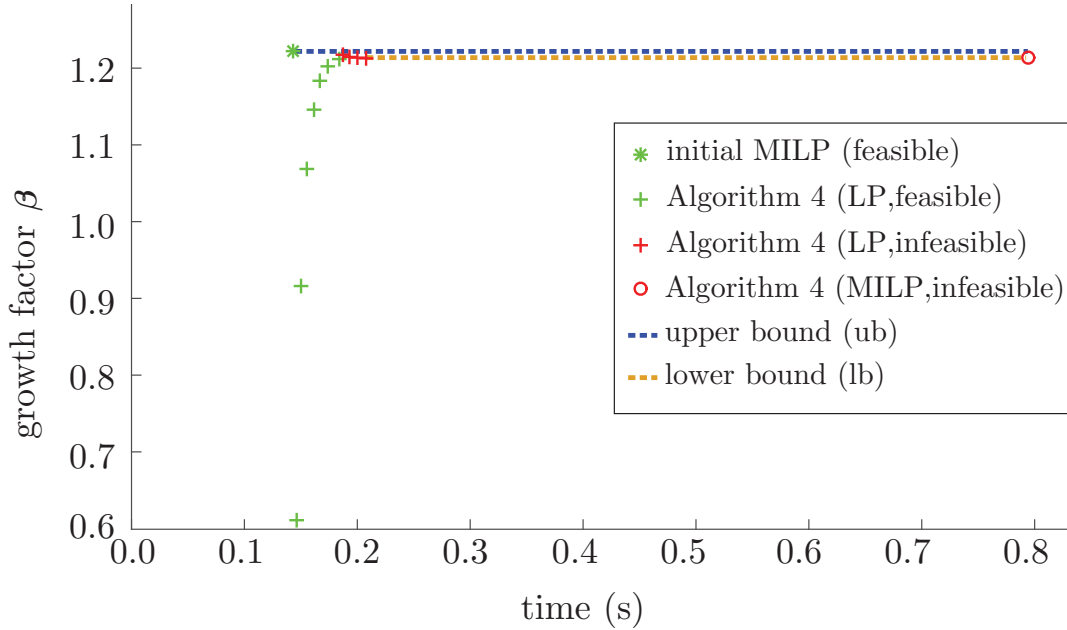
group diagram). Subsequently, we can again check if a signal group diagram exists that satisfies $\text{Prop}(lb + \delta)$. This process repeats itself until $ub - lb \leq \delta$. This refined algorithm often needs to solve less MILP problems; instead it solves several linear programming problems, which can be done efficiently.

Let the result of this first step be a signal group diagram that can handle a growth factor $\beta_{\mathbb{Z}}^{\max}$. If the result of this first step is not yet an integral signal group diagram, we solve the rounding MILP problem that will be proposed in Section 5.7. If this rounding MILP problem is feasible, it returns an integral signal group diagram that can also handle a growth factor $\beta_{\mathbb{Z}}^{\max}$ of the arrival rates. This integral signal group diagram indeed maximizes the capacity of the intersection (up to precision δ) amongst all integral signal group diagrams.

Remark 5.2. *With the initial MILP problem we find an upper bound ub . This initial MILP problem can also be used to detect infeasibility; when this MILP problem is infeasible, for all growth factors β no feasible signal group diagram exists that satisfies $\text{Prop}(\beta)$. However, this is not an 'if and only if' relation, i.e., whenever the initial MILP problem is feasible then it is still possible that for all growth factors β no feasible signal group diagram exists that satisfies $\text{Prop}(\beta)$. Note that the refined algorithm only needs to solve two MILP problems (and several LP problems) to detect that for all growth factors $\beta \geq \delta$ no signal group diagram exists that satisfies $\text{Prop}(\beta)$.*



(a) Original bisection algorithm. This bisection algorithm first solves an initial MILP problem to find an upper bound on the maximum growth factor $\beta_{\mathbb{Z}}^{\max}$. Subsequently, it uses Algorithm 1 to check if some signal group diagram exists that satisfies $\text{Prop}((ub + lb)/2)$. If such a signal group diagram exists, the lower bound lb is updated. Otherwise the upper bound ub is updated. This bisection algorithm continues until $ub - lb \leq \delta$.



(b) A refined bisection algorithm. The initial MILP problem gives a feasible signal group diagram. Subsequently, the integral-valued (and the binary-valued) design variables are set to the values associated with this signal group diagram and we iteratively use Algorithm 1 to obtain a better lower bound (which is obtained at time ~ 0.25 seconds); we denote this version of Algorithm 1 by: Algorithm 1 (LP). Subsequently, we also optimize the integral-valued (and the binary-valued) design variables; we use Algorithm 1 to check if a signal group diagram exists that satisfies $\text{Prop}(lb + \delta)$; we denote this version of Algorithm 1 by: Algorithm 1 (MILP). Such a signal group diagram does not exist and the algorithm terminates at time ~ 0.8 seconds.

Figure 5.4: Two bisection algorithms that are applied to the intersection L1 from (Fleuren and Lefeber, 2016a) with $\underline{K}_i = \overline{K}_i = 1$ for each signal group $i \in \mathcal{S}$. These bisection algorithms find the maximum growth factor $\beta_{\mathbb{Z}}^{\max}$ for which a signal group exists that satisfies $\text{Prop}(\beta_{\mathbb{Z}}^{\max})$.

5.6.3 Minimizing the average (weighted) delay

This section concerns the minimization of the average (weighted) delay that road users experience at the intersection. We use a two-step approach to search for the optimal integral signal group diagram. During the first step of this two-step approach we search for the optimal signal group diagram amongst the signal group diagrams satisfying Prop(1) (and (4.5)). To this end, we approximate the delay that road users experience at a queue under pre-timed control with an approximate formula. In this chapter we may use the same formulae as the ones used in Chapter 4, e.g., the (extended) version of the formula of (van den Broek et al., 2006). Recall that this approximate formula assumes that (4.5) is satisfied. Therefore, we force these inequalities by including the linear constraints (4.5) in every MILP (and every LP) that we solve. We approximate the delay that results from the delay formula by piecewise linear functions. This piecewise linear approximation is exact for each integral signal group diagram.

First, we consider the period duration to be fixed. We can then find the optimal signal group diagram by solving a single MILP problem. Thereupon, we consider the period duration to be a design variable that may be chosen between \underline{T} and \overline{T} . We can then find the optimal signal group diagram by solving one MILP problem for each integral period duration $T \in [\underline{T}, \overline{T}]$. This may be very cumbersome as the interval $[\underline{T}, \overline{T}]$ may be very large. Therefore, we also give a heuristic approach to optimize over the period duration. This heuristic approach does not necessarily return the optimal signal group diagram amongst the signal group diagrams satisfying Prop(1). However, it is much faster and its results are promising, see also Section 5.8.3.

Fixed period duration

In this section we consider the period duration $T := 1/\mathbf{T}'$ to be fixed. Note that each signal group diagram satisfying Prop(1) has an integral period duration. Hence, we fix the period duration T to an integral value. We formulate a MILP problem to find the optimal signal group diagram with a period duration of T seconds amongst the signal group diagrams satisfying Prop(1) (and (4.5)). To this end, we approximate the average (weighted) delay that road users experience at the intersection by using piecewise linear functions. Moreover, for each signal group $i \in \mathcal{S}$ we add a constraint to this optimization problem so that we only optimize over the signal group diagrams that satisfy Prop(1).

Objective function Let d_q be the approximated delay at queue $q \in \mathcal{Q}$. Recall that we can write the average (weighted) delay D as follows:

$$D = \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q = \sum_{i \in \mathcal{S}} d_i, \quad (5.6)$$

where $d_i := \sum_{q \in \mathcal{Q}_i} w_q d_q$ is the contribution of signal group i to the average (weighted) delay. In this section we approximate the average weighted delay D that road users

experience by a piecewise linear approximation D^{pw1} . This piecewise linear approximation is very similar to the one defined in Section 4.4. The difference is that we let all the breakpoints of this piecewise linear approximation correspond to the integral red times (instead of the integral effective red times).

Remark 5.3. *Special attention needs to be paid to the linear approximation of the stochastic delay term d_i^{stoch} . When $\underline{K}_i = \overline{K}_i$, we can easily find all the integral total red times R_i that are (possibly) feasible from the relation $r_i := R_i + K_i(l_i^s + l_i^e - Y_i)$ (and the interval (4.11) for r_i). However, this is not possible when $\underline{K}_i > \overline{K}_i$ as the value for K_i is then determined through optimization. From Assumption 5.1 and the integrality of the yellow time Y_i it follows that $(l_i^s + l_i^e - Y_i)$ is integral when $\underline{K}_i > \overline{K}_i$. As a consequence, the red time R_i is then integral if and only if r_i is integral and, as a result, we can approximate the stochastic delay term d_i^{stoch} as was done in Section 4.4.*

Additional constraints In this section we add a constraint to the optimization problem for each signal group $i \in \mathcal{S}$. As a result of these additional constraints, we only optimize over the signal group diagrams satisfying Prop(1). Consider a signal group diagram that satisfies Prop(1). Let T be its period duration. Moreover, denote the number of realizations that signal group $i \in \mathcal{S}$ has for this signal group diagram by K_i . Since the signal group diagram satisfies Prop(1), the total effective green time $g_i := G_i - K_i(l_i^s + l_i^e - Y_i)$ of each signal group $i \in \mathcal{S}$ satisfies the following inequality:

$$g_i \geq \lfloor \rho_i^{\text{SG}} T + K_i(l_i^s + l_i^e - Y_i) \rfloor + 1 - K_i(l_i^s + l_i^e - Y_i).$$

For each signal group $i \in \mathcal{S}$, we include an inequality constraint to force the latter inequality, see also Section 5.6.1. For each signal group $i \in \mathcal{S}$ with $\underline{K}_i = \overline{K}_i$ we include the following linear constraint:

$$\sum_{k=1}^{\overline{K}_i} \gamma^{\text{eff}}(\odot_k, \bullet_k) T \geq \lfloor \rho_i^{\text{SG}} T + \underline{K}_i(l_i^s + l_i^e - Y_i) \rfloor + 1 - \underline{K}_i(l_i^s + l_i^e - Y_i),$$

and for each signal group $i \in \mathcal{S}$ with $\underline{K}_i < \overline{K}_i$ we include the constraint:

$$\sum_{k=1}^{\overline{K}_i} \gamma^{\text{eff}}(\odot_k, \bullet_k) T \geq \lfloor \rho_i^{\text{SG}} T \rfloor + 1.$$

These additional linear constraints ensure that we only optimize over the signal group diagrams satisfying Prop(1).

We can obtain the desired signal group diagram with a period duration of T seconds by solving the formulated optimization problem with the objective of minimizing the approximated delay D^{pw1} . In the next section we use this formulated optimization problem to find the optimal signal group diagram when we may also choose the period duration T .

Variable period duration

In the previous section we have fixed the period duration and we have given a MILP formulation to find the optimal signal group diagram amongst the signal group diagrams satisfying Prop(1). In this section we consider the period duration T to be a design variable that may be chosen between its lower bound \underline{T} and its upper bound \overline{T} . We can find the optimal signal group diagram simply by solving the formulated MILP problem for each integral period duration in the interval $[\underline{T}, \overline{T}]$. However, this interval can be large. Hence, we propose a heuristic approach to optimize over the period duration. This heuristic approach roughly works as follows; we fill in the details later. The approach alternates between fixing the period duration (and optimizing the fractions γ^{eff} , the integral-valued design variables \mathbf{z} and the binary-valued design variables \mathbf{b}) and fixing the integral-valued and binary-valued design variables (and optimizing the fractions γ^{eff} and the reciprocal of the period duration \mathbf{T}'). To this end, we require an initial period duration T_0 . We find this initial period duration with Algorithm 1; recall that this algorithm finds the smallest period duration for which a signal group diagram exists that satisfies Prop(1) (and in this section also satisfying the inequalities (4.5)). Subsequently, we find the optimal signal group diagram with a period duration of T_0 seconds amongst the signal group diagrams that satisfy Prop(1). To this end, we solve the MILP problem proposed in the previous section. Thereupon, we fix the integral-valued and binary-valued design variables to the values associated with this optimal signal group diagram and, subsequently, we optimize over the period duration; we elaborate on this step later. Let the resulting period duration be T_1 seconds. We again fix the period duration, this time to the value T_1 , and solve a MILP problem. The result is the optimal signal group diagram with a period duration of T_1 seconds that satisfies Prop(1). Subsequently, we again fix the integral-valued and binary-valued design variables to the values associated with this optimal signal group diagram and, subsequently, we optimize over the period duration. This process repeats until (in one of these two alternating steps) we are not able to improve upon the best signal group diagram that we have found so far. Below we elaborate on the step during which we optimize over the period duration. Thereupon, we give an example.

Optimization over the period duration What remains in the specification of the heuristic approach is how to optimize over the period duration. When we optimize over the period duration, we consider the integral-valued design variables and the binary-valued design variables to be fixed. Let these integral-valued design variables and binary-valued design variables be fixed to the values in \mathbf{z} respectively \mathbf{b} . Define $D_{\mathbf{z}, \mathbf{b}}^{\text{pw1}}(T)$ to be the minimum value for D^{pw1} of any feasible signal group diagram for which:

- the period duration equals T seconds,
- the integral-valued and binary-valued design variables are equal to \mathbf{z} respectively \mathbf{b} ,
and

- Prop(1) (and (4.5)) is satisfied.

The goal of the step that optimizes over the period duration is to find the period duration that minimizes $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$. For each given period duration, the value for $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$ can be obtained by solving a linear programming problem. This linear programming problem is obtained from the mixed-integer linear programming problem formulated in the previous section (the section that considers the period duration to be fixed) by setting the integral-valued and binary-valued design variables to the values in \mathbf{z} respectively \mathbf{b} .

Before we search for the period duration that minimizes $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$, we first consider the function $D_{\mathbf{z},\mathbf{b}}(T)$. Define $D_{\mathbf{z},\mathbf{b}}(T)$ to be the smallest value for the average (weighted) delay D associated with any (feasible) signal group diagram for which:

- the period duration equals T seconds,
- the integral-valued and binary-valued design variables are equal to \mathbf{z} respectively \mathbf{b} , and
- linear inequalities (4.5) are satisfied.

The value for $D_{\mathbf{z},\mathbf{b}}(T)$ can be obtained by solving MIP problem (4.8) with the objective of minimizing the convex function $D = \sum_{i \in \mathcal{S}} d_i$; for this MIP problem we fix the period duration to T seconds and we fix the integral-valued design variables and the binary-valued design variables to the values in \mathbf{z} respectively \mathbf{b} . In Figure 5.5 and Figure 5.6 we visualize $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$ and $D_{\mathbf{z},\mathbf{b}}(T)$ as a function of the period duration T for different examples. Note that the function $D_{\mathbf{z},\mathbf{b}}(T)$ seems to have a very nice form. In the following lemma we prove that any local minimum of the function $D_{\mathbf{z},\mathbf{b}}(T)$ is also a global minimum:

Lemma 5.5. *Let the delay of any queue $q \in \mathcal{Q}$ be approximated with a function that is convex in the variables γ^{eff} and \mathbf{T}' , e.g., the delay of each queue $q \in \mathcal{Q}$ is approximated with the (extended version of the) approximation of (van den Broek et al., 2006). Then any local minimum of the function $D_{\mathbf{z},\mathbf{b}}(T)$ is also a global minimum.*

Proof. See Appendix E.1.2 for the proof. □

Lemma 5.5 states that any local minimum of the function $D_{\mathbf{z},\mathbf{b}}(T)$ is also a global minimum. The delay $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$ also roughly follows the shape of the function $D_{\mathbf{z},\mathbf{b}}(T)$, see for example the figures 5.5–5.6. Assume for the moment that any local minimum of $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$ is also a global minimum. We can then find a global minimum as follows. For some period duration we have already obtained the value for $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$. This value was obtained during the previous step of the algorithm; during this step we fixed the period duration and optimized the values of the integral-valued and binary-valued design variables (and the variables γ^{eff}). Let T^* be the period duration for which we have already obtained the value $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^*)$. First, we search for a better period duration to the left of T^* . To this end, we sequentially obtain the following values:

$$D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* - 1), D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* - 2), D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* - 3), \dots \quad (5.7)$$

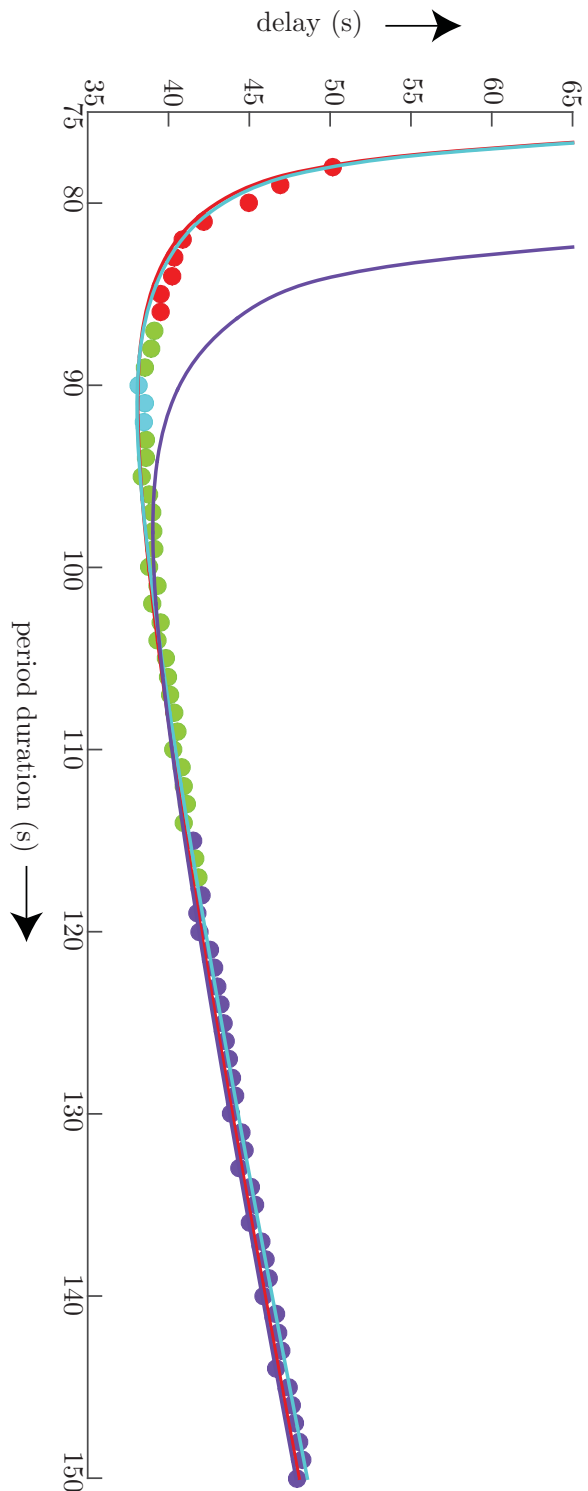
We can obtain each of these values by solving a linear programming problem. We continue to solve these linear programming problems until the objective function increases, i.e., until $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* - (k + 1)) > D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* - k)$ for some $k \geq 0$. Thereupon, we search for a better period duration to the right of T^* . To this end, we sequentially obtain the following values:

$$D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* + 1), D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* + 2), D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* + 3), \dots \quad (5.8)$$

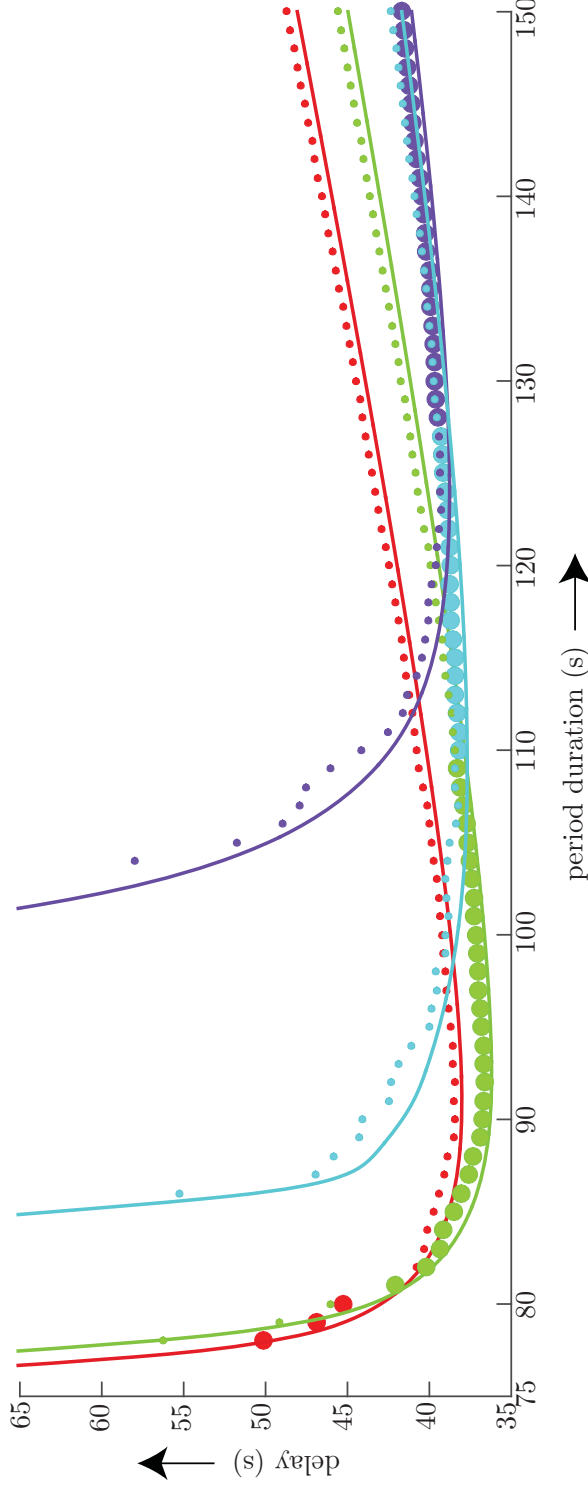
We can obtain each of these values by solving a linear programming problem. We again continue to solve these linear programming problems until the objective function increases, i.e., until $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* + k + 1) > D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T^* + k)$ for some $k \geq 0$. The smallest value for $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$ that we have obtained so far is a global minimum of the function $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$.

Unfortunately, the function $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$ might have multiple local minima. Therefore, we suggest a slightly different termination criterium. To this end, we store the smallest value of the function $D_{\mathbf{z},\mathbf{b}}^{\text{pwl}}(T)$ found so far. We first search for a better period duration to the left of the period duration T^* , i.e., we subsequently obtain the values (5.7). This search continues until Q subsequent values exceed the stored value, where Q is an input parameter. Subsequently, we search for a better period duration to the right of T^* . This search also terminates when Q subsequent values exceed the stored value.

This concludes the section on the optimization of the period duration. We have now given the complete specification of the heuristic method. In the next section we apply this heuristic method to the example L1 from (Fleuren and Lefebber, 2016a). For this example we allow the four most heavily loaded signal groups to have an additional realization.



(a) Results for intersection L1 from (Fleuren and Lefebber, 2016a) for which each signal group has one realization. We zoom in on this figure in Figure 5.6. More information is given in the caption on the next page.



(b) Results for intersection L1 from (Fleuren and Lefeber, 2016a) for which each signal group has one realization except for the four most heavily loaded signal groups; these signal groups are allowed to have an additional realization.

Figure 5.5: The minimum (average) delay of the intersection L1 from (Fleuren and Lefeber, 2016a) as a function of the period duration T when we use the approximation of (van den Broek et al., 2006). The circles visualize the function $D_{\mathbf{z}, \mathbf{b}}^{\text{pw}}(T)$ for different values of \mathbf{z} and \mathbf{b} ; each color refers to specific values for the integral-valued and binary-valued design variables, i.e., the values of the integral-valued and binary-valued design variables are the same for the results visualized in the same color. Each continuous line visualizes the function $D_{\mathbf{z}, \mathbf{b}}(T)$. The difference between the small and the large circles is that the large circles visualize (for each fixed period duration) the solution that minimizes the approximated delay D^{pw} amongst all signal group diagrams satisfying Prop(1) (and (4.5)), which does not hold for the small circles.

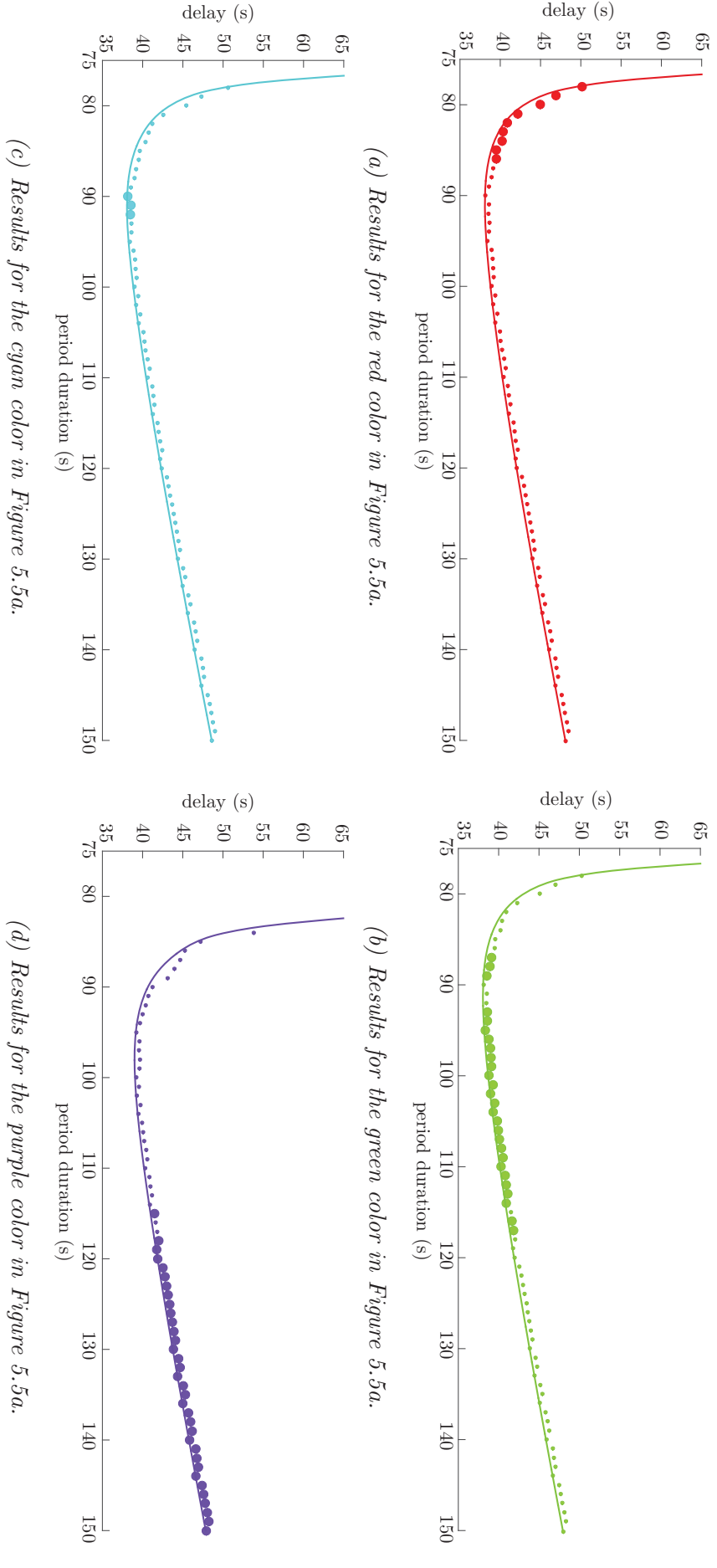


Figure 5.6: The results from Figure 5.5a visualized per color. Each color refers to specific values for the integral-valued and binary-valued design variables, i.e., the values of the integral-valued and binary-valued design variables are the same for the results visualized in the same color. Each continuous line visualizes (as a function of the period duration T) the function $D_{z,b}(T)$, where z and b are specified by the color. The circles in Figure 5.5b (small and large) visualize the function $D_{z,b}^{\text{pw1}}(T)$, where z and b are specified by the color. The difference between the small and the large circles is that the large circles visualize (for each fixed period duration) the solution that minimizes the approximated delay D^{pw1} amongst all signal group diagrams satisfying Prop(1) (and (4.5)), which does not hold for the small circles.

An example

In this section we consider an example: we consider the minimization of the delay for example L1 from (Fleuren and Lefeber, 2016a). We allow the four most heavily loaded signal groups to have an additional realization. All other signal groups have a single realization. For this example we have $Q = 5$ and we use the (extended) approximate formula of (van den Broek et al., 2006) to approximate the delay that road users experience. Recall that the value for Q specifies when to stop optimizing over the period duration.

In Figure 5.5b we have visualized the functions $D_{\mathbf{z},\mathbf{b}}^{\text{pw1}}(T)$ and $D_{\mathbf{z},\mathbf{b}}(T)$ for some relevant vectors \mathbf{z} and \mathbf{b} for this example. In Figure 5.7 we visualize the proposed algorithm. First we compute the initial period duration, which is 78 seconds. Subsequently, at this period duration of 78 seconds we obtain the optimal signal group diagram amongst the ones satisfying Prop(1) (and (4.5)). The result is a delay of ~ 49.9 seconds. Subsequently, we fix the integral-valued and the binary-valued design variables. We first search for a better signal group diagram to the left of the period duration of 78 seconds. This search terminates after the LP problem is found to be infeasible for the period durations 77, 76, 75, 74 and 73. Thereupon we search for a better signal group diagram to the right of the period duration of 78 seconds. The search terminates after we have computed the delay at a period duration of 95 seconds; the delay at the $Q = 5$ subsequent period durations 91, \dots , 95 all exceed the delay at a period duration of 90 seconds. The optimal signal group diagram found so far has a delay of ~ 38.2 seconds. Subsequently we set the period duration to 90 seconds and find the optimal signal group diagram. The associated delay is ~ 36.5 . Thereupon, we again fix the integral-valued and binary-valued design variables and we optimize over the period duration. The improvement is modest; we find a delay of ~ 36.4 for a period duration of 92 seconds. Finally, we set the period duration to 92 seconds and solve a MILP problem; this results again in the same signal group diagram with a delay of ~ 36.4 . As a consequence, the algorithm terminates. For this example, we have solved Algorithm 1 once and, in addition, we have solved 34 LP problems and 3 MILP problems. This is much faster than solving one MILP problem for each integral period duration in $[T, \bar{T}]$.

A disadvantage of this heuristic approach is that it might not find the optimal solution. To increase the probability of finding a global minimum we can run the algorithm from different starting points, i.e., from different initial period durations. For example, when we start the heuristic from n starting points, we can use the following starting points:

$$T_0, \lfloor (n-1)/n T_0 + 1/n \bar{T} \rfloor, \lfloor (n-2)/n T_0 + 2/n \bar{T} \rfloor, \dots, \lfloor 1/n T_0 + (n-1)/n \bar{T} \rfloor, \bar{T}. \quad (5.9)$$

This concludes the first step of the two-step approach that searches for the integral signal group diagram that minimizes the average (weighted) delay that road users experience at the intersection. If the result of this first step is not yet an integral signal group diagram, we solve the rounding MILP problem that will be proposed in Section 5.7; the result of this rounding MILP problem is an integral signal group diagram.

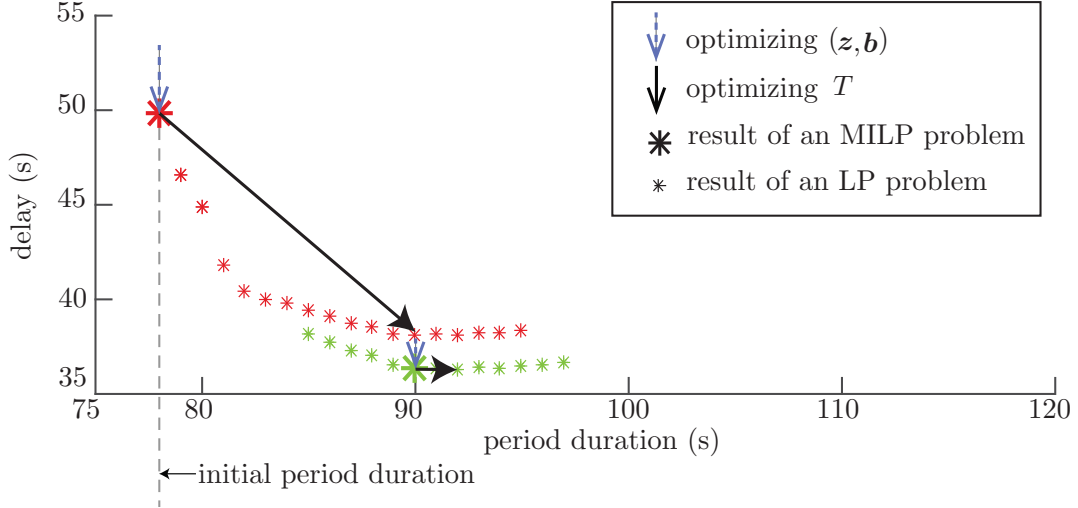


Figure 5.7: Visualization of the algorithm that minimizes the average (weighted) delay amongst the signal group diagrams that satisfy Prop(1). Each color refers to specific values for the integral-valued and binary-valued design variables, i.e., the values of the integral-valued and binary-valued design variables are the same for the results that are visualized in the same color. In this figure we consider the example L1 from (Fleuren and Lefeber, 2016a). We allow the four most heavily loaded signal groups to have an additional realization. All other signal groups have a single realization. For this example we have $Q = 5$ and we use the (extended) approximate formula of (van den Broek et al., 2006) to approximate the delay that road users experience.

Remark 5.4. Note that for some starting points we can abort the computations prematurely. For example, when during the step that optimizes the period duration, the optimal period duration is found to equal T , and we have already found (when starting from another starting point) the optimal signal group diagram with a period duration of T seconds. For this starting point the heuristic method finds a solution that we have already found for some other starting point.

5.7 Second step of the two-step approach: A rounding MILP

In the previous section we have considered the first step of the two-step approach that optimizes integral signal group diagrams. In this section we consider the second step of this two-step approach; we introduce a relatively simple MILP problem, which we call the *rounding MILP problem*. This MILP problem can be used to obtain an integral signal group diagram from a signal group diagram returned by the first step of the two-step approach. This optimization problem searches for a feasible integral signal group diagram for which each green time and each red time (of the not necessarily integral signal group diagram) is rounded upwards or downwards. We formulate this rounding MILP problem in Section 5.7.1. Subsequently, in Section 5.7.2 we prove that this rounding MILP problem indeed returns an integral signal group diagram. In Section 5.7.3 we

prove that this rounding MILP problem is feasible whenever at most one signal group has multiple realizations for the signal group diagram that we try to round. Section 5.7.4 concerns the optimality of the two-step approach. In that section we prove that when minimizing the period duration of the signal group diagram or maximizing the capacity of the intersection, the two-step approach returns the optimal integral signal group diagram when at most one signal group has multiple realizations for the signal group diagram returned by the first step. Furthermore, we prove that this two-step approach returns the optimal integral signal group diagram when minimizing the delay that road users experience and, in addition, each signal group has a single realization for the signal group diagram returned by step one.

5.7.1 A rounding MILP problem

Consider a signal group diagram that is found during step one of the two-step approach. Denote its corresponding solution to the linear constraints of MILP problem (4.8) by $(\tilde{\gamma}^{\text{eff}}, \tilde{T}', \tilde{z}, \tilde{b})$. Let K_i be the number of realizations of signal group i for this solution. Furthermore, define $\mathcal{K}_i := \{1, \dots, K_i\}$ and let Ψ_R be the set of conflicting realizations, i.e.,

$$\Psi_R := \{(i, k), (j, k')\} \mid \{i, j\} \in \Psi_S, k \in \mathcal{K}_i, k' \in \mathcal{K}_j\}.$$

In this section we formulate a MILP problem that can be used to obtain an integral signal group diagram; this MILP problem attempts to find an integral signal group diagram from the solution $(\tilde{\gamma}^{\text{eff}}, \tilde{T}', \tilde{z}, \tilde{b})$ by rounding each green time and each red time up or down. The formulated MILP problem is very similar to MIP problem (4.1), which considers the number of realizations K_i of each signal group to be fixed. In the following sections we introduce the design variables, the constraints and the objective function of the rounding MILP problem.

Design variables

Consider the following constraint graph:

$$\begin{aligned} V &= \{(\textcircled{i})_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\} \cup \{(\textcircled{i})_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}, \\ A &= A_g \cup A_r \cup A_c, \end{aligned}$$

where

$$\begin{aligned} A_g &= \{((\textcircled{i})_k, \textcircled{i})_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}, \\ A_r &= \{((\textcircled{i})_{k-1}, \textcircled{i})_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}, \\ A_c &= \{((\textcircled{i})_k, \textcircled{j})_{k'} \mid \{(i, k), (j, k')\} \in \Psi_R\}. \end{aligned}$$

For each arc $(\varepsilon_1, \varepsilon_2) \in A$ we have the real-valued design variable $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) \in A$. Recall that $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)$ is the time between an occurrence of event ε_1 and an occurrence of event ε_2 , where ε_1 (ε_2) is a switch to a green indication or a switch to a red indication. We

define χ^{ind} as a vector of the real-valued design variables $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$. For each signal group $i \in \mathcal{S}$ we have $K_i - 1$ binary-valued design variables, which we denote by $\mathbf{u}_{i,2}, \dots, \mathbf{u}_{i,K_i}$. The binary-valued design variable $\mathbf{u}_{i,k}$ is used to round the k th red time of signal group i down ($\mathbf{u}_{i,k} = 0$) or up ($\mathbf{u}_{i,k} = 1$). It appears that these binary-valued design variables are the only ones needed for ensuring that the rounding MILP problem returns an integral signal group diagram, i.e., we do not need binary-valued design variables for the signal groups that have one realization. We define \mathbf{u} as a vector of the binary-valued design variables $\mathbf{u}_{i,k}$. The design variables of the rounding MILP problem are comprised of the real-valued design variables in χ^{ind} and the binary-valued design variables in \mathbf{u} . Note that we do not optimize the period duration; the period duration is fixed to $1/\tilde{T}'$ seconds.

Linear constraints

In this section we introduce the linear constraints of the rounding MILP problem. We define $\tilde{\chi}^{\text{ind}} := \tilde{\gamma}^{\text{eff}}/\tilde{T}' + \omega_\chi$, which is the vector χ^{ind} for the signal group diagram that we try to round.

Fix each integral red time For each signal group $i \in \mathcal{S}$ and each realization $k \in \mathcal{K}_i$ for which the red time $\tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k)$ is already integral, we fix this red time to its integral value. Thus, for each realization $k \in \mathcal{K}_i$ of signal group $i \in \mathcal{S}$ with $\tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) \in \mathbb{Z}$, we have the constraints:

$$\chi^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) = \tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k). \quad (5.10a)$$

Round each red time down or up Each red time should be rounded down or rounded up. For the each signal group i we restrict its first red time as follows:

$$\lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \rfloor \leq \chi^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \leq \lceil \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \rceil. \quad (5.10b)$$

Note that the above constraint is redundant if $\tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \in \mathbb{Z}$; the above constraint is then implied by (5.10a). All other red times are forced to be rounded up or rounded down, i.e., for each realization $k = 2, \dots, K_i$ of signal group $i \in \mathcal{S}$ we have the following constraint:

$$\chi^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) = \lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) \rfloor + \mathbf{u}_{i,k}. \quad (5.10c)$$

Note that this constraint is redundant (and can be omitted) when the red time is integral, i.e., this constraint is redundant when $\tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) \in \mathbb{Z}$; this constraint is then implied by (5.10a). In that case we do not need the binary-valued design variable $\mathbf{u}_{i,k}$ for this realization of signal group $i \in \mathcal{S}$.

Rounding each green time up or down Each green time should be rounded down or rounded up. To this end, we include the following constraint for each realization $k \in \mathcal{K}_i$ of signal group $i \in \mathcal{S}$:

$$\lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) \rfloor \leq \chi^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) \leq \lceil \tilde{\chi}^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) \rceil. \quad (5.10d)$$

Lower bound on the total green time of each signal group For each signal group $i \in \mathcal{S}$ we bound the total green time from below:

$$\sum_{k=1}^{K_i} \chi^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) \geq \lfloor \sum_{k=1}^{K_i} \tilde{\chi}^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) \rfloor. \quad (5.10e)$$

Assume that the signal group diagram associated with the solution $(\tilde{\gamma}^{\text{eff}}, \tilde{T}', \tilde{z}, \tilde{b})$ satisfies $\text{Prop}(\beta)$. All signal group diagrams that satisfy (5.10e) then also satisfy $\text{Prop}(\beta)$. To see this, define \tilde{G}_i to be the total green time of signal group $i \in \mathcal{S}$ for the signal group diagram that we try to round. Since this signal group diagram satisfies $\text{Prop}(\beta)$, for each signal group $i \in \mathcal{S}$ we have:

$$\tilde{G}_i \geq \lceil \beta \rho_i^{\text{SG}} T + K_i(l_i^s + l_i^e - Y_i) \rceil.$$

With (5.10e) we force the total green time of signal group $i \in \mathcal{S}$ to exceed $\lfloor \tilde{G}_i \rfloor$ and, as a consequence, it also exceeds the right-hand side of the above inequality. This proves that all signal group diagrams that satisfy (5.10e) also satisfy $\text{Prop}(\beta)$.

Note that the constraint on the total green time (5.10e) is redundant (and can be omitted) for each signal group $i \in \mathcal{S}$ with at most one green time. For these signal groups, constraint (5.10e) is implied by the fact that each green time is rounded down or up. Note that constraint (5.10e) is (indirectly) only a function of the binary-valued design variables $\mathbf{u}_{i,k}$, $k = 2, \dots, K_i$ and the variable $\tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1)$; we can rewrite this constraint to:

$$\tilde{T} - \sum_{k=2}^{K_i} (\lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) \rfloor + \mathbf{u}_{i,k}) - \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \geq \lfloor \sum_{k=1}^{K_i} \tilde{\chi}^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) \rfloor.$$

where $\tilde{T} := 1/\tilde{T}'$. We have obtained this expression by combining (5.10c) with the fact that the green and red intervals of signal group $i \in \mathcal{S}$ together comprise one period.

Minimum clearance times The remaining constraints that we include in the rounding MILP problem are also included in MILP problem (4.1). The first of these constraints is the lower bound on a clearance time, i.e., for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ we have:

$$\underline{C}_{i,j} \leq \chi^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_{k'}). \quad (5.10f)$$

Recall that $\underline{C}_{i,j}$ is (w.l.o.g.) assumed to be integral, see also Section 5.4.

Well-posedness To have a well-posed optimization problem, we restrict the sum of an effective green time plus the subsequent clearance time to be strictly positive. This constraint can be formulated as follows. For each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ we have:

$$\chi^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) + \chi^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_{k'}) \geq \epsilon_{i,j} + \omega_\chi(\textcircled{i}_k, \textcircled{i}_k) + \omega_\chi(\textcircled{i}_k, \textcircled{j}_{k'}). \quad (5.10g)$$

Recall that $\epsilon_{i,j} + \omega_\chi(\textcircled{i}_k, \textcircled{i}_k) + \omega_\chi(\textcircled{i}_k, \textcircled{j}_{k'})$ is (w.l.o.g.) assumed to be integral, see also Section 5.4.

Cycle periodicity constraints The design variables $\chi^{\text{ind}}(\epsilon_1, \epsilon_2)$, $(\epsilon_1, \epsilon_2) \in A$ should satisfy the cycle periodicity constraints, i.e., for some integral cycle basis \mathcal{B} of constraint graph $G = (V, A)$ we have the constraints:

$$\sum_{(\epsilon_1, \epsilon_2) \in \mathcal{C}^+} \chi^{\text{ind}}(\epsilon_1, \epsilon_2) - \sum_{(\epsilon_1, \epsilon_2) \in \mathcal{C}^-} \chi^{\text{ind}}(\epsilon_1, \epsilon_2) = z_{\mathcal{C}} / \tilde{T}', \quad \mathcal{C} \in \mathcal{B}. \quad (5.10h)$$

We set the value of $z_{\mathcal{C}}$ to the integral value associated with the signal group diagram that we try to round, which is the value

$$z_{\mathcal{C}} := \sum_{(\epsilon_1, \epsilon_2) \in \mathcal{C}^+} \tilde{\chi}^{\text{ind}}(\epsilon_1, \epsilon_2) - \sum_{(\epsilon_1, \epsilon_2) \in \mathcal{C}^-} \tilde{\chi}^{\text{ind}}(\epsilon_1, \epsilon_2).$$

Remark 5.5. *It is not necessary to include circuital constraints (4.1i) and (4.1j) in the formulation of this rounding MILP problem; circuital constraint (4.1i) states that two conflicting realizations and the clearance times between these realizations comprise one period and circuital constraint (4.1j) states that the green and red intervals of signal group $i \in \mathcal{S}$ together comprise one period. Both these circuital constraints are satisfied by the signal group diagram that we try to round and, as a consequence, these circuital constraints are implied by the cycle periodicity constraints (5.10h).*

Objective function

Consider the minimization of the period duration of the signal group diagram or the maximization of the capacity of the intersection. It is then sufficient to just find any feasible signal group diagram with the rounding MILP problem; each such integral signal group diagram is optimal. The objective function of the rounding MILP problem is not relevant for these objective functions and, therefore, any linear function of the real-valued design variables suffices.

Consider the minimization of the average (weighted) delay that road users experience. In that case this objective function is relevant; with the rounding MILP problem we then search for the integral signal group diagram that minimizes the average (weighted) delay. This objective function is not linear. Therefore, we approximate this (convex) function with piecewise linear functions. This piecewise linear approximation is exact for

each integral signal group diagram that satisfies the constraints of the rounding MILP problem (5.10). Recall that the average (weighted) delay equals:

$$D = \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q = \sum_{i \in \mathcal{S}} d_i,$$

where $d_i := \sum_{q \in \mathcal{Q}_i} w_q d_q$ is the contribution of signal group i to the average delay. For ease of notation we define $R_{i,k} := \chi^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k)$, which is the k th red time of signal group $i \in \mathcal{S}$. Moreover, we define $\tilde{R}_{i,k} := \tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k)$, which is the k th red time for the signal group diagram that we try to round. We approximate each delay term d_i , $i \in \mathcal{S}$ by a piecewise linear approximation.

Consider a signal group $i \in \mathcal{S}$. For this signal group we split the delay term d_i in $K_i + 1$ different terms $d_i = d_{i,1}^{\text{det}} + \dots + d_{i,\bar{K}_i}^{\text{det}} + d_i^{\text{stoch}}$, see (4.9). The deterministic term $d_{i,k}^{\text{det}}$, $k = 1, \dots, \bar{K}_i$ depends only on $R_{i,k}$. Recall that the value of $R_{i,k}$ is restricted to an interval with a length of at most one, i.e.,

$$\lfloor \tilde{R}_{i,k} \rfloor \leq R_{i,k} \leq \lceil \tilde{R}_{i,k} \rceil.$$

Therefore, we can replace this deterministic term $d_{i,k}^{\text{det}}$ by a linear approximation $f_{i,k}^{\text{det}}$ that is exact for the integral red time $\lfloor \tilde{R}_{i,k} \rfloor$ and for the integral red time $\lceil \tilde{R}_{i,k} \rceil$:

$$f_{i,k}^{\text{det}} := d_{i,k}^{\text{det}}(\lfloor \tilde{R}_{i,k} \rfloor) + (d_{i,k}^{\text{det}}(\lceil \tilde{R}_{i,k} \rceil) - d_{i,k}^{\text{det}}(\lfloor \tilde{R}_{i,k} \rfloor))(R_{i,k} - \lfloor \tilde{R}_{i,k} \rfloor).$$

The term d_i^{stoch} depends on the total red time R_i of signal group i :

$$R_i = \sum_{k=1}^{K_i} R_{i,k} = \sum_{k=2}^{K_i} \left(\lfloor \tilde{R}_{i,k} \rfloor + \mathbf{u}_{i,k} \right) - \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1).$$

Note that R_i only depends on the binary-valued design variables $\mathbf{u}_{i,k}$, $k = 2, \dots, K_i$ and $\tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1)$ and that the total red time R_i is restricted to the following interval:

$$\sum_{k=1}^{K_i} \lfloor \tilde{R}_{i,k} \rfloor \leq R_i \leq \lceil \sum_{k=1}^{K_i} \tilde{R}_{i,k} \rceil.$$

The lower bound follows directly from the fact that we round each red time up or down. The upper bound follows from the lower bound on the total green time (5.10e) combined with the fact that the green and red intervals of signal group $i \in \mathcal{S}$ together comprise one period. We approximate the stochastic term d_i^{stoch} by a piecewise linear function, see also Section 3.3.4. The break points of this piecewise linear function correspond to the integral total red times R_i that satisfy the above inequality. To model this piecewise linear function, we use a real-valued design variable $\mathbf{d}_i^{\text{stoch}}$; its value approximates d_i^{stoch} .

The formulated objective function is linear in the red times $R_{i,k} := \chi^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k)$, $i \in \mathcal{S}$, $k \in \mathcal{K}$ and the real-valued design variables $\mathbf{d}_i^{\text{stoch}}$. Furthermore, note that when fixing the binary-valued design variables, the objective function can be formulated as a linear function of only the variables $R_{i,1} := \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1)$, $i \in \mathcal{S}$. To this end, note that

each term d_i then only depends on $\tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \in [[\tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1)], \lceil \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \rceil]$. We can obtain the linear function by replacing the delay term d_i by f_i :

$$f_i := d_i(\lfloor \tilde{R}_{i,1} \rfloor) + (d_i(\lceil \tilde{R}_{i,1} \rceil) - d_i(\lfloor \tilde{R}_{i,1} \rfloor))(R_{i,k} - \lfloor \tilde{R}_{i,1} \rfloor).$$

Consider the minimization of the delay that road users experience at the intersection. During the first step of the two-step approach we then include the constraints (4.5). These constraints force each queue to be emptied during each effective green time for the deterministic queueing process associated with the deterministic delay term. We have included these constraints during the first step for the following two reasons. First, because we can then write the deterministic term d_i^{det} as the following sum $d_{i,1}^{\text{det}} + \dots + d_{i,\bar{K}_i}^{\text{det}}$, where each of these terms depends on only one red time. Second, the deterministic delay term d_i^{det} and also the delay $d_i := d_i^{\text{det}} + d_i^{\text{stoch}}$ are not convex when we do not force each queue to be emptied during every effective green time, see Appendix D.2. In this section we do not force each queue to be emptied during an effective green time. However, we still use the approximation for the delay that is based on this assumption. We do this for the following reasons. First, when we include the constraints that force each queue to be emptied during every effective green time, the proposed rounding MILP problem possibly does not return an integral signal group diagram. Second, for the signal group diagram that we try to round it holds that each queue is emptied during every effective green time. We only round each green time and each red time up or down and, therefore, whenever a queue is not emptied during an effective green time, it is probably close to being emptied. As a consequence, we expect the assumption that each queue is empty at the end of an effective green interval (for the deterministic and fluid-like queueing process associated with the deterministic delay term) to have a small effect on the delay. To obtain the correct value for the deterministic delay term (and also the correct value for the delay itself) we again compute the deterministic delay term for the signal group diagram returned by the rounding MILP problem; in this computation we do account for the queue (possibly) not being empty at the end of an effective green interval.

5.7.2 The rounding MILP problem returns an integral signal group diagram

In the following lemma we state which solutions to the rounding MILP problem (5.10) correspond to an integral signal group diagram.

Lemma 5.6. *Assume that the yellow times Y_i , $i \in \mathcal{S}$ are integral. Consider a signal group diagram with an integral period duration $T > 0$. A solution $(\chi^{\text{ind}}, \mathbf{u})$ to its rounding MILP problem (5.10) corresponds to an integral signal group diagram if and only if χ^{ind} is integral.*

Proof. Let K_i be the number of realizations of signal group $i \in \mathcal{S}$ for the signal group diagram that we try to round. Assuming that the yellow times Y_i , $i \in \mathcal{S}$ are integral, the

signal group diagram is integral if and only if all switches to a green indication and all switches to a red indication are scheduled at an integral second. This implies that the signal group diagram is integral if and only if the times $t^{\text{ind}}(\varepsilon)$, $\varepsilon \in \mathcal{E}$ are integral, where

$$\mathcal{E} = \{\textcircled{i}_k \mid i \in \mathcal{S}, k = 1, \dots, K_i\} \cup \{\textcircled{\textcolor{red}{i}}_k \mid i \in \mathcal{S}, k = 1, \dots, K_i\}.$$

Thus, what remains is to prove that the time $t^{\text{ind}}(\varepsilon)$ is integral for each event $\varepsilon \in \mathcal{E}$ if and only if χ^{ind} is integral.

The integrality of the times $t^{\text{ind}}(\varepsilon)$, $\varepsilon \in \mathcal{E}$ follows from the iteration scheme that can be used to obtain these switching times, see also Section 3.2.1. For each of the connected components of the constraint graph G , this iteration scheme sets the value of $t^{\text{ind}}(\varepsilon)$ to zero for one of the events in this connected component. Subsequently, this iteration scheme obtains all values for $t^{\text{ind}}(\varepsilon)$, $\varepsilon \in \mathcal{E}$ by using the following relation:

$$t^{\text{ind}}(\varepsilon_2) = t^{\text{ind}}(\varepsilon_1) + \chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) - z(\varepsilon_1, \varepsilon_2)T, \quad (5.11)$$

where $z(\varepsilon_1, \varepsilon_2)$ is some integer. Assume that the constraint graph G contains some arc $(\varepsilon_1, \varepsilon_2)$ for which $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)$ is not integral. From (5.11) it then follows that either $t^{\text{ind}}(\varepsilon_1)$ is not integral or $t^{\text{ind}}(\varepsilon_2)$ is not integral.

Now assume that χ^{ind} is integral. We prove that $t^{\text{ind}}(\varepsilon)$ is then integral for all events $\varepsilon \in \mathcal{E}$. We do so by a proof of induction. Without loss of generality assume that the constraint graph G consists of only one connected component. Let ε_0 be the event for which we have set its time to zero, i.e., $t^{\text{ind}}(\varepsilon_0) := 0$. Define \mathcal{E}_k , $k \geq 0$ as the set of vertices of the constraint graph G whose distance from vertex ε_0 (in number of arcs) equals k , e.g., $\mathcal{E}_0 = \{\varepsilon_0\}$ and $\mathcal{E}_1 := \{\varepsilon \mid (\varepsilon, \varepsilon_0) \in A\} \cup \{\varepsilon \mid (\varepsilon_0, \varepsilon) \in A\}$. We use the following induction hypothesis: the event times $t^{\text{ind}}(\varepsilon)$ are integral for all the vertices $\varepsilon \in \mathcal{E}_0 \cup \dots \cup \mathcal{E}_{k-1}$. Note that this induction hypothesis is trivial for $k = 1$. Using this induction hypothesis we prove that the event times $t^{\text{ind}}(\varepsilon)$ are integral for all vertices $\varepsilon \in \mathcal{E}_0 \cup \dots \cup \mathcal{E}_k$, which implies the lemma. Consider some event $\varepsilon_k \in \mathcal{E}_k$. By definition, the vertex ε_k is connected to some event $\varepsilon_{k-1} \in \mathcal{E}_{k-1}$. By our induction hypothesis it holds that $t^{\text{ind}}(\varepsilon_{k-1})$ is integral. Therefore, from (5.11) it follows that $t^{\text{ind}}(\varepsilon_k)$ is also integral. This proves the induction step and concludes this proof. \square

Consider a signal group diagram returned by the first step of the two-step approach. Let $P(\mathbf{u})$ be the polyhedron spanned by the constraints of its rounding MILP problem (5.10) when the binary-valued design variables are set to the values in \mathbf{u} . In the following theorem we state that the polyhedron $P(\mathbf{u})$ is integral. Assuming that the yellow times Y_i , $i \in \mathcal{S}$ are integral, it follows from Lemma 5.6 that each vertex of the polyhedron $P(\mathbf{u})$ corresponds to an integral signal group diagram.

Theorem 5.7. *Consider a signal group diagram with an integral period duration. Define the polyhedron $P(\mathbf{u})$ to be the polyhedron that is spanned by the constraints of its rounding MILP problem (5.10) when the binary-valued design variables are fixed to the values in \mathbf{u} . The polyhedron $P(\mathbf{u})$ is integral for each binary vector \mathbf{u} .*

Proof. For ease of notation we define $\chi := \chi^{\text{ind}}$. The polyhedron P can be written in the following form:

$$P(\mathbf{u}) := \{\chi \mid M_C \chi = b_C, \underline{b}_P \leq M_P \chi \leq \bar{b}_P, \underline{\chi} \leq \chi \leq \bar{\chi}\}, \quad (5.12)$$

where M_C is a cycle-arc incidence matrix associated with some integral cycle basis of the constraint graph G , M_P is a path-arc incidence matrix of the constraint graph G , and the vectors b_C , \underline{b}_P , \bar{b}_P , $\underline{\chi}$, and $\bar{\chi}$ are all integral. For more details we refer to Appendix E.2. In the following lemma we relate the polyhedron $P := P(\mathbf{u})$ to a (simpler) polyhedron P' and prove that P has integral values if and only if P' has integral values.

Lemma 5.8. *Consider a polyhedron P of the form:*

$$P = \{\chi \mid M_C \chi = b_C, \underline{b}_P \leq M_P \chi \leq \bar{b}_P, \underline{\chi} \leq \chi \leq \bar{\chi}\},$$

where M_C is a cycle-arc incidence matrix associated with an integral cycle basis \mathcal{B} of a graph G , M_P is some path-arc incidence matrix associated with paths $\mathcal{P}_1, \dots, \mathcal{P}_n$ in the same graph G , and where b_C , \underline{b}_P , \bar{b}_P , $\underline{\chi}$, and $\bar{\chi}$ are vectors. Define the polyhedron P' :

$$P' = \{\chi' \mid M'_C \chi' = b'_C, \underline{\chi}' \leq \chi' \leq \bar{\chi}'\},$$

where

$$M'_C := \begin{bmatrix} M_C & \mathbf{0} \\ M_P & -I \end{bmatrix}, \quad b'_C := \begin{bmatrix} b_C \\ \mathbf{0} \end{bmatrix}, \quad \underline{\chi}' := \begin{bmatrix} \underline{\chi} \\ \underline{b}_P \end{bmatrix}, \quad \bar{\chi}' := \begin{bmatrix} \bar{\chi} \\ \bar{b}_P \end{bmatrix}.$$

Here we define $\mathbf{0}$ to be a matrix of zeros. The size of $\mathbf{0}$ can be abstracted from the context. The polyhedron P has integral vertices if and only if the polyhedron P' has integral vertices. Furthermore, the matrix M'_C is a cycle-arc incidence matrix associated with some graph G' .

Proof. We introduce an additional variable s_i for each path \mathcal{P}_i , $i = 1, \dots, n$. Using the equality $s = M_P \chi$, we can replace the inequalities $\underline{b}_P \leq M_P \chi \leq \bar{b}_P$ by the inequalities $\underline{b}_P \leq s \leq \bar{b}_P$. Therefore, we can rewrite polyhedron P to:

$$P = \{\chi \mid \exists s \in \mathbb{R}^n : \begin{bmatrix} M_C & \mathbf{0} \\ M_P & -I \end{bmatrix} \begin{bmatrix} \chi \\ s \end{bmatrix} = \begin{bmatrix} b_C \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \underline{\chi} \\ \underline{b}_P \end{bmatrix} \leq \begin{bmatrix} \chi \\ s \end{bmatrix} \leq \begin{bmatrix} \bar{\chi} \\ \bar{b}_P \end{bmatrix}\},$$

where I is an identity matrix, which we use to write the equality $s = M_P \chi$ as $M_P \chi - I s = \mathbf{0}$. Using the definitions of M'_C , b'_C , $\underline{\chi}'$ and $\bar{\chi}'$ we find:

$$P = \{\chi \mid \exists s \in \mathbb{R}^n : M'_C \begin{bmatrix} \chi \\ s \end{bmatrix} = b'_C, \underline{\chi}' \leq \begin{bmatrix} \chi \\ s \end{bmatrix} \leq \bar{\chi}'\},$$

which is very similar to the definition of P' :

$$P' = \left\{ \begin{bmatrix} \chi \\ s \end{bmatrix} \mid M'_C \begin{bmatrix} \chi \\ s \end{bmatrix} = b'_C, \underline{\chi}' \leq \begin{bmatrix} \chi \\ s \end{bmatrix} \leq \bar{\chi}' \right\}.$$

Define $\theta(\chi)$ as follows:

$$\theta(\chi) := \begin{bmatrix} \chi \\ M_P \chi \end{bmatrix}.$$

From the equality $s = M_P \chi$ it follows that $P' := \{\theta(\chi) \mid \chi \in P\}$. It holds that $\chi \in P$ is a vertex of the polyhedron P if and only if $\theta(\chi)$ is a vertex of the polyhedron P' . This follows from the observation that the vector $\theta(\chi)$ can be written as a convex combination $\theta(\chi) = \alpha\theta(\chi_1) + (1 - \alpha)\theta(\chi_2)$ of two other vectors $\theta(\chi_1) \in P'$ and $\theta(\chi_2) \in P'$ if and only if the vector $\chi \in P$ can be written as the convex combination $\chi = \alpha\chi_1 + (1 - \alpha)\chi_2$ of the two other vectors $\chi_1 \in P$ and $\chi_2 \in P$. Note that, in addition, $\theta(\chi)$ is integral if χ is integral, which implies that the vertices of P are integral if and only if the vertices of P' are integral.

What remains is to prove that M'_C is a cycle-arc incidence matrix associated with an integral cycle basis \mathcal{B}' of some graph G' . For each of the auxiliary variables s_i we add an arc a_i to graph $G = (V, A)$ resulting in graph $G' = (V', A')$; arc a_i starts at the vertex at which path \mathcal{P}_i starts and arc a_i ends at the vertex at which path \mathcal{P}_i ends. Let \mathcal{C}_i , $i = 1, \dots, n$ denote the cycle consisting of path \mathcal{P}_i and arc a_i (in the backward direction). We can prove that the set of cycles $\mathcal{B}' := \mathcal{B} \cup \{\mathcal{C}_1 \cup \dots, \mathcal{C}_n\}$ is an integral cycle basis of the graph G' ; note that its cycle-arc incidence matrix is M'_C . First, we prove that the set \mathcal{B}' consists of the correct number of cycles. Let d (d') be the number of cycles in a cycle basis of the graph G (G'). The additional arc a_i , $i = 1, \dots, n$ connects two vertices that were already connected, e.g., via path \mathcal{P}_i . As a result, graph G and graph G' have the same number of connected components, i.e., $\nu(G) = \nu(G')$. This implies $d' := |A'| - |V'| + \nu(G') = |A| + n - |V| + \nu(G) = d + n$. Therefore, \mathcal{B}' consists of the correct number of cycles. Let X be a non-singular $d \times d$ submatrix of cycle-arc incidence matrix M_C . By numbering the arcs of the graph G' appropriately we find that the cycle-arc incidence matrix M'_C has a $d' \times d'$ submatrix of the form:

$$X' := \begin{bmatrix} X & \mathbf{0} \\ D & -I \end{bmatrix},$$

where $\mathbf{0}$ is the zero-matrix of size $d \times n$ and I is the identity matrix of size $n \times n$; the last n rows of X' are indexed by the cycles $\mathcal{C}_1, \dots, \mathcal{C}_n$ and the last n columns of X' are indexed by the arcs a_1, \dots, a_n . Hence, $\det(X') = \det(X) \det(-I) = -\det(X)$ and from Theorem 5.2 it follows that \mathcal{B}' is an integral cycle basis. This concludes this proof. \square

Thus, the polyhedron P has integral vertices if and only if the polyhedron $P' = \{\chi' \mid M'_C \chi' = b'_C, \underline{\chi}' \leq \chi' \leq \bar{\chi}'\}$, which is defined in Lemma 5.8, has integral vertices. Note that the vectors b'_C , $\underline{\chi}'$, and $\bar{\chi}'$ are integral and the matrix M'_C is a cycle-arc incidence vector associated with some graph G' . Therefore, the theorem follows from the following lemma, which states that the polyhedron P' has integral vertices.

Lemma 5.9. *Consider a polyhedron P' of the form:*

$$P' = \{\mathbf{x}' \mid M'_C \mathbf{x}' = b'_C, \underline{\mathbf{x}}' \leq \mathbf{x}' \leq \bar{\mathbf{x}}'\}, \quad (5.13)$$

where M'_C is a cycle-arc incidence matrix associated with an integral cycle basis of some graph G' , and the vectors b'_C , $\underline{\mathbf{x}}'$ and $\bar{\mathbf{x}}'$ are integral. The polyhedron P' has integral vertices.

Proof. Recall that a totally unimodular cycle basis $\mathcal{B}'' = \{\mathcal{C}''_1, \dots, \mathcal{C}''_d\}$ exists for each graph. Let M''_C be the cycle-arc incidence matrix associated with such a totally unimodular cycle basis of the graph G' . Furthermore, let $\mathcal{B}' = \{\mathcal{C}'_1, \dots, \mathcal{C}'_d\}$ be the integral cycle basis associated with cycle-arc incidence matrix M'_C . Since cycle basis \mathcal{B}' is integral, for each cycle \mathcal{C}''_i , $i = 1, \dots, d$ we can find integral values $\alpha_{i,j}$, $j = 1, \dots, d$ such that:

$$C''_i = \sum_{j=1}^d \alpha_{i,j} C'_j.$$

Therefore, it holds that:

$$M''_C = U M'_C, \quad \text{where,} \quad U = [\alpha_{i,j}]_{i,j=1,\dots,d} \in \mathbb{Z}^{d \times d}.$$

Hence, we can rewrite polyhedron P' to $P' := \{\mathbf{x}' \mid U M'_C \mathbf{x}' = U b'_C, \underline{\mathbf{x}}' \leq \mathbf{x}' \leq \bar{\mathbf{x}}'\} = \{\mathbf{x}' \mid M''_C \mathbf{x}' \leq b''_C, \underline{\mathbf{x}}' \leq \mathbf{x}' \leq \bar{\mathbf{x}}'\}$, where b''_C is an integral vector and M''_C is TU. From Theorem 5.3 it follows that the polyhedron P' has integral vertices, which concludes this lemma. \square

We have proved that the polyhedron $P(\mathbf{u})$ is integral for each binary vector \mathbf{u} , which concludes the proof of this theorem. \square

Consider a signal group diagram returned by step one of the two-step approach. Theorem 5.7 proves that its polyhedron $P(\mathbf{u})$ is integral. Assuming that the yellow times Y_i , $i \in \mathcal{S}$ are integral, it follows from Lemma 5.6 that each vertex of the polyhedron $P(\mathbf{u})$ corresponds to an integral signal group diagram.

Assume that the rounding MILP problem (5.10) has a linear objective function (and not a piecewise linear one). Let \mathbf{u}^* be the vector \mathbf{u} for the solution that is returned by this rounding MILP problem (5.10). An optimal solution of the MILP problem must then be positioned at one of the vertices of $P(\mathbf{u}^*)$. Since $P(\mathbf{u}^*)$ is an integral polyhedron, this implies that the rounding MILP problem has an optimal solution that is integral (this integral solution corresponds to an integral signal group diagram). We can obtain this integral solution by fixing the binary-valued design variables to the values in \mathbf{u}^* and subsequently solving the linear programming problem with a simplex algorithm.

Now consider the minimization of the delay that road users experience. This objective function can be formulated as done in Section 5.7.1. Recall that this objective function is linear in the variables \mathbf{x}^{ind} when we fix the binary-valued design variables. As a consequence, for this objective function we can obtain an integral signal group diagram as

follows. First we solve the rounding MILP problem (5.10). Let \mathbf{u}^* be the vector \mathbf{u} of the corresponding solution. We can then set the binary-valued design variables of the rounding MILP problem (5.10) to the values in \mathbf{u}^* and reformulate the objective function (minimization of the average delay that road users experience) as a linear objective function. Since $P(\mathbf{u}^*)$ is an integral polyhedron, solving the linear programming problem with a simplex algorithm results in an integral signal group diagram.

5.7.3 Feasibility of the rounding MILP problem

In this section we consider the feasibility of the rounding MILP problem. The following lemma proves that this rounding MILP problem is feasible when at most one signal group $i \in \mathcal{S}$ has multiple realizations for the signal group diagram that we attempt to round.

Lemma 5.10. *Consider a signal group diagram with an integral period duration satisfying Prop(1). When at most one signal group has multiple realizations for this signal group diagram, its rounding MILP problem (5.10) is feasible.*

Proof. Let $\tilde{\chi}^{\text{ind}}(\varepsilon_1, \varepsilon_2), (\varepsilon_1, \varepsilon_2) \in A$ be the value for $\chi^{\text{ind}}(\varepsilon_1, \varepsilon_2)$ associated with the signal group diagram for which we have formulated the rounding MILP problem (5.10). Let $\tilde{\chi}^{\text{ind}}$ be a vector of these values $\tilde{\chi}^{\text{ind}}(\varepsilon_1, \varepsilon_2), (\varepsilon_1, \varepsilon_2) \in A$. First, consider the case that each signal group has a single realization. The rounding MILP problem then has no binary-valued design variables. As a consequence, the signal group diagram that we attempt to round satisfies all constraints of this MILP problem. This implies that the rounding MILP problem is then feasible.

Consider the case that one signal group has multiple realizations. Without loss of generality assume that signal group 1 has multiple realizations. For each of the red times $\tilde{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k), k = 2, \dots, K_1$ we have a binary-valued design variable $\mathbf{u}_{1,k}$. We force such a red time to be rounded up or rounded down with constraints (5.10c). Suppose that for these red times of signal group 1 we would replace the constraint (5.10c) by the constraints (5.10b). We then do not force such a red time $\chi^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k)$ to be rounded up or rounded down, but instead we allow the red time $\chi^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k)$ to be any real value between the floored and the ceiled value of $\tilde{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k)$. Furthermore, suppose that we omit the lower bound (5.10e) on the total green time of signal group 1. The binary-valued design variables are then used by none of the constraints anymore and, therefore, we only have the real-valued design variables χ^{ind} . Let the polyhedron P be the polyhedron that is spanned by the constraints of the resulting linear programming problem. We prove that this polyhedron has an integral vertex for which the lower bound (5.10e) on the total green time of signal group 1 is satisfied. For this integral vertex each red time is either rounded up or rounded down and, therefore, we can find the appropriate value for each of the binary-valued design variables $\mathbf{u}_{1,k}$. This implies that this integral vertex satisfies all the constraints of the rounding MILP problem (5.10). Hence, this rounding MILP problem is feasible.

First we prove that each vertex of P is integral. The proof is the same as that of Theorem 5.7 because we can write the polyhedron P as follows:

$$P := \{\mathbf{x} \mid M_C \mathbf{x} = b_C, \underline{b}_P \leq M_P \mathbf{x} \leq \bar{b}_P, \underline{\chi} \leq \mathbf{x} \leq \bar{\chi}\}, \quad (5.14)$$

where M_C is a cycle-arc incidence matrix associated with some integral cycle basis of the constraint graph G , M_P is a path-arc incidence matrix of the constraint graph G , and the vectors b_C , \underline{b}_P , \bar{b}_P , $\underline{\chi}$, and $\bar{\chi}$ are all integral. For more details we refer to Appendix E.2, where the only difference is that for some realizations $k \in \mathcal{S}$ of signal group 1 we now have the following bounds:

$$\begin{aligned} \underline{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) &:= \lfloor \tilde{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) \rfloor, \\ \bar{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) &:= \lceil \tilde{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) \rceil, \end{aligned}$$

instead of the bounds:

$$\begin{aligned} \underline{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) &:= \lfloor \tilde{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) \rfloor + \mathbf{u}_{i,k}, \\ \bar{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) &:= \lceil \tilde{\chi}^{\text{ind}}(\textcircled{1}_{k-1}, \textcircled{1}_k) \rceil + \mathbf{u}_{i,k}. \end{aligned}$$

It remains to prove that lower bound (5.10e) is satisfied by at least one of these vertices. Note that the vector $\tilde{\chi}^{\text{ind}}$ is included in P and satisfies the lower bound (5.10e). As this vector is included in the polyhedron P , it can be written as a convex combination of the vertices of P . This implies that P has at least one vertex for which the lower bound (5.10e) is satisfied. This concludes this proof. \square

Remark 5.6. *The above lemma states that the rounding MILP problem is feasible when at most one signal group has multiple realizations for the signal group diagram that we attempt to round. This rounding MILP problem, however, might be infeasible if more than one signal group receives multiple realizations. In Appendix E.3 we consider an example for which the rounding MILP problem is infeasible. In that appendix we consider a signal group diagram satisfying Prop(1) for which four signal groups receive multiple realizations. The rounding MILP problem of this signal group diagram is infeasible.*

5.7.4 Optimality of the two-step approach

In this section we give some proofs on the optimality of the two-step approach. The following lemma concerns the minimization of the period duration.

Lemma 5.11. *Consider the problem of minimizing the period duration. Let at most one signal group have multiple realizations for the signal group diagram returned by step one of the two-step approach. The two-step approach then returns the optimal integral signal group diagram.*

Proof. Let T be the period duration of the signal group diagram returned by step one. During step one of the two-step approach, we find the optimal signal group diagram amongst the signal group diagrams that satisfy Prop(1). Recall that an integral signal group diagrams also satisfies Prop(1). Therefore, the period duration T is a lower bound on the period duration of any feasible integral signal group diagram. From Lemma 5.10 it follows that the rounding MILP problem is feasible and, hence, we can find an integral signal group diagram with a period duration of T seconds, which proves that this signal group diagram is optimal. \square

The following lemma concerns the maximization of the capacity of the intersection.

Lemma 5.12. *Consider the problem of maximizing the capacity of the intersection. Let at most one signal group have multiple realizations for the signal group diagram returned by step one of the two-step approach. The two-step approach then returns the optimal (up to some arbitrary precision) integral signal group diagram.*

Proof. Let β be the maximum growth factor returned by step one. This growth factor is the largest growth factor (up to some arbitrary precision) for which a signal group diagram exists that satisfies Prop(β). Recall that an integral signal group diagram that can handle a growth factor β also satisfies Prop(β). Therefore, the growth factor β is an upper bound (up to some arbitrary precision) on the largest growth factor that is sustainable by any feasible integral signal group diagram. From Lemma 5.10 it follows that the rounding MILP problem is feasible and, hence, we can find an integral signal group diagram that can handle a growth factor β , which proves this lemma. \square

The following lemma concerns the minimization of the average (weighed) delay that road users experience at the intersection.

Lemma 5.13. *Consider the problem of minimizing the average (weighted) delay that road users experience. Let each signal group have a single realization for the signal group diagram returned by step one of the two-step approach. The two-step approach then returns the optimal integral signal group diagram.*

Proof. During step one of the two-step approach, we find the optimal signal group diagram that satisfies Prop(1) (and (4.5)). Let J be the objective value (delay) of the signal group diagram returned by step one. Recall that the objective function is the piecewise linear approximation D^{pw1} . This piecewise linear approximation is exact for each integral signal group diagram, i.e., we have $D^{\text{pw1}} = D$ for each integral signal group diagram schedule. Moreover, recall that each integral signal group diagram also satisfies Prop(1). As a consequence, J is a lower bound on the objective value of any integral signal group diagram that satisfies (4.5). We prove that the rounding MILP problem finds an integral signal group diagram for which the objective value is J seconds and the inequalities (4.5) are satisfied; this would prove the lemma. Each signal group has a single realization for the signal group diagram returned by step one. As a result, its rounding MILP problem (5.10)

has only real-valued design variables; it is then not a mixed-integer programming problem, but a linear programming problem. The objective function (minimization of the delay) of this linear programming problem can be written as a linear function of these real-valued design variables, see Section 5.7.1. The signal group diagram returned by step one, satisfies all constraints of this linear programming problem. As a consequence, solving this linear programming with a simplex algorithm (the second step of the two-step approach) results in an integral signal group diagram with an objective value of J seconds. Moreover, the resulting signal group diagram satisfies the inequalities (4.5); these inequalities are automatically satisfied by each integral signal group diagram for which each signal group has only one realization. This proves the lemma. \square

5.8 Numerical results

In this section we perform a numerical study to test the two-step approach introduced in this chapter.

This section is split into three different parts. In Section 5.8.1 we consider the minimization of the period duration, in Section 5.8.2 we consider the maximization of the capacity of the intersection, and in Section 5.8.3 we consider the minimization of the average (weighted) delay that road users experience. We test these three objective functions on the thirteen real-life intersections from (Fleuren and Lefebvre, 2016a), which are grouped by their sizes into three groups: small (S), medium (M) and large (L). For all test cases we fix the minimum number of realizations to one for each signal group $i \in \mathcal{S}$. We do however vary the maximum number of realizations. We consider three variants: we allow the zero, two or four most heavily loaded signal groups to have an additional realization. This gives a total of $13 \times 3 = 39$ test cases except when minimizing the period duration; when minimizing the period duration, we also consider six different scalings of the arrival rates λ_q : 1.00, 1.05, 1.10, 1.15, 1.20 and 1.25. Therefore, for this objective function we have $13 \times 3 \times 6 = 234$ test cases. We have solved each test case once on a computer with specifications: Intel i5-4300U CPU @1.90GHZ with 16.0GB of RAM.

5.8.1 Minimizing the period duration

In this section we perform a numerical study to test the two-step approach for the minimization of the period duration. For all the test cases minimizing the period duration we were able to find the optimal integral signal group diagram. In Table 5.1 we give the objective values for each of these test cases. In Table 5.2 we compare these values with the minimum period duration that result from the optimization over all signal group diagrams (also the ones that are not integral). Finally, in Table 5.3 we give the (geometric) average computation times. In this table we also give the (arithmetic) average number of (MI)LP problems that were solved to obtain the integral signal group diagrams; if the

signal group diagram returned by step one is not integral, the number of MILP problems also includes the rounding MILP problem.

5.8.2 Maximizing the capacity

In this section we consider the two-step approach that maximizes the capacity of the intersection. To obtain the signal group diagram in step one of the two-step approach, we use a bisection accuracy of $\delta := 10^{-5}$. For all the test cases we were able to find the optimal integral signal group diagram. In Table 5.4 we give the objective values of the $13 \times 3 = 39$ test cases. Furthermore, in this table we compare these objective values with the objective values that result from the optimization over all signal group diagrams (also the ones that are not integral). We can see that the maximum growth factor decreases slightly (on average only 0.56 percent) when we restrict ourselves to integral signal group diagrams (instead of all signal group diagrams). In Table 5.5 we give the computation times, the number of MILP problems that were solved, and the number of linear programming problems that were solved for these test cases.

Scaling 'Ad. real.'	1.00			1.05			1.10			1.15			1.20			1.25		
	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4
S1	36	36	36	40	40	40	44	44	44	49	49	49	56	56	56	65	65	65
S2	30	30	30	30	30	30	30	30	30	30	30	30	33	33	33	37	37	37
S3	41	41	41	44	44	44	44	44	44	48	48	48	51	51	51	55	55	55
S4	66	66	66	70	70	70	74	74	74	79	79	79	84	84	84	89	89	85
S5	65	65	65	70	70	70	73	73	73	80	80	80	87	87	87	96	96	96
M1	45	45	45	47	47	47	49	49	49	51	51	51	60	60	60	71	71	71
M2	47	47	47	49	49	49	54	54	54	58	58	58	64	64	64	73	73	73
M3	143	143	143	170	170	170	208	208	208	∞	∞	∞	∞	∞	∞	∞	∞	∞
M4	82	82	82	88	88	88	95	95	95	102	102	102	111	111	111	124	124	124
L1	77	77	77	79	79	79	83	83	83	94	94	94	135	135	135	∞	∞	∞
L2	85	85	85	93	93	93	100	100	100	107	107	107	126	126	126	∞	∞	∞
L3	78	78	78	82	82	82	86	86	86	95	95	95	108	108	108	∞	∞	∞
L4	72	72	72	75	75	75	84	84	84	92	92	92	103	103	103	117	117	117

Table 5.1: The objective values for each of the $13 \times 6 \times 3$ test cases in the numerical study that minimizes the period duration. The first column indicates which intersection is considered, the first row indicates which scaling of the arrival rates is considered, and the second row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations. This table contains infinite values; these infinite values indicate that no feasible integral signal group diagram exists.

Scaling		1.00			1.05			1.10			1.15			1.20			1.25		
Ad.	real.	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4	0	2	4
S1		1.00	1.00	1.00	1.02	1.02	1.02	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.02	1.02	1.02
S2		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.07	1.07	1.07	1.08	1.08	1.08
S3		1.05	1.05	1.05	1.08	1.08	1.08	1.02	1.02	1.02	1.05	1.05	1.05	1.05	1.05	1.05	1.06	1.06	1.06
S4		1.00	1.00	1.00	1.01	1.01	1.01	1.01	1.01	1.01	1.03	1.03	1.03	1.03	1.03	1.01	1.03	1.03	1.02
S5		1.01	1.01	1.01	1.04	1.04	1.04	1.02	1.02	1.02	1.06	1.06	1.06	1.05	1.05	1.05	1.04	1.04	1.04
M1		1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.02	1.02	1.02	1.07	1.07	1.07	1.05	1.05	1.05
M2		1.03	1.03	1.03	1.02	1.02	1.02	1.03	1.03	1.03	1.01	1.01	1.01	1.01	1.01	1.01	1.04	1.04	1.04
M3		1.01	1.01	1.01	1.02	1.02	1.02	1.01	1.01	1.01	-	-	-	-	-	-	-	-	-
M4		1.02	1.02	1.02	1.03	1.03	1.03	1.03	1.03	1.03	1.02	1.02	1.02	1.01	1.01	1.01	1.03	1.03	1.03
L1		1.03	1.03	1.03	1.01	1.01	1.01	1.01	1.01	1.01	1.09	1.09	1.09	1.11	1.11	1.11	-	-	-
L2		1.02	1.02	1.02	1.04	1.04	1.04	1.03	1.03	1.03	1.01	1.01	1.01	1.09	1.09	1.09	-	-	-
L3		1.05	1.05	1.05	1.02	1.02	1.02	1.02	1.02	1.02	1.03	1.03	1.03	1.04	1.04	1.04	∞	∞	∞
L4		1.01	1.01	1.01	1.00	1.00	1.00	1.04	1.04	1.04	1.03	1.03	1.03	1.04	1.04	1.04	1.04	1.04	1.04

Table 5.2: Ratio of the objective values that result from the optimization of integral signal group diagrams (Table 5.1) to the objective values that result from the optimization of signal group diagrams that are not necessarily integral (Table 4.3). In this Table we have shown the results for each of the $13 \times 6 \times 3$ test cases in the numerical study that minimizes the period duration. The first column indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations. This table contains infinite values; these infinite values indicate that no integral signal group diagram exists, however a signal group diagram exists that is not integral. A '-' indicates that no signal group diagram exists at all (so also no signal group diagram exists that is not integral).

Ad. real.		0			2			4		
		Time	#MILP	#LP	Time	#MILP	#LP	Time	#MILP	#LP
CPLEX	Small	0.16	1.73	1.23	0.20	1.70	1.30	0.22	1.70	1.20
	Medium	0.25	2.14	1.57	0.31	1.90	1.57	0.52	1.86	1.57
	Large	0.49	2.81	2.33	1.00	2.76	2.38	3.04	2.62	2.43
GUROBI	Small	0.01	2.03	1.30	0.02	1.80	1.27	0.06	1.70	1.20
	Medium	0.04	2.05	1.62	0.21	2.00	1.67	1.44	2.00	1.52
	Large	0.49	2.67	2.48	7.50	2.67	2.43	117.44	2.57	2.38
SCIP	Small	0.04	1.73	1.27	0.08	1.70	1.27	0.16	1.70	1.20
	Medium	0.16	1.86	1.57	0.51	1.86	1.57	1.98	1.90	1.52
	Large	1.84	2.62	2.38	9.29	2.62	2.33	38.56	2.62	2.38
Column average		-	2.18	1.75	-	2.11	1.75	-	2.07	1.71

Table 5.3: The (geometric) average computation times, the (arithmetic) average number of MILP problems that were solved, and the (arithmetic) average number of LP problems that were solved for the $13 \times 6 \times 3$ test cases in the numerical study that minimize the period duration. We also give the (arithmetic) column average for the number of MILP problems and the number of LP problems that were solved. We distinguish between three types of intersections (small, medium and large), and three types of solvers (CPLEX 12.6.1.0, GUROBI 6.0.5. and SCIP 3.2.0). Furthermore, we vary the number of signal groups that is allowed to have an additional realization, see the first row of this table. If this number of signal groups equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Ad. real.	Objective			Ratio		
	0	2	4	0	2	4
S1	1.39	1.39	1.39	1.000	1.000	1.000
S2	1.55	1.55	1.55	0.994	0.994	0.994
S3	1.64	1.64	1.64	0.992	0.992	0.992
S4	1.47	1.47	1.49	0.992	0.992	0.995
S5	1.34	1.34	1.34	0.992	0.992	0.992
M1	1.35	1.35	1.35	0.996	0.996	0.996
M2	1.35	1.35	1.35	0.995	0.995	0.995
M3	1.11	1.11	1.11	0.997	0.997	0.997
M4	1.34	1.34	1.34	0.999	0.999	0.999
L1	1.21	1.21	1.21	0.992	0.992	0.992
L2	1.22	1.22	1.23	0.994	0.994	0.992
L3	1.24	1.24	1.24	0.991	0.991	0.991
L4	1.26	1.26	1.26	0.991	0.991	0.991
Average	-	-	-	0.994	0.994	0.994

Table 5.4: Objective values that result from the optimization of integral signal group diagrams (Objective), plus their ratios (Ratio) to the objective values that result from the optimization of signal group diagrams that are not necessarily integral (Table 4.5). In this table we have shown the results for each of the 13×3 test cases in the numerical study that maximizes the capacity of the intersection. The first column indicates which intersection is considered, and the second row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations. Furthermore, we have given the (geometric) average of some of the columns.

Ad. real.		0			2			4		
Solver	Intersections	Time	#MILP	#LP	Time	#MILP	#LP	Time	#MILP	#LP
CPLEX	Small	2.74	4.20	58.00	3.11	5.00	70.00	2.99	4.20	57.40
	Medium	2.69	3.75	53.75	2.79	3.50	52.00	3.42	4.00	52.25
	Large	3.90	5.50	67.75	5.32	6.00	76.00	8.38	5.50	71.25
GUROBI	Small	0.10	4.60	54.40	0.15	4.60	58.00	0.31	4.40	55.40
	Medium	0.17	3.75	52.75	0.44	4.00	51.25	4.51	4.25	52.25
	Large	1.11	5.75	67.25	13.78	6.75	76.00	669.09	6.00	70.25
SCIP	Small	0.69	4.40	53.20	0.92	4.20	53.00	1.83	4.20	52.20
	Medium	1.00	3.75	49.25	1.96	3.75	50.25	5.33	4.00	50.50
	Large	4.62	5.50	66.75	22.80	5.50	67.00	119.27	6.25	72.00
Column average		-	4.58	58.12	-	4.81	61.50	-	4.76	59.28

Table 5.5: The (geometric) average computation times, the (arithmetic) average number of MILP problems that were solved, and the (arithmetic) average number of LP problems that were solved for the 13×3 test cases in the numerical study that maximize the capacity of the intersection. We also give the (arithmetic) column average for the number of MILP problems and the number of LP problems that were solved. We distinguish between three types of intersections (small, medium and large), and three types of solvers (CPLEX 12.6.1.0, GUROBI 6.0.5. and SCIP 3.2.0). Furthermore, we vary the number of signal groups that is allowed to have an additional realization, see the first row of this table. If this number of signal groups equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

5.8.3 Minimizing the delay

In this section we consider the minimization of the average delay that road users experience. For each of the test cases we compare the brute force algorithm (BF) with the heuristic method (H); both these methods are proposed in Section 5.6.3. For the brute force algorithm we solve one MILP problem for each integral period duration $T \in [\underline{T}, \overline{T}]$. We however do not solve such a MILP problem if we already know that this MILP problem is infeasible, e.g., when it is not possible to construct the objective function due to the emptiness of an interval (3.11), an interval (4.10), or an interval (4.11). We test three different variants of the heuristic approach. We let the heuristic approach start from one, two and four different starting points; we denote these three variants by H1, H2 respectively H4. These starting points are defined by (5.9). For the heuristic method we set the parameter Q to five. During the second step of the two-step approach we always found a signal group diagram with an average delay that is at most 0.88 percent larger than the average delay of the signal group diagram returned by step one. This implies that when we use the brute force approach, for all the test cases the objective value is at most 0.88 percent larger than that of the optimal integral signal group diagram.

In Table 5.6 we give the objective values that were returned by the methods BF, H1, H2 and H4. In Table 5.7 we give the ratio of the objective value returned by the methods H1, H2 and H4 to the objective value returned by the method BF. We can conclude that the methods H1, H2 and H4 perform very well. The H1-method performs the worst. This

method is on average only 0.1%, 0.4% and 2.0% off when we allow zero, two respectively four signal groups to have an additional realization. The H2-method performs better. When we allow zero or two signal groups to have an additional realization, this method always returns the same solution as the BF method. Furthermore, when we allow four signal groups to have an additional realization, it is only off by 0.1%. The H4-method always finds the same solution as the BF-method.

In Table 5.8 and Table 5.9 we give the computation times respectively the number of (mixed-integer) linear programming problems that were solved for the test cases. The methods H1, H2 and H4 are much faster than the BF-method. This is caused by the relatively small number of MILP problems that were solved in comparison with the BF-method.

Finally, we assess the effect of allowing some signal groups to have multiple realizations on the average delay that road users experience, see Table 5.10. When allowing two signal groups to have an additional realization, the improvement is (on average) approximately two percent. When allowing four signal groups to have an additional realization, the improvement is (on average) approximately seven percent; an exception is the H1-method for which the improvement is only five percent on average.

5.9 Summary

In this chapter we have considered the optimization of integral signal group diagrams. We have considered three different objective functions: minimization of the period duration of the signal group diagram, maximization of the capacity of the intersection, and minimization of the average weighed delay that road users experience at the intersection. For each of these objective functions we have given a two-step approach to search for the optimal integral signal group diagram. During the first step of this approach we search for the optimal signal group diagram that satisfies some structural property; this structural property is also satisfied by the optimal integral signal group diagram. The signal group diagram returned by this first step is not necessarily integral. Therefore, during the second step we solve a MILP problem, which we call the *rounding MILP problem*. With this relatively simple MILP problem we attempt to find an integral signal group diagram by rounding each green time and each red time (of the not necessarily integral signal group diagram) up or down.

We have tested the two-step approach for a large numerical test set. For all the test cases that minimize the period duration and all the test cases that maximize the capacity of the intersection, we were always able to find the optimal integral signal group diagram. When minimizing the delay that road users experience (with the proposed brute force method), we were always able to find an integral signal group diagram for which the average (weighted) delay is at most 0.88 percent larger than that of the optimal integral signal group diagram.

Ad. real.	0				2				4			
	BF	H1	H2	H4	BF	H1	H2	H4	BF	H1	H2	H4
S1	13.31	13.31	13.31	13.31	13.31	13.31	13.31	13.31	12.50	13.21	12.50	12.50
S2	14.28	14.28	14.28	14.28	14.28	14.28	14.28	14.28	14.28	14.28	14.28	14.28
S3	23.24	23.24	23.24	23.24	23.24	23.24	23.24	23.24	23.24	23.24	23.24	23.24
S4	24.33	24.33	24.33	24.33	23.50	23.50	23.50	23.50	19.47	21.33	19.47	19.47
S5	30.83	30.83	30.83	30.83	30.83	30.83	30.83	30.83	28.67	30.83	28.67	28.67
M1	22.14	22.14	22.14	22.14	21.98	22.14	21.98	21.98	21.48	21.48	21.48	21.48
M2	24.57	25.00	24.57	24.57	24.02	25.00	24.02	24.02	24.01	24.30	24.02	24.01
M3	56.46	56.46	56.46	56.46	53.85	53.85	53.85	53.85	51.29	51.29	51.29	51.29
M4	31.71	31.71	31.71	31.71	29.34	29.34	29.34	29.34	25.83	25.83	25.83	25.83
L1	38.13	38.13	38.13	38.13	37.90	37.90	37.90	37.90	36.38	36.38	36.38	36.38
L2	48.36	48.36	48.36	48.36	48.14	48.36	48.14	48.14	44.26	44.26	44.26	44.26
L3	32.30	32.30	32.30	32.30	32.21	32.30	32.21	32.21	31.42	32.30	31.85	31.42
L4	28.58	28.58	28.58	28.58	27.62	27.62	27.62	27.62	27.37	27.37	27.37	27.37

Table 5.6: The objective values for the test cases in the numerical study that minimize the average delay that road users experience. The first column indicates which example is considered, the first row indicates how many signal groups are allowed to have an additional realization (Ad. real.), and the second row indicates which method is used (BF, H1, H2, or H4). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Ad. real.	0			2			4		
Method	H1	H2	H4	H1	H2	H4	H1	H2	H4
S1	1.000	1.000	1.000	1.000	1.000	1.000	1.057	1.000	1.000
S2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
S3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
S4	1.000	1.000	1.000	1.000	1.000	1.000	1.096	1.000	1.000
S5	1.000	1.000	1.000	1.000	1.000	1.000	1.075	1.000	1.000
M1	1.000	1.000	1.000	1.007	1.000	1.000	1.000	1.000	1.000
M2	1.017	1.000	1.000	1.041	1.000	1.000	1.012	1.001	1.000
M3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
M4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
L1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
L2	1.000	1.000	1.000	1.005	1.000	1.000	1.000	1.000	1.000
L3	1.000	1.000	1.000	1.003	1.000	1.000	1.028	1.014	1.000
L4	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Average	1.001	1.000	1.000	1.004	1.000	1.000	1.020	1.001	1.000

Table 5.7: Quality of the proposed heuristic method to minimize the average delay that road users experience. We give the ratio of the objective value of the heuristic methods (H1, H2, H4) to the objective value of the brute force (BF) method. Moreover, we give the (geometric) average per column and in bold we visualize the values that are strictly larger than one. The first column indicates which example is consider, and the first row indicates how many signal groups are allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

We have proposed two different approaches to search for the integral signal group diagram that minimizes the delay that road users experience: a brute force approach, which solves one MILP problem for each integral period duration in the interval $[T, \overline{T}]$, and a heuristic approach. The heuristic approach performs very well. When we start this heuristic method from four different starting points, for our numerical study it always finds the same solution as the brute force approach. When we start this heuristic method from two (one) different starting points, it performs the worst whenever we allow four signal groups to have an additional realization. The objective function returned by the heuristic method is then on average 0.1% (2.0%) larger than the objective function that is returned by the brute force method. The heuristic method is much faster than the brute force algorithm as it only has to solve a MILP problem for a fraction of the integral period duration in the interval $[T, \overline{T}]$.

The next chapter of this thesis concerns the optimization of the layout of an intersection. With the optimization problem proposed in that chapter we are able to answer questions like: what is the layout of the intersection that maximizes the capacity and what is the minimum number of lanes that the intersection requires in order to have sufficient capacity? Subsequently, we consider some practical issues of the proposed optimization framework and we elaborate on how this framework can be used in practice.

Ad. real.		0				2				4			
		BF	H1	H2	H4	BF	H1	H2	H4	BF	H1	H2	H4
Solver	Intersections												
	Small	7.54	1.78	1.86	1.97	12.10	1.95	2.82	3.38	37.76	2.78	5.86	6.82
	Medium	32.24	5.94	8.66	8.07	67.64	8.39	12.70	15.04	268.85	14.22	23.12	36.04
	Large	102.57	4.93	8.56	12.36	337.81	8.85	26.29	37.31	1046.82	31.69	91.77	137.41
Gurobi	Small	1.25	0.11	0.14	0.18	3.88	0.20	0.39	0.55	24.68	0.63	2.02	2.76
	Medium	20.15	1.02	1.92	2.40	72.68	2.84	4.92	6.78	300.38	10.40	19.25	30.92
	Large	83.72	2.84	5.90	8.87	345.92	15.34	33.20	42.97	1440.97	62.64	149.08	208.44
	Small	4.00	0.68	0.74	0.85	13.85	1.03	1.88	2.42	70.31	2.64	7.28	9.36
Scip	Medium	36.17	3.90	7.10	7.00	142.98	8.53	14.81	17.33	726.79	22.73	47.37	74.06
	Large	221.47	8.98	18.88	27.81	1051.85	27.63	75.34	128.71	5674.36	131.68	503.62	693.36

Table 5.8: The (geometric) average computation times (in seconds) for the test cases in the numerical study that minimize the average delay that road users experience. We have distinguished between four methods (BF, H1, H2, H4), three types of intersections (small, medium and large), three types of solvers (CPLEX 12.6.1.0, GUROBI 6.0.5. and SCIP 3.2.0), and we have varied the number of signal groups that is allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Ad. real.	Intersections	#MILP				#LP			
		BF	H1	H2	H4	BF	H1	H2	H4
0	Small	91.0	3.4	4.4	6.4	0.0	36.8	36.8	36.8
	Medium	123.5	4.5	6.3	8.3	0.0	57.8	89.3	74.8
	Large	106.0	4.0	5.7	7.9	0.0	39.7	63.8	66.3
2	Small	91.2	3.6	5.2	7.4	0.0	38.2	56.4	60.8
	Medium	124.5	5.3	7.3	9.5	0.0	66.5	86.5	83.5
	Large	107.0	4.8	7.3	9.3	0.0	38.3	68.3	60.2
4	Small	91.2	3.7	5.6	7.8	0.0	40.2	68.0	68.0
	Medium	124.3	5.3	7.0	9.8	0.0	69.5	83.7	100.3
	Large	107.0	5.3	7.8	10.3	0.0	47.2	75.4	81.0
Column average		107.3	4.4	6.3	8.5	0.0	48.2	69.8	70.2

Table 5.9: The (arithmetic) average number of MILP problems that were solved, and the (arithmetic) average number of LP problems that were solved for the test cases in the numerical study that minimize the average delay that road users experience. We also give the (arithmetic) column averages. We have distinguished between three methods (BF, H1, H2, H4), three types of intersections (small, medium and large), and we have varied the number of signal groups that is allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Ad. real.	2				4			
	BF	H1	H2	H4	BF	H1	H2	H4
S1	1.000	1.000	1.000	1.000	1.065	1.008	1.065	1.065
S2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
S3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
S4	1.035	1.035	1.035	1.035	1.250	1.140	1.250	1.250
S5	1.000	1.000	1.000	1.000	1.075	1.000	1.075	1.075
M1	1.007	1.000	1.007	1.007	1.031	1.031	1.031	1.031
M2	1.023	1.000	1.023	1.023	1.023	1.029	1.023	1.023
M3	1.048	1.048	1.048	1.048	1.101	1.101	1.101	1.101
M4	1.081	1.081	1.081	1.081	1.227	1.227	1.227	1.227
L1	1.006	1.006	1.006	1.006	1.048	1.048	1.048	1.048
L2	1.005	1.000	1.005	1.005	1.093	1.093	1.093	1.093
L3	1.003	1.000	1.003	1.003	1.028	1.000	1.014	1.028
L4	1.035	1.035	1.035	1.035	1.044	1.044	1.044	1.044
Average	1.018	1.015	1.018	1.018	1.073	1.053	1.072	1.073

Table 5.10: Improvement in objective value (average delay) for the methods BF, H1, H2 and H4 when we allow some signal groups to have two realizations (with respect to the case that each signal group has a single realization). We also give the (geometric) average per column and have visualized all values that are strictly larger than one in bold. The first row indicates how many signal groups are allowed to have an additional realization. If this number of signal groups equals $k = 0, 2, 4$, the k most heavily loaded signal groups are allowed to have an additional realization; for each of these signal groups the optimization decides whether this signal group should have one or two realizations.

Optimizing intersection layout

6.1 Introduction

With the optimization problem formulated in Chapter 4 we are able to assess the capacity of the intersection. In other words, we can find the largest growth factor of the arrival rates that is sustainable.

If the capacity is insufficient, a few measures are possible. One possible measure is to increase the capacity of the intersection by changing its layout. For example, we may be able to increase the capacity by changing the lane-use arrows marked on each of the arrival lanes. Changing these lane-use arrows is often a relatively small change in the layout of the intersection. The questions that arise: what should these lane-use arrows be such that the capacity of the intersection is maximized? What is the corresponding signal group diagram for which this largest growth factor is sustainable?

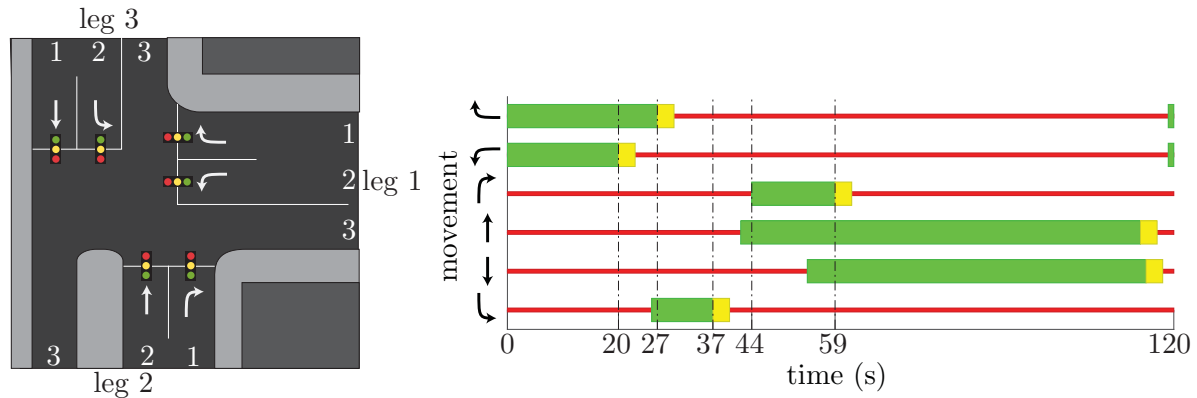
If changing the lane-use arrows still results in insufficient capacity, a more drastic measure may be needed. We can then, in addition to changing the lane-use arrows, add lanes to the intersection. Questions that arise are: what is the minimum number of lanes required for the intersection to have sufficient capacity? Which of these lanes should be arrival lanes and which ones should be departure lanes? What should the lane-use arrows on the arrival lanes be? What is the corresponding signal group diagram? To answer these questions the layout of the intersection has to be optimized simultaneously with a signal group diagram that specifies when each of the traffic lights at the intersection is green, yellow and red. Consider, for example, the three different layouts of a T-junction that we have visualized in Figure 6.1. For the second layout variant (Figure 6.1b), lane 2 of leg 3 allows both a through movement and a left-turn movement. As a consequence, both arrival lanes of this leg must have the same indication at all times; otherwise a vehicle following the through movement of leg 3 might be able to drastically reduce its waiting time by switching lanes, which can result in a dangerous situation. However, this same indication is not required for the first and the third layout variant; for these variants no lane (on leg

This chapter is based on the following paper: Fleuren, S. and Lefeber, E. (in preparation). Lane-based optimization of signal group diagrams using cycle periodicity constraints

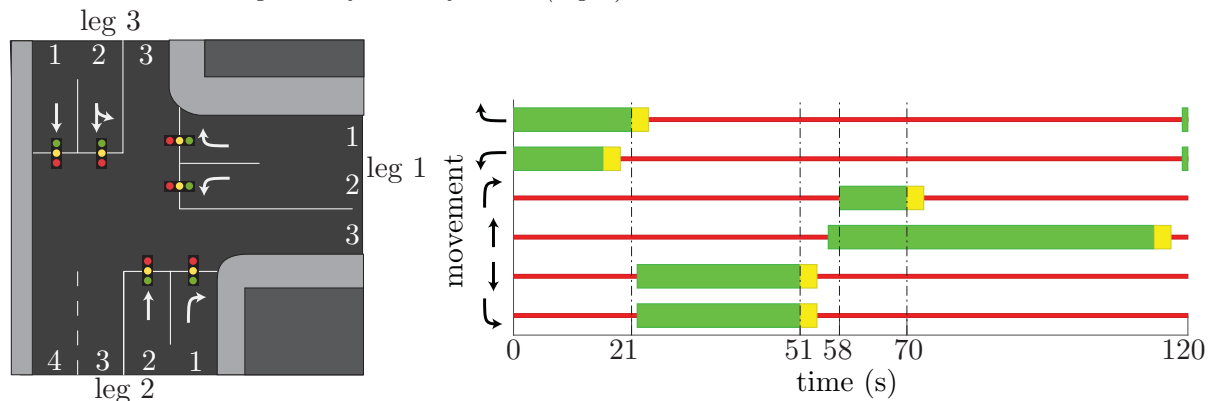
3) exists that accommodates both the through movement and the left-turn movement. This implies that the feasibility of a signal group diagram depends on the layout of the intersection. Therefore, to answer the aforementioned questions, e.g., what is the number of lanes required at the intersection?, we require the simultaneous optimization of the layout of the intersection and a signal group diagram. In this chapter we formulate an optimization problem that is able to perform this simultaneous optimization. Possible objectives are the minimization of the number of lanes at the intersection (we find the minimum number of lanes required for the intersection to have sufficient capacity) and the maximization of the capacity of the intersection (maximization of the growth factor β of the arrival rates). Both these problems can be formulated as a MILP problem.

Some methods in literature are already able to simultaneously optimize the layout of the intersection and the signal group diagram, e.g., (Wong and Heydecker, 2011; Wong and Wong, 2003; Wong, 1996; Yan et al., 2014). These existing methods are referred to as lane-based approaches. We elaborate on the novelty of the optimization problem formulated in this chapter. First, the existing lane-based approaches extend the currently existing group-based approaches. In Chapter 3 we have compared these group-based approaches with the novel formulation proposed in this thesis; the novel formulation seems to be superior. Therefore, in this chapter we extend this novel formulation. In Chapter 4 we have extended this formulation to also optimize the number of realizations of each signal group. As a consequence, in this chapter we are also able to optimize the number of realizations of each traffic light. Second, in this chapter we also optimize the number of lanes of each leg, which is considered to be fixed by the currently existing lane-based methods. This makes it, for example, possible to find the smallest intersection (the intersection with the smallest number of lanes) that has sufficient capacity. Third, to model the constraints that a signal group diagram has to satisfy, the existing lane-based methods keep track of one traffic light for each movement and one traffic light for each of the arrival lanes. In this chapter we show that it suffices to keep track of only one traffic light for each movement. From their signal timings we can obtain the signal timings of all traffic lights at the intersection. As a consequence, the size of the formulated optimization problem is reduced and the solution time required to solve the problem is expected to decrease.

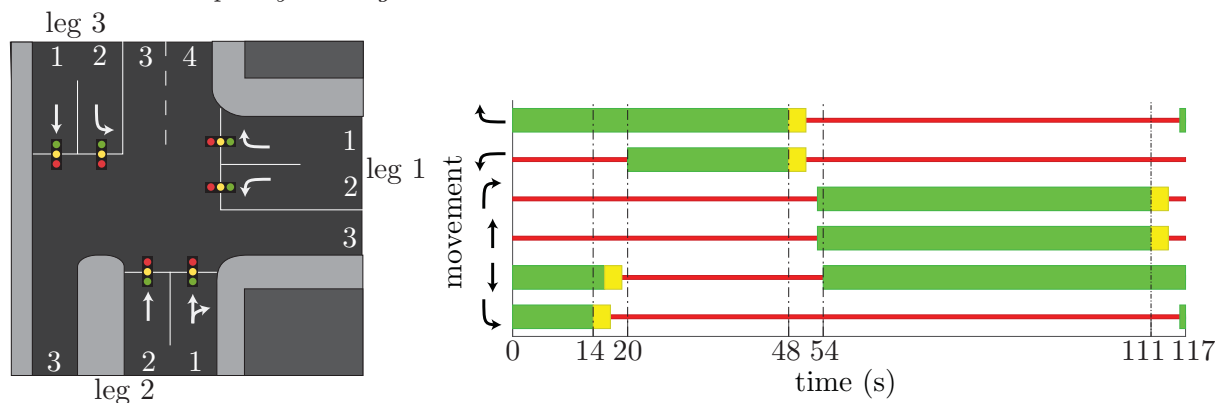
This chapter is structured as follows. First, in Section 6.2 we elaborate on the input data that is required. Thereupon, we formulate the MILP problem in Section 6.3. Subsequently in Section 6.4 we perform a numerical study and, finally, in Section 6.5 we give a summary.



(a) Layout variant 1 of a T-junction (left) and the (integral) signal group diagram associated with the maximum growth factor of 1.180 (right).



(b) Layout variant 2 of a T-junction (left) and the (integral) signal group diagram associated with the maximum growth factor of 0.953 (right). Since the movements of leg 3 share a lane, these movements must receive green simultaneously. As a consequence, the through movement of this leg may not be green simultaneously with the two movements of leg 2 (even though they are not conflicting), which results in a decrease in the capacity of the intersection; this layout variant has a capacity shortage.



(c) Layout variant 3 of a T-junction (left) and the (integral) signal group diagram associated with the maximum growth factor of 1.585 (right). Since the movements of leg 2 share a lane, these movements must receive green simultaneously. As a consequence, the right-turn movement of this leg may not be green simultaneously with the right-turn movement of leg 1 and the left-turn movement of leg 3. However, as the arrival rate for this right-turn movement of leg 3 is small and now two lanes provide a through movement, this layout variant has a large capacity.

Figure 6.1: Three possible layouts for the T-junction of Figure 3.1a; the data of this intersection can be found in Section 3.2.1.

6.2 Input data

In this section we summarize all input data required for the optimization problem formulated in this chapter.

Legs, lanes and lane-use arrows

We require the number of legs at the intersection, which we denote by N^{leg} . We define \mathcal{L} as the set of legs at the intersection:

$$\mathcal{L} := \{1, \dots, N^{\text{leg}}\}.$$

For each of these legs $i \in \mathcal{L}$ we require an upper bound on the number of lanes (arrival lanes plus departure lanes) \bar{L}_i that leg i is allowed to have; via optimization it is determined how many of these lanes are actually used. We define \mathcal{Q}_i^v as the set of lanes at leg i :

$$\mathcal{Q}_i^v := \{1, \dots, \bar{L}_i\}.$$

The set \mathcal{Q}_i^v can be interpreted as a set of queues that model the vehicular traffic that is waiting at leg $i \in \mathcal{L}$; each lane $l \in \mathcal{Q}_i^v$ is a single (first-in-first-out) queue. The arrival rate of traffic at such a queue $l \in \mathcal{Q}_i^v$ is only allowed to be non-zero whenever lane l of leg i is an arrival lane; the optimization automatically distributes all traffic arriving at the intersection amongst the different lanes.

For each leg $i \in \mathcal{L}$ we require the number of vehicular movements N_i^{arrow} that arrive at this leg. With each of these vehicular movements we associate a distinct lane-use arrow. We number these lane-use arrows from 1 until N_i^{arrow} . This numbering is based on the angle α associated with each of these arrows (see Figure 6.2 for a visualization of this angle α); lane-use arrow 1 has the largest angle α (sharpest turn to the right) and lane-use arrow N_i^{arrow} has the smallest angle α (sharpest turn to the left). We define \mathcal{A}_i to be the set of lane-use arrows of leg $i \in \mathcal{L}$:

$$\mathcal{A}_i := \{1, \dots, N_i^{\text{arrow}}\}.$$









Figure 6.2: Visualization of the angle α that is used to number the lane-use arrows of a leg. On the left we have visualized a right-turn arrow. For this arrow it holds that $\alpha \approx 270$. On the right we have visualized a left-turn arrow. For this arrow it holds that $\alpha \approx 90$.

The layout variants in Figure 6.1 have three legs ($N^{\text{leg}} = 3$). Each of these legs is allowed to have at most four lanes ($\bar{L}_i = 4$, $i = 1, 2, 3$), and each leg has two distinct lane-use arrows ($N_i^{\text{arrow}} = 2$, $i = 1, 2, 3$). Therefore, for this example we have:

$$\begin{aligned}
\mathcal{L} &:= \{1, 2, 3\}, \\
\mathcal{Q}_i^v &:= \{1, 2, 3, 4\}, & i \in \mathcal{L}, \\
\mathcal{A}_i &:= \{1, 2\}, & i \in \mathcal{L}.
\end{aligned}$$

Below we visualize the numbering of the lane-use arrows for this example:







leg	Lane-use arrow	
	1	2
1		
2		
3		

Remark 6.1. *Perhaps it is not possible to reach each of the legs in $\mathcal{L} \setminus \{i\}$ from leg $i \in \mathcal{L}$; a movement from leg i to leg j may for example be prohibited because leg j is required to be a one-way street (traffic is only allowed to enter the intersection at this leg j). In that case, leg j is not allowed to have any departure lanes.*

Movements

At an intersection several movements are allowed. We distinguish between two types of movements. On the one hand, we have pedestrian and cyclist movements; these movements cross a leg via a pedestrian crossing or cyclist crossing. On the other hand, we have vehicular movements; each such vehicular movement is characterized by the leg at which the vehicles arrive and the leg at which the vehicles depart. We require a set of movements \mathcal{M} at the intersection, which we split into a set \mathcal{M}^{pc} of pedestrian and cyclist movements and a set \mathcal{M}^v of vehicular movements. We define $i(m)$ as the leg at which vehicular movement $m \in \mathcal{M}^v$ enters the intersection and we define $j(m)$ as the lane-use arrow that this vehicular movement follows when crossing the intersection. Furthermore, we define $\theta(i, j)$ as the leg at which vehicles arriving at leg i and following lane-use arrow j (of leg i) depart.

For the example in Figure 6.1 we have $\mathcal{M}^{\text{pc}} = \emptyset$ and $\mathcal{M}^v = \{1, 3, 4, 5, 11, 12\}$; we have numbered the vehicular movements in accordance with the standard Dutch numbering of signal groups. For this example we have:

movement (m)	1	3	4	5	11	12
visualization						
arrival leg ($i(m)$)	1	1	2	2	3	3
lane-use arrow ($j(m)$)	1	2	1	2	1	2
departure leg ($\theta(i(m), j(m))$)	3	2	1	3	2	1

Queues

The traffic that is waiting at the intersection is modelled by using queues $q \in \mathcal{Q}$. The set \mathcal{Q} is comprised of the queues in \mathcal{Q}_i^v , $i \in \mathcal{L}$ which are used to model the vehicular traffic that is waiting at the intersection, and the queues in \mathcal{Q}^{pc} , which are used to model the traffic waiting at the pedestrian and cyclist crossings. Let $\mathcal{Q}_m^{\text{pc}}$ denote the set of queues that is used to model the traffic waiting at the intersection for this movement $m \in \mathcal{M}^{\text{pc}}$. For the example (Figure 6.1) we have no such pedestrian and cyclist movements.

Starting lost times, ending lost times, and yellow times

To make the conversion from the indications (green, yellow and red) to the effective modes (effective green and effective red) and vice versa we require starting lost times, ending lost times, and yellow times. In this chapter we require these starting lost times l_m^s , ending lost times l_m^e , and yellow times l_m^e for each movement $m \in \mathcal{M}$. We do however assume that these lost times and yellow times are the same for each two vehicular movements $m, m' \in \mathcal{M}^v$ that arrive at the same leg ($i(m) = i(m')$). As in Chapter 3 it is also possible to require these lost times and yellow times for each queue $q \in \mathcal{Q}$; we then do not require an assumption on the lost times and the yellow times. For the example we have the following yellow times and lost times:

$$\begin{aligned}
 l_m^s &= 1, & m \in \mathcal{M}, \\
 l_m^e &= 1, & m \in \mathcal{M}, \\
 Y_m^s &= 3, & m \in \mathcal{M},
 \end{aligned}$$

Remark 6.2. *We assume that the lost times and yellow times are the same for each two vehicular movements $m, m' \in \mathcal{M}^v$ that arrive at the same leg ($i(m) = i(m')$). As a consequence, we can force two movements that are both permitted by the same lane, say lane l of leg i , to have the same indication by forcing their effective green times and effective red times to be the same. These effective green times and effective red times then also specify when lane l of leg i is effective green and effective red.*

Consider the case that the movements permitted by lane l of leg i do not have the same lost times and yellow times. For example consider the case that two movements, say movement m and movement m' , are allowed on lane l of leg i . Let these two movements

have different lost times and yellow times. The effective green times and effective red times associated with movements m and m' may then differ even though their indications are the same. When the effective green times and effective red times of these movements are different, it is not well-defined when lane l of leg i is effective green and when it is effective red; is lane l of leg i effective green simultaneous with movement m or simultaneous with movement m' ? A possibility is to define the effective green times and effective red times of lane l of leg i to be a convex combination of the effective green times and effective red times of movement m and movement m' . Assume for example that the arrival rate of traffic at lane l of leg i is λ_m ($\lambda_{m'}$) for movement m (m'). Then we can take the k th effective green (effective red) time of lane l of leg i to be $\alpha := \lambda_m/(\lambda_m + \lambda_{m'})$ times the k th effective green (effective red) time of movement m plus $(1 - \alpha)$ times the k th effective green (effective red) time of movement m' . However, unfortunately, as we also optimize the distribution of traffic amongst the different arrival lanes, the resulting optimization problem is then non-linear. Therefore, we assume that the lost times and yellow times are the same for each two vehicular movements $m, m' \in \mathcal{M}^v$ that arrive at the same leg ($i(m) = i(m')$).

Arrival rates and saturation rates of pedestrian and cyclist movements

The queues $q \in \mathcal{Q}^{pc}$ are used to model the traffic waiting at the pedestrian and cyclist crossings of the intersection. For each of the queues $q \in \mathcal{Q}^{pc}$ we require the arrival rate λ_q , which indicates how much traffic arrives at queue q . Furthermore, for each of these queues we require the saturation flow rate μ_q . We define the load ρ_q of queue $q \in \mathcal{Q}^{pc}$ to be the ratio of the arrival rate and the saturation flow rate, i.e., $\rho_q := \lambda_q/\mu_q$. For the example (Figure 6.1) we have no such pedestrian movements.

Arrival rates and saturation flow rates of vehicular movements

For each leg $i \in \mathcal{L}$ we require arrival rates $\lambda_{i,j} > 0$ $j \in \mathcal{A}_i$ (in PCE/s). The arrival rate $\lambda_{i,j}$ indicates how much traffic arriving at leg i follows lane-use arrow j (on average per second); the optimization decides automatically how to distribute the arrival rate $\lambda_{i,j}$ amongst the different lanes of leg i .

We also need the saturation flow rate for each (possible) arrival lane $l \in \mathcal{Q}_i^v$, $i \in \mathcal{L}$ at the intersection. This saturation flow rate may depend on the movements that are allowed on this lane; for example vehicles making a right-turn tend to require more time to depart than vehicles going straight. Therefore, for each leg $i \in \mathcal{L}$, each lane $l \in \mathcal{Q}_i^v$ and each lane-use arrow $j \in \mathcal{A}_i$, we require the saturation flow rate $\mu_{i,l,j}$; the input $\mu_{i,l,j}$ denotes the saturation flow rate of lane l of leg i when this lane is solely used by traffic following lane-use arrow j . For the example, the vehicular movements have the following arrival and departure rates:

movement (m)	1	3	4	5	11	12
arrival rate ($\lambda_{i(m),j(m)}$)	$\frac{320}{3600}$	$\frac{280}{3600}$	$\frac{180}{3600}$	$\frac{980}{3600}$	$\frac{820}{3600}$	$\frac{150}{3600}$
saturation flow rate ($\mu_{i(m),1,j(m)}$)	$\frac{1615}{3600}$	$\frac{1805}{3600}$	$\frac{1615}{3600}$	$\frac{1900}{3600}$	$\frac{1900}{3600}$	$\frac{1805}{3600}$
saturation flow rate ($\mu_{i(m),2,j(m)}$)	$\frac{1615}{3600}$	$\frac{1805}{3600}$	$\frac{1615}{3600}$	$\frac{1900}{3600}$	$\frac{1900}{3600}$	$\frac{1805}{3600}$
saturation flow rate ($\mu_{i(m),3,j(m)}$)	$\frac{1615}{3600}$	$\frac{1805}{3600}$	$\frac{1615}{3600}$	$\frac{1900}{3600}$	$\frac{1900}{3600}$	$\frac{1805}{3600}$

Signal group diagram

We also optimize a signal group diagram. This signal group diagram specifies exactly when each movement is green, yellow and red. For each movement $m \in \mathcal{M}$ we have a lower bound \underline{g}_m (\underline{r}_m) and an upper bound \bar{g}_m (\bar{r}_m) on each effective green (effective red) time of movement $m \in \mathcal{M}$. Furthermore, for each movement $m \in \mathcal{M}$ we require a lower bound \underline{K}_m on its number of realizations and an upper bound \bar{K}_m on its number of realizations. We define the following sets:

$$\begin{aligned}\underline{\mathcal{K}}_m &:= \{1, \dots, \underline{K}_m\}, \\ \bar{\mathcal{K}}_m &:= \{1, \dots, \bar{K}_m\}, \\ \mathcal{K}_m^d &:= \bar{\mathcal{K}}_m \setminus \underline{\mathcal{K}}_m.\end{aligned}$$

Moreover, a set $\Psi_{\mathcal{M}}$ of conflicting movements needs to be provided, i.e., movement m conflicts with movement m' if and only if $\{m, m'\} \in \Psi_{\mathcal{M}}$. For each two conflicting movements $\{m, m'\} \in \Psi_{\mathcal{M}}$ a clearance time $\underline{c}_{m,m'}$, which is the minimum amount of time between movement m switching to effective red and movement m' switching to effective green, needs to be provided. Moreover, we need a lower bound \underline{T} and an upper bound \bar{T} on the period duration of the signal group diagram. From these inputs we can construct a constraint graph G analogously to its definition used in Chapter 4. To this end, we introduce the event \widehat{m}_k (\bar{m}_k), which denotes the start (end) of realization k of movement m . The constraint graph $G = (V, A)$ is:

$$\begin{aligned}V &= \{\widehat{m}_k \mid m \in \mathcal{M}, k \in \bar{\mathcal{K}}_m\} \cup \{\bar{m}_k \mid m \in \mathcal{M}, k \in \bar{\mathcal{K}}_m\}, \\ A &= A_g \cup A_r \cup A_c,\end{aligned}$$

where,

$$\begin{aligned}A_g &:= \{(\widehat{m}_k, \bar{m}_k) \mid m \in \mathcal{M}, k \in \bar{\mathcal{K}}_m\}, \\ A_r &:= \{(\bar{m}_{k-1}, \widehat{m}_k) \mid m \in \mathcal{M}, k \in \bar{\mathcal{K}}_m\}, \\ A_c &:= \{(\bar{m}_k, \widehat{m}_{k'}) \mid \{(m, k), (m', k')\} \in \Psi_R\}, \\ \Psi_R &:= \{\{(m, k), (m', k')\} \mid \{m, m'\} \in \Psi_{\mathcal{S}}, k \in \bar{\mathcal{K}}_m, k' \in \bar{\mathcal{K}}_{m'}\}.\end{aligned}$$

In the above definition we use $\bar{m}_0 := \bar{m}_{\bar{K}_m}$. We require an integral cycle basis \mathcal{B} of this constraint graph $G = (V, A)$; this integral cycle basis can be obtained with the method proposed in Chapter 4. For the example we have:

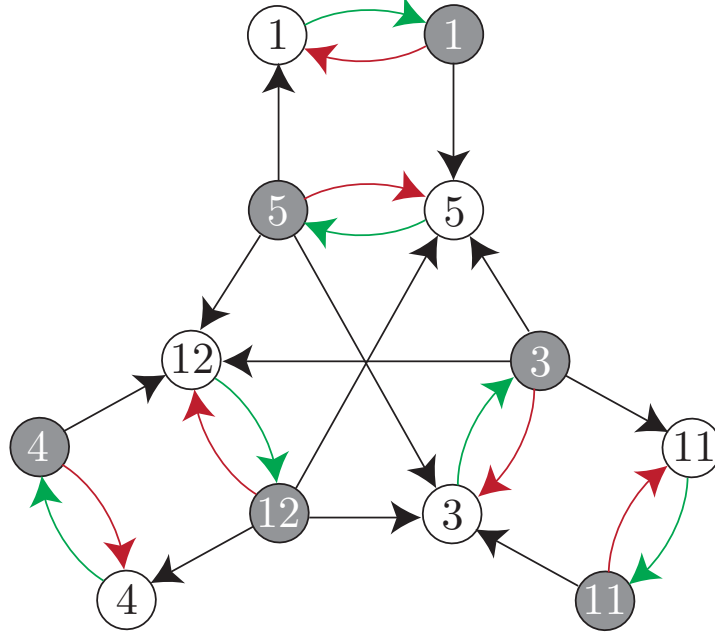


Figure 6.3: The constraint graph $G = (V, A)$ associated with the T-junction whose layout variants are given in Figure 6.1. This constraint graph is based on $\bar{K}_m = 1$ for all movements. The arcs in green (red) visualise effective green (effective red) intervals. The arcs in black represent clearance intervals. Each white (grey) vertex with the text m represents the vertex \textcircled{m}_1 (\textcircled{m}_1).

movement (m)	1	3	4	5	11	12
Minimum effective green time (\underline{g}_m)	6	6	6	6	6	6
Maximum effective green time (\bar{g}_m)	∞	∞	∞	∞	∞	∞
Minimum effective red time (\underline{r}_m)	6	6	6	6	6	6
Maximum effective red time (\bar{r}_m)	∞	∞	∞	∞	∞	∞
Minimum number of realizations (\underline{K}_m)	1	1	1	1	1	1
Maximum number of realizations (\bar{K}_m)	1	1	1	1	1	1

The conflicts between the movements are as follows:

$$\Psi_{\mathcal{M}} = \{\{1, 5\}, \{3, 5\}, \{3, 11\}, \{3, 12\}, \{4, 12\}, \{5, 12\}\}.$$

The corresponding minimum clearance times are given below:

$$\begin{aligned} \underline{c}_{1,5} &= 4, & \underline{c}_{3,5} &= 4, & \underline{c}_{3,11} &= 5, & \underline{c}_{3,12} &= 5, & \underline{c}_{4,12} &= 4, & \underline{c}_{5,12} &= 4, \\ \underline{c}_{5,1} &= 4, & \underline{c}_{5,3} &= 4, & \underline{c}_{11,3} &= 3, & \underline{c}_{12,3} &= 5, & \underline{c}_{12,4} &= 6, & \underline{c}_{12,5} &= 4. \end{aligned}$$

We consider a lower bound of 30 seconds and an upper bound of 120 seconds on the period duration of the signal group diagram ($\underline{T} = 30$ and $\bar{T} = 120$). The constraint graph G associated with this example is visualized in Figure 6.3.

Remark 6.3. In this chapter we assume that the minimum clearance time $\underline{c}_{m,m'}$ is a constant, i.e., the minimum clearance times do not depend on the layout of the intersection.

In practice however the minimum clearance time between movement m and movement m' is a safety measure that is based on the distance between the stop line at movement m (m') and the conflict (the place where the two movements cross or merge) amongst other things. This distance to the conflict depends on the layout of the intersection, which is also optimized in this chapter. In Chapter 7 we consider the clearance times to depend on the layout of the intersection and propose a heuristic (engineering) approach that iteratively updates these clearance times. During each iteration we solve an optimization problem as formulated in this chapter.

6.3 Formulating the MILP problem

In this section we consider the formulation of a MILP problem with which we can simultaneously optimize the layout of the intersection and a signal group diagram that specifies when each of the movements receives a green, yellow and red indication; we still refer to such a diagram as a *signal group diagram* even though it visualizes the indication per movement and not per signal group. From this signal group diagram we can obtain the indication of each of the traffic lights at the intersection; for each arrival lane, the indication of its traffic light equals the indication of any movement allowed on this lane; all movements allowed on the same lane must have the same indication. Throughout this section we consider an example: the optimization of the layout of the T-junction introduced in Section 3.2.1. In Figure 6.1 we give three layout variants of this T-junction with their corresponding signal group diagrams; these integral signal group diagrams maximize the growth factor of the arrival rates that the intersection is able to handle.

6.3.1 Design variables

In the previous section we have elaborated on the required inputs. In this section we elaborate on the design variables of the optimization problem. To clearly distinguish the design variables from the other variables and parameters we visualize them in bold.

Lanes and lane-use arrows

We model the layout of the intersection by using binary variables. For each lane $l \in \mathcal{Q}_i^v$ of leg $i \in \mathcal{L}$ the binary-valued design variable $\Delta_{i,l}^a$ indicates if it is an arrival lane or not; the binary variable equals one whenever the lane is an arrival lane and it equals zero otherwise. Furthermore, the binary variable $\Delta_{i,l,j}$ indicates if lane $l \in \mathcal{Q}_i^v$ of leg $i \in \mathcal{L}$ is equipped with lane-use arrow $j \in \mathcal{A}_i$. In Figure 6.4 we zoom in on leg 3 of the layout variant in Figure 6.1b. In Table 6.1 we give the values of the binary variables associated with this leg. From these values we can directly obtain that this leg has two arrival lanes but from these variables we cannot obtain the number of departure lanes of this leg; the number of departure lanes required at leg i depends on the number of lanes that provide

Lane (l)	1	2	3	4
$\Delta_{3,l}^a$	1	1	0	0
$\Delta_{3,l,1}$	1	1	0	0
$\Delta_{3,l,2}$	0	1	0	0

Table 6.1: The binary-valued design variables that are associated with the leg of Figure 6.4.

a movement towards leg i . For example, for the second layout variant (Figure 6.1b) only one departure lane is required at leg 3; traffic from leg 2 requires only one departure lane (only one lane at leg 1 accommodates a right-turn movement to leg 3) and traffic from leg 2 also requires only one departure lane (only one lane at leg 2 accommodates a through movement to leg 3). Thus, the number of departure lanes required at a leg i does not follow from the binary variables associated with leg i but follows from the binary variables associated with the other legs at the intersection. To keep track of the number of departure lanes (exits) of leg $i \in \mathcal{L}$ we use a real-valued design variable \mathbf{e}_i . Furthermore, we have a real-valued design variable \mathbf{a}_i that denotes the number of arrival lanes at leg $i \in \mathcal{L}$; its value directly depends on the variables $\Delta_{i,l}^a$, $l \in \mathcal{Q}_i^v$. For the leg visualized in Figure 6.4, we have $\mathbf{a}_3 = 2$ and $\mathbf{e}_3 = 1$.

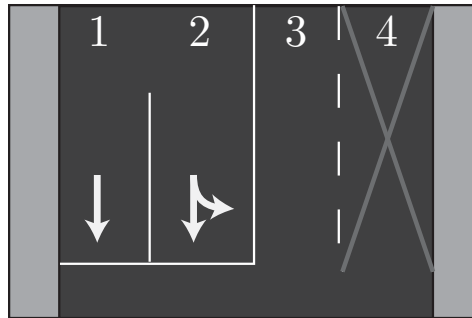


Figure 6.4: Zooming in on leg 3 of the layout variant of Figure 6.1b. This leg has two arrival lanes and one departure lane; lane 4 is not used.

Remark 6.4. The number of arrival lanes \mathbf{a}_i and the number of departure lanes \mathbf{e}_i should be integral-valued. However, during the optimization we can relax these variables to be real-valued; these variables are then either integral-valued or can be rounded (downwards) to an integral variable without destroying feasibility of the solution.

Arrival rates

For each leg $i \in \mathcal{L}$ and each lane-use arrow $j \in \mathcal{A}_i$ we are given its arrival rate $\lambda_{i,j}$. However, we do not yet know how this traffic is distributed amongst the different lanes of the intersection. To this end, we introduce real-valued design variables $\lambda_{i,l,j}$, $i \in \mathcal{L}$, $l \in \mathcal{Q}_v^i$, $j \in \mathcal{A}_i$; the variable $\lambda_{i,l,j}$ denotes the arrival rate of traffic following lane use

arrow j at lane l of leg i . Besides the distribution of traffic, we may also choose a growth factor β of the arrival rate $\lambda_{i,j}$, $i \in \mathcal{L}$, $j \in \mathcal{A}_i$.

Signal group diagram

We optimize the signal group diagram analogously to Chapter 4. Thus, we have a real-valued design variable for the reciprocal of the period duration \mathbf{T}' . Moreover, we have one real-valued design variable $\gamma(\varepsilon_1, \varepsilon_2)$ for each arc $(\varepsilon_1, \varepsilon_2) \in A$ of the constraint graph $G = (V, A)$; this variable expresses the duration between event ε_1 and event ε_2 as a fraction of the period duration $T := 1/\mathbf{T}'$.

In addition to the real-valued design variables, we have a binary-valued design variable $\mathbf{b}_{m,k}$ for each realization $k \in \mathcal{K}_i^d$ of movement $m \in \mathcal{M}$; this binary variable indicates whether movement m has k (or more) realizations or not. Furthermore, we have $d := |A| - |V| + \nu(G)$ integral variables $\mathbf{z}_{\mathcal{C}}$; for each cycle \mathcal{C} in the integral cycle basis \mathcal{B} we have one integer design variable $\mathbf{z}_{\mathcal{C}}$; these integer design variables are used to model the periodicity of the signal group diagram.

6.3.2 Objective function

One possible objective is to find the smallest intersection that has sufficient capacity. To this end we fix the growth factor β ; the intersection has sufficient capacity whenever this growth factor is sustainable. The objective function is then:

$$\text{minimize } \sum_{i \in \mathcal{L}} (\mathbf{a}_i + \mathbf{e}_i).$$

Another possibility is to find the intersection that has the most capacity. We then maximize the growth factor β , i.e., the objective function is:

$$\text{maximize } \beta.$$

Remark 6.5. *When maximizing the capacity of the intersection it might not be desirable to change the arrival lanes and departure lanes; this can be a relatively large (and costly) change in the layout of the intersection. We can then fix the following variables: each binary $\Delta_{i,l}^a$ that indicates if lane $l \in \mathcal{Q}_i^v$ of leg $i \in \mathcal{L}$ is an arrival lane, the number of arrival lanes \mathbf{a}_i of each leg $i \in \mathcal{L}$ and the number of departure lanes \mathbf{e}_i of each leg $i \in \mathcal{L}$. In this case we only optimize the lane-use arrows, the growth factor of the arrival rates, the distribution of traffic amongst the different arrival lanes, and a signal group diagram.*

6.3.3 Linear constraints

In this section we give all (linear) constraints on the design variables of the optimization problem.

Arrival lanes on the right-hand side

Since we drive on the right-hand side of the road, the arrival lanes must be positioned on the right-hand side of each leg (when facing towards the intersection). This can be modelled with the following constraints:

$$\Delta_{i,l}^a \geq \Delta_{i,l+1}^a, \quad i \in \mathcal{L}, \quad l \in \mathcal{Q}_i^v \setminus \{\bar{L}_i\}. \quad (6.1)$$

These constraints force lane l of leg i (and as a result also lanes $l-1, l-2, \dots, 1$) to be an arrival lane whenever lane $l+1$ of leg i is an arrival lane.

Remark 6.6. *When vehicles drive on the left-hand side of the road, we consider a different numbering of the lanes and lane-use arrows. The lanes of a leg are then numbered from left to right instead of from right to left (when facing towards the intersection). Furthermore, the lane-use arrows are numbered from 1 until N_i^{arrow} based on their angles α (see Figure 6.2 for a visualization of this angle α), where lane-use arrow 1 has the smallest angle α (sharpest turn to the left) and lane-use arrow N_i^{arrow} has the largest angle α (sharpest turn to the right). The constraints (6.1) then model that the arrival lanes are positioned on the left-hand side of a leg.*

Lane-use arrows only on arrival lanes

Only arrival lanes may be equipped with a lane-use arrow and, furthermore, each arrival lane must be equipped with at least one lane-use arrow:

$$\Delta_{i,l}^a \leq \sum_{j \in \mathcal{A}_i} \Delta_{i,l,j} \leq L \Delta_{i,l}^a, \quad i \in \mathcal{L}, \quad l \in \mathcal{Q}_i^v, \quad (6.2)$$

where L is some large positive number; in this case $L = |\mathcal{A}_i|$ is sufficiently large. On the one hand, the above constraint forces the number of lane-use arrows on lane l of leg i ($\sum_{j \in \mathcal{A}_i} \Delta_{i,l,j}$) to be zero whenever it is not an arrival lane. On the other hand, it forces this number of lane-use arrows to be at least one if this lane is an arrival lane.

Order of lane-use arrows

For safety reasons some assignments of lane-use arrows across adjacent lanes are not allowed. See for example Figure 6.5; this figure gives one assignment that is allowed and one that is not allowed. The assignment on the right-hand side of Figure 6.5 is not allowed since the through movement (lane-use arrow 2) is positioned on lane 2 while the right-turn arrow (lane-use arrow 1) is positioned on lane 3. To put it differently: some ordering amongst these lane-use arrows exists; when lane l is equipped with a lane-use arrow j , lane $l+1$ is not allowed to have lane-use arrows $1, \dots, j-1$. We formulate this constraint as follows:

$$\sum_{j'=1}^{j-1} \Delta_{i,l+1,j'} \leq L(1 - \Delta_{i,l,j}), \quad i \in \mathcal{L}, \quad j \in \mathcal{A}_i \setminus \{1\}, \quad l \in \mathcal{Q}_i^v \setminus \{\bar{L}_i\} \quad (6.3)$$

where in this case $L = (j - 1)$ is sufficiently large.

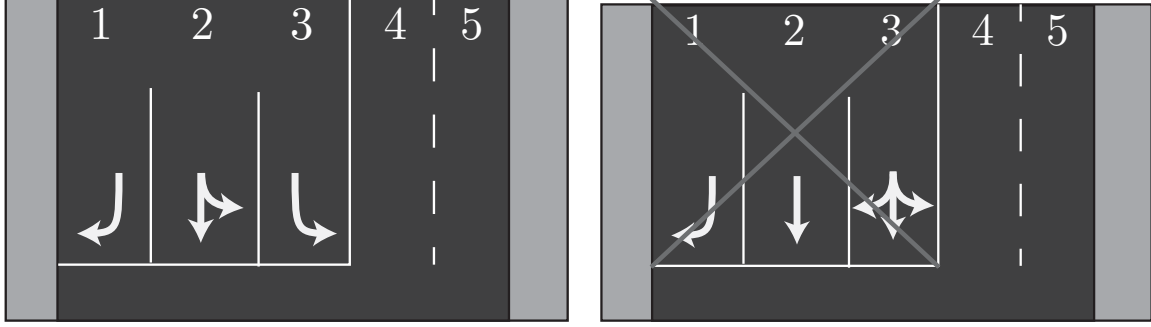


Figure 6.5: An assignment of the lane-use arrows that is allowed (left) and an assignment that is not allowed (right).

Number of arrival lanes

The number of arrival lanes \mathbf{a}_i of leg i can directly be obtained from the binary variables $\Delta_{i,l}^a$, $l \in \mathcal{Q}_i^v$:

$$\mathbf{a}_i = \sum_{l \in \mathcal{Q}_i^v} \Delta_{i,l}^a, \quad i \in \mathcal{L}. \quad (6.4)$$

Number of departure lanes

A total of $\sum_{l \in \mathcal{Q}_i^v} \Delta_{i,l,j}$ arrival lanes accommodate lane-use arrow j at leg i . All traffic following lane-use arrow j on these $\sum_{l \in \mathcal{Q}_i^v} \Delta_{i,l,j}$ arrival lanes departs the intersection at leg $\theta(i, j)$. Hence, leg $\theta(i, j)$ is required to have at least $\sum_{l \in \mathcal{Q}_i^v} \Delta_{i,l,j}$ departure lanes:

$$\mathbf{e}_{\theta(i,j)} \geq \sum_{l \in \mathcal{Q}_i^v} \Delta_{i,l,j}, \quad i \in \mathcal{L}, \quad j \in \mathcal{A}_i. \quad (6.5)$$

Remark 6.7. Note that the above lower bounds on \mathbf{e}_i are integral-valued. Hence, whenever minimizing the number of lanes at the intersection, the resulting number of departure lanes \mathbf{e}_i is always integral. Assume that \mathbf{e}_i is not integral. We can then always round this variable (down) without destroying feasibility of the solution. Alternatively, for any solution we can calculate the minimum number of departure lanes required at each of the legs $i \in \mathcal{L}$ and set \mathbf{e}_i , $i \in \mathcal{L}$ to these corresponding values; this also results in a feasible solution.

Maximum number of lanes The number of lanes used at leg $i \in \mathcal{L}$ is bounded from above:

$$\mathbf{a}_i + \mathbf{e}_i \leq \bar{L}_i. \quad (6.6)$$

Distribution of traffic amongst the different lanes

All traffic arriving at the intersection has to be distributed amongst the different arrival lanes at the intersection. The arrival rate for lane-use arrow j at leg i equals $\lambda_{i,j}$. This arrival rate is scaled with a growth factor β and has to be distributed amongst the different arrival lanes of leg i :

$$\sum_{l \in \mathcal{Q}_i^v} \lambda_{i,l,j} = \beta \lambda_{i,j}, \quad i \in \mathcal{L}, \quad j \in \mathcal{A}_i. \quad (6.7)$$

Furthermore, traffic is only allowed to follow a lane-use arrow if the lane is equipped with this lane-use arrow:

$$0 \leq \lambda_{i,l,j} \leq L \Delta_{i,l,j}, \quad i \in \mathcal{L}, \quad l \in \mathcal{Q}_i^v, \quad j \in \mathcal{A}_i. \quad (6.8)$$

Number of realizations

The next few constraints of the optimization problem all concern the optimization of the signal group diagram. The constraints (6.9) –(6.24) correspond to constraints of MILP problem (4.8). Therefore, for a more detailed explanation of these constraints we refer to Chapter 4.

The binary-valued design variable $\mathbf{b}_{m,k}$ indicates whether movement m has a k th realization or not. These binary variables are related as follows:

$$\mathbf{b}_{m,k+1} \leq \mathbf{b}_{m,k}, \quad m \in \mathcal{M}, \quad k \in \mathcal{K}_m^d \setminus \{\bar{K}_m\}. \quad (6.9)$$

For each realization $k \in \mathcal{K}_m^d$ of movement $m \in \mathcal{M}$ the optimization decides whether movement m has such a k th realization. Whenever movement m has no k th realization ($\mathbf{b}_{m,k} = 0$), we force the k th realization and the preceding effective red interval to have a duration of zero seconds:

$$\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) \geq 0, \quad m \in \mathcal{M}, \quad k \in \mathcal{K}_m^d, \quad (6.10)$$

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) \geq 0, \quad m \in \mathcal{M}, \quad k \in \mathcal{K}_m^d, \quad (6.11)$$

$$\gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k) \leq L \mathbf{b}_{i,k}, \quad m \in \mathcal{M}, \quad k \in \mathcal{K}_m^d, \quad (6.12)$$

where $L = 1$ is sufficiently large. As a consequence of these constraints, movement m effectively has no k th realization whenever $\mathbf{b}_{m,k} = 0$.

Minimum and maximum effective green and effective red times

We have a lower and an upper bound on each effective green time and each effective red time. Each effective green time of movement m has an upper bound of \bar{g}_m seconds:

$$\gamma(\textcircled{m}_k, \textcircled{m}_k) \leq \bar{g}_m \mathbf{T}', \quad m \in \mathcal{M}, \quad k \in \bar{\mathcal{K}}_m. \quad (6.13)$$

Similarly, each effective red time of movement m is bounded from above:

$$\gamma(\textcircled{m}_{k-1}, \textcircled{m}_k) \leq \bar{r}_m \mathbf{T}', \quad m \in \mathcal{M}, \quad k \in \bar{\mathcal{K}}_m. \quad (6.14)$$

An effective green interval is bounded from below as well. Movement $m \in \mathcal{M}$ is guaranteed to have all realizations $k \in \underline{\mathcal{K}}_m$. For these realizations we formulate this lower bound as follows:

$$\underline{g}_m \mathbf{T}' \leq \gamma(\textcircled{m}_k, \textcircled{m}_k), \quad m \in \mathcal{M}, \quad k \in \underline{\mathcal{K}}_m. \quad (6.15)$$

For each realization $k \in \mathcal{K}_m^d$ the optimization decides whether movement m has such a k th realization; the k th effective green time becomes zero whenever no k th realization exists ($\mathbf{b}_{m,k} = 0$). Hence, this lower bound of \underline{g}_m seconds must become redundant in case $\mathbf{b}_{m,k} = 0$:

$$\underline{g}_m \mathbf{T}' - (1 - \mathbf{b}_{m,k})L \leq \gamma(\textcircled{m}_k, \textcircled{m}_k), \quad m \in \mathcal{M}, \quad k \in \mathcal{K}_m^d, \quad (6.16)$$

where in this case $L = \underline{g}_m / \underline{T}$ is sufficiently large. Similarly, we have a lower bound on each effective red time:

$$\underline{r}_m \mathbf{T}' \leq \gamma(\textcircled{m}_{k-1}, \textcircled{m}_k), \quad m \in \mathcal{M}, \quad k \in \underline{\mathcal{K}}_m, \quad (6.17)$$

$$\underline{r}_m \mathbf{T}' - (1 - \mathbf{b}_{m,k})L \leq \gamma(\textcircled{m}_{k-1}, \textcircled{m}_k), \quad m \in \mathcal{M}, \quad k \in \mathcal{K}_m^d, \quad (6.18)$$

where in this case $L = \underline{r}_m / \underline{T}$ is sufficiently large. The latter constraint allows the effective red interval preceding the k th realization of movement m to have a duration of zero seconds whenever no k th realization exists ($\mathbf{b}_{m,k} = 0$).

Asymmetry

We reduce the symmetry of the MILP problem by forcing the first effective red interval of each movement to be the largest:

$$\gamma(\textcircled{m}_{\bar{K}_i}, \textcircled{m}_1) \geq \gamma(\textcircled{m}_{k-1}, \textcircled{m}_k), \quad m \in \mathcal{M}, \quad k \in \bar{\mathcal{K}}_m \setminus \{1\}. \quad (6.19)$$

Reducing the symmetry of the MILP problem is expected to decrease the computation time needed to solve the MILP problem.

Clearance times

Minimum clearance times have to be satisfied for each pair of conflicting movements $\{m, m'\} \in \Psi_{\mathcal{M}}$. Before switching movement m' to effective green, movement m has to be effective red for at least $\underline{c}_{m,m'}$ seconds:

$$\underline{c}_{m,m'} \leq \gamma(\textcircled{m}_k, \textcircled{m'}_{k'}), \quad \{(m, k), (m', k')\} \in \Psi_R, \quad (6.20)$$

where Ψ_R is the set of conflicting realizations; its definition can be found in Section 6.2. Whenever movement $m \in \mathcal{M}$ does not have a k th realization ($\mathbf{b}_{m,k} = 0$), the following events are forced to occur at the same time: \textcircled{m}_{k-1} , \textcircled{m}_k and \textcircled{m}_k . As a consequence, the clearance times to (from) the k th realization of movement $m \in \mathcal{M}$ are then related to the clearance times to (from) the $k - 1$ st realization of this movement. This relation is expressed in the following constraints. For each pair of conflicting realizations $\{(m, k), (m', k')\} \in \Psi_R$, $k \in \mathcal{K}_i^d$ we have the following constraint:

$$-\mathbf{b}_{m,k}L \leq \gamma(\textcircled{m}_k, \textcircled{m'}_{k'}) - \gamma(\textcircled{m}_{k-1}, \textcircled{m'}_{k'}) \leq \mathbf{b}_{m,k}L, \quad (6.21)$$

Furthermore, for each pair of conflicting realizations $\{(m, k), (m', k')\} \in \Psi_R$, $k \in \mathcal{K}_i^d$ we have the following constraint:

$$-\mathbf{b}_{m,k}L \leq \gamma(\textcircled{m}_{k'}, \textcircled{m}_k) - (\gamma(\textcircled{m}_{k'}, \textcircled{m}_{k-1}) + \gamma(\textcircled{m}_{k-1}, \textcircled{m}_{k-1})) \leq \mathbf{b}_{m,k}L. \quad (6.22)$$

Well-posedness

We allow a minimum clearance time $\underline{c}_{m,m'}$ to be negative, see Appendix C.1 for a motivation for such negative clearance times. Whenever the minimum clearance time $\underline{c}_{m,m'}$ is negative, movement m' may become effective green at most $\text{abs}(\underline{c}_{m,m'})$ seconds before movement m becomes effective red. To have a well-posed optimization problem we do however restrict the duration of a negative clearance time from movement m to movement m' . For each pair of conflicting realizations $\{(m, k), (m', k')\} \in \Psi_R$ with $k \in \underline{\mathcal{K}}_m$ we have the following constraint:

$$\gamma(\textcircled{m}_k, \textcircled{m}_k) + \gamma(\textcircled{m}_k, \textcircled{m'}_{k'}) \geq \epsilon \mathbf{T}', \quad (6.23)$$

where ϵ is a small positive number. This constraint restricts the realization $k \in \underline{\mathcal{K}}_m$ of movement m plus the clearance time to realization k' of movement m' to be at least $\epsilon > 0$ seconds. As a consequence, it can be proved (see Section 4.2.2 and Section 4.3.4) that the clearance fraction $\gamma(\textcircled{m}_k, \textcircled{m'}_{k'})$ is defined unambiguously (and represents the correct interval in time). This, for example, ensures that the lower bound on this clearance time (6.20) is correctly modeled. This lower bound does not need to be satisfied whenever movement m has no k th realization ($\mathbf{b}_{m,k} = 0$). Therefore, for each pair of conflicting realizations $\{(m, k), (m', k')\} \in \Psi_R$ with $k \in \mathcal{K}_m^d$ we have the following constraint:

$$\gamma(\textcircled{m}_k, \textcircled{m}_k) + \gamma(\textcircled{m}_k, \textcircled{m'}_{k'}) \geq \epsilon \mathbf{T}' - (1 - \mathbf{b}_{i,k})L. \quad (6.24)$$

A value of $1 + \epsilon/\underline{T}$ for L is sufficiently large. By making the above constraint redundant in case $\mathbf{b}_{m,k} = 0$, we ensure that no unwanted constraints obstruct the k th realization of movement m from having a duration of zero seconds in case that $\mathbf{b}_{m,k} = 0$ (this duration of zero seconds is forced by constraints (6.10)–(6.12)).

Modeling periodicity of the signal group diagram

The periodicity of the signal group diagram is forced with the so called cycle periodicity constraints:

$$\sum_{(i,j) \in \mathcal{C}^+} \gamma(i,j) - \sum_{(i,j) \in \mathcal{C}^-} \gamma(i,j) = z_{\mathcal{C}}, \quad \mathcal{C} \in \mathcal{B}, \quad (6.25)$$

where \mathcal{B} is an integral cycle basis of the constraint graph $G = (V, A)$; this integral cycle basis can be obtained with the method proposed in Chapter 4. For some cycles in constraint graph G the integer $z_{\mathcal{C}}$ must equal some specified value. The effective green intervals and effective red intervals of a movement together span exactly one period and, therefore, we have the following circuital constraint for each movement $m \in \mathcal{M}$:

$$\sum_{k \in \bar{\mathcal{K}}_m} \gamma(\textcircled{m}_{k-1}, \textcircled{m}_k) + \gamma(\textcircled{m}_k, \textcircled{m}_k) = 1. \quad (6.26)$$

Furthermore, we have the following circuital constraint for each pair of conflicting green intervals $\{(m, k), (m', k')\} \in \Psi_R$:

$$\gamma(\textcircled{m}_k, \textcircled{m}_k) + \gamma(\textcircled{m}_k, \textcircled{m'}_{k'}) + \gamma(\textcircled{m'}_{k'}, \textcircled{m'}_{k'}) + \gamma(\textcircled{m'}_{k'}, \textcircled{m}_k) = 1, \quad (6.27)$$

which implies that each period consists of the k th realization of movement m ($\gamma(\textcircled{m}_k, \textcircled{m}_k)T$), a clearance time from the k th realization of movement m to the k' th realization of movement m' ($\gamma(\textcircled{m}_k, \textcircled{m'}_{k'})T$), the k' th realization of movement m' ($\gamma(\textcircled{m'}_{k'}, \textcircled{m'}_{k'})T$), and the clearance time back ($\gamma(\textcircled{m'}_{k'}, \textcircled{m}_k)T$).

Lane with multiple lane-use arrows

Define two vehicular movements m and m' to 'share a lane' whenever some lane accommodates both movements. Two vehicular movements m and m' that share a lane must have identical signal timings. However, the following lemma states that it is not necessary to force each two lane-sharing movements m and m' to have equal signal timings separately; it suffices to force this equal signal timing only for the movements m and m' satisfying $j(m') = j(m) + 1$; all other movements m and m' then automatically have equal signal timings whenever they share a lane.

Lemma 6.1. *Assume that the signal timings of two lane-sharing movements m and m' are forced to be equal only if $j(m') = j(m) + 1$. Then automatically the signal timings of each two lane-sharing movements are equal.*

Proof. Consider two movements m and m' that share a lane l at leg $i(m)$. We prove that these two movements have equal signal timings. Without loss of generality assume that $j(m') > j(m)$. It is trivial that the signal timings of movements m and m' are equal if $j(m') = j(m) + 1$. Therefore, we assume $j(m') > j(m) + 1$. We can prove that lane l of leg $i(m)$ is then also equipped with lane-use arrows $j(m) + 1, j(m) + 2, \dots, j(m') - 1$. Assume the contrary: lane l of leg $i(m)$ is not equipped with some lane-use arrow $j = j(m) + 1, j(m) + 2, \dots, j(m') - 1$. From the ordering of the lane-use arrows (6.3) it follows that lane-use arrow j is on none of the lanes. As a consequence, (6.7)–(6.8) imply that no traffic arrives at lane i for lane-use arrow j , i.e., $\lambda_{i,j} = 0$. However, this contradicts $\lambda_{i,j} > 0$, which we assume for all lane-use arrows $j \in \mathcal{A}_i$ and all legs $i \in \mathcal{L}$.

Thus, lane l of leg $i(m)$ is equipped with lane-use arrows $j(m), j(m) + 1, \dots, j(m')$. Therefore, the movement that follows lane-use arrow $j(m)$ (movement m) has equal signal timings as the movement following lane-use arrow $j(m) + 1$. In turn, the movement following lane-use arrow $j(m) + 1$ has equal signal timings as the movement following lane-use arrow $j(m) + 2$ et cetera. As a result, all movements (including movement m and movement m') accommodated by lane l of leg $i(m)$ have equal signal timings, which proves the lemma. \square

Consider two movements m and m' satisfying $j(m') = j(m) + 1$ and arriving at the same lane ($i(m) = i(m')$). For each such two movements m and m' we force these movements to have equal signal timings whenever they share a lane; from the above lemma it then follows that each two lane-sharing movements have equal signal timings. We force this equal signal timing as follows. We introduce an auxiliary variable $s_{m,m'}$ that indicates whether movement m and movement m' share a lane. We force this real-valued design variable to equal one whenever movements m and m' share a lane; otherwise this variable is free to take any value in the range $[0, 1]$. We do so with the following constraints:

$$s_{m,m'} \geq \Delta_{i,l,j} + \Delta_{i,l,j+1} - 1, \quad l \in \mathcal{Q}_i^v, \quad (6.28)$$

$$s_{m,m'} \leq 1, \quad (6.29)$$

$$s_{m,m'} \geq 0, \quad (6.30)$$

We force the signal timings of movement m and movement m' to be the same when $s_{m,m'}$ equals one. Two movements have equal signal timings when:

- 1 these movements have the same number of realizations,
- 2 effective green time k of movement m equals effective green time k of movement m' ,
- 3 effective red time k of movement m equals effective red time k of movement m' , and
- 4 the first realization starts at the same time for both movements.

In case that movement m and movement m' share a lane, we force their number of realizations to be the same. Let K_m denote the number of realizations of movement m ,

we have:

$$K_m := \underline{K}_m + \sum_{k \in \mathcal{K}_m^d} \mathbf{b}_{m,k}.$$

We formulate the constraint as follows:

$$-L_1(1 - \mathbf{s}_{m,m'}) \leq K_m - K_{m'} \leq L_2(1 - \mathbf{s}_{m,m'}), \quad (6.31)$$

where L_1 and L_2 are two large numbers; in this case $L_1 := \max\{\bar{K}_{m'} - \underline{K}_m, 0\}$ and $L_2 := \max\{\bar{K}_m - \underline{K}_{m'}, 0\}$ are sufficiently large. In case that movement m and movement m' do not share a lane, then $\mathbf{s}_{m,m'}$ is free to attain any value in the range $[0, 1]$ and, as a result, we do not restrict the difference $K_m - K_{m'}$. However, if these movements do share a lane, then $\mathbf{s}_{m,m'}$ equals one, which forces the difference $K_m - K_{m'}$ to equal zero. Note that substituting the definitions of K_m and $K_{m'}$ into (6.31) results in a linear constraint. Note that, as a consequence of constraint (6.31), movements m and m' cannot have any realization $k \notin \bar{\mathcal{K}}_m \cap \bar{\mathcal{K}}_{m'}$ whenever they share a lane. Whenever movement m and movement m' share a lane, we force each of their effective green times and each of their effective red times to be the same for both movements. Thus, for each realization $k \in \bar{\mathcal{K}}_m \cap \bar{\mathcal{K}}_{m'}$ we have the following constraints:

$$-(1 - \mathbf{s}_{m,m'}) \leq \gamma(\textcircled{m}_k, \textcircled{m}_k) - \gamma(\textcircled{m}'_k, \textcircled{m}'_k) \leq (1 - \mathbf{s}_{m,m'}), \quad (6.32)$$

$$-(1 - \mathbf{s}_{m,m'}) \leq \gamma(\textcircled{m}_{k-1}, \textcircled{m}_k) - \gamma(\textcircled{m}'_{k-1}, \textcircled{m}'_k) \leq (1 - \mathbf{s}_{m,m'}). \quad (6.33)$$

Note that these constraints do not restrict the durations of these effective green and effective red times whenever movements m and m' do not share a lane. Furthermore, note that when the movements do not have a realization $k \in \bar{\mathcal{K}}_m \cap \bar{\mathcal{K}}_{m'}$, the k th effective green time and the k th effective red time of movements m and m' are forced to equal zero by (6.10)–(6.12); constraints (6.32)–(6.33) are then automatically satisfied for these effective green and effective red intervals.

For two movements to have equal signal timings we also require their first effective green interval to start at the same time. In other words, we require the events \textcircled{m}_1 and \textcircled{m}'_1 to occur simultaneously. Consider a path \mathcal{P} in constraint graph $G = (V, A)$ from vertex \textcircled{m}_1 to vertex \textcircled{m}'_1 . Let \mathcal{P}^+ be the set of arcs that this path traverses in the forward direction (from tail to head) and \mathcal{P}^- be the set of arcs that this path traverses in the backward direction (from head to tail). The length of this path $\ell(\mathcal{P}) := \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^-} \gamma(\varepsilon_1, \varepsilon_2)$ denotes the time (expressed as a fraction of the period duration) between an occurrence of event \textcircled{m}_1 (the start of the first effective green interval of movement m) and an occurrence of event \textcircled{m}'_1 (the start of the first effective green interval of movement m'). Therefore, the two events \textcircled{m}_1 and \textcircled{m}'_1 occur simultaneously if and only if the length $\ell(\mathcal{P})$ of this path is integral valued. Let $\underline{\ell}(\mathcal{P})$ and $\bar{\ell}(\mathcal{P})$ be a lower bound respectively an upper bound on the value of $\ell(\mathcal{P})$. With the following constraints we can then force the events \textcircled{m}_1 and \textcircled{m}'_1 to occur simultaneously when movement m and movement m' share a lane:

$$\underline{\ell}(\mathcal{P})(1 - \mathbf{s}_{m,m'}) \leq \ell(\mathcal{P}) - \mathbf{z}_{\mathcal{P}} \leq \bar{\ell}(\mathcal{P})(1 - \mathbf{s}_{m,m'}), \quad l \in \mathcal{Q}_i^v, \quad (6.34)$$

$$\lceil \underline{\ell}(\mathcal{P}) \rceil \mathbf{s}_{m,m'} \leq \mathbf{z}_{\mathcal{P}} \leq \lfloor \bar{\ell}(\mathcal{P}) \rfloor \mathbf{s}_{m,m'}, \quad l \in \mathcal{Q}_i^v, \quad (6.35)$$

where $\mathbf{z}_{\mathcal{P}}$ is an integral-valued design variable. Whenever $\mathbf{s}_{m,m'} = 1$, the length $\ell(\mathcal{P})$ of path \mathcal{P} is forced to equal some integral value $\mathbf{z}_{\mathcal{P}}$; from $\ell(\mathcal{P}) \in [\underline{\ell}(\mathcal{P}), \bar{\ell}(\mathcal{P})]$ it follows that this integral value satisfies $\mathbf{z}_{\mathcal{P}} \in \{\lceil \underline{\ell}(\mathcal{P}) \rceil, \lceil \underline{\ell}(\mathcal{P}) \rceil + 1, \dots, \lfloor \bar{\ell}(\mathcal{P}) \rfloor\}$. When $\mathbf{s}_{m,m'} = 0$, we do not restrict the length $\ell(\mathcal{P})$ of path \mathcal{P} and force the integral variable $\mathbf{z}_{\mathcal{P}}$ to equal zero.

We can find a path \mathcal{P} from vertex \textcircled{m}_1 to vertex $\textcircled{m'}_1$ as follows. Consider the following directed conflict graph $G' = (V', A')$:

$$\begin{aligned} V' &:= \{\textcircled{m}_1 \mid m \in \mathcal{M}\}, \\ A' &:= \{(\textcircled{m}_1, \textcircled{m'}_1) \mid \{m, m'\} \in \Psi_{\mathcal{M}}, m < m'\}. \end{aligned}$$

This graph has one vertex for each movement $m \in \mathcal{M}$ and it has one arc for each pair of conflicting movements $\{m, m'\} \in \Psi_{\mathcal{M}}$. Let \mathcal{P}' be the shortest path (in number of arcs) in graph G' from vertex \textcircled{m}_1 to vertex $\textcircled{m'}_1$; we can easily find such a path by applying any shortest path algorithm. Let the path \mathcal{P}' traverse the sequence of vertices $\textcircled{m}_1, \textcircled{m_1}, \dots, \textcircled{m_n}, \textcircled{m'}_1$, where $m_1 := m$ and $m_n := m'$. From this path we can obtain the following path \mathcal{P} in the (much larger) constraint graph G :

$$\textcircled{m}_1, \textcircled{m_1}, \textcircled{m_2}, \textcircled{m_3}, \dots, \textcircled{m_n}, \textcircled{m'}_1. \quad (6.36)$$

which, as desired, is a path from vertex \textcircled{m}_1 to vertex $\textcircled{m'}_1$.

We also require a lower bound $\underline{\ell}(\mathcal{P})$ and an upper bound $\bar{\ell}(\mathcal{P})$ on the length $\ell(\mathcal{P})$. We can prove the inclusion $\ell(\mathcal{P}) \in (0, n)$ and, as a consequence, $\underline{\ell}(\mathcal{P}) = 0$ and $\bar{\ell}(\mathcal{P}) = n$. Define $\gamma(\textcircled{m}_1, \textcircled{m'}_1)' := \gamma(\textcircled{m}_1, \textcircled{m_1}) + \gamma(\textcircled{m_1}, \textcircled{m'}_1)$. It suffices to prove $\gamma(\textcircled{m}_1, \textcircled{m'}_1)' \in (0, 1)$ for each pair of conflicting movements $\{m, m'\} \in \Psi_{\mathcal{M}}$. This inclusion follows from the well-posedness constraint (6.23) together with circuitual constraint (6.27).

Remark 6.8. Consider two movements m and m' satisfying $j(m') = j(m) + 1$ and arriving at the same leg ($i(m) = i(m')$). Usually such movements have at least one conflict in common. Such a common conflict is for example the movement coming from leg $\theta(i(m'), j(m'))$ and moving towards leg $\theta(i(m), j(m))$; in Figure 6.6 we have visualized three examples of this common conflict.

When two movements m and m' have a conflict in common, the shortest path \mathcal{P}' has a length $|\mathcal{P}'|$ of 2. Therefore, in this case $\ell(\mathcal{P}) \in (0, 2)$, where \mathcal{P} is the path in constraint graph G constructed from the path \mathcal{P}' as explained before; events \textcircled{m}_1 and $\textcircled{m'}_1$ then occur simultaneously if and only if $\ell(\mathcal{P}) = 1$. Therefore, constraints (6.34)–(6.35) can then be replaced by:

$$-(1 - \mathbf{s}_{m,m'}) \leq \ell(\mathcal{P}) - 1 \leq (1 - \mathbf{s}_{m,m'}),$$

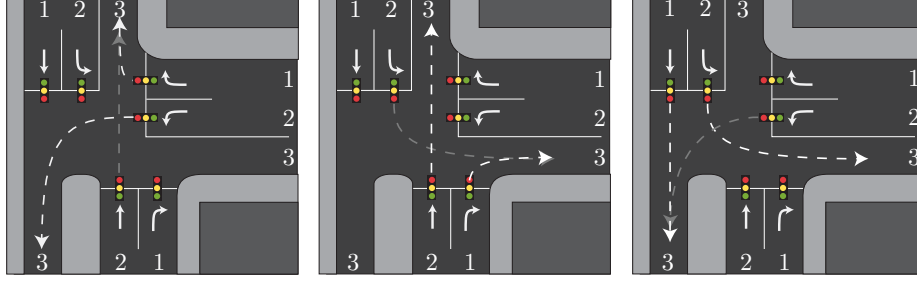


Figure 6.6: Examples of the common conflict (grey) coming from leg $\theta(i(m'), j(m'))$ and moving towards leg $\theta(i(m), j(m))$ for two movements m and m' (white).

which forces $\ell(\mathcal{P})$ to equal 1 when $s_{m,m'} = 1$ and it does not restrict $\ell(\mathcal{P})$ when $s_{m,m'} = 0$. This implies that we do not require the additional integral variable $z_{\mathcal{P}}$ when movements m and m' have a conflict in common.

Remark 6.9. Theoretically it is possible that no path exists from vertex \textcircled{m}_1 to vertex $\textcircled{m'}_1$; these two vertices are then in different connected components of constraint graph G . For these movements m and m' we can then add arc $(\textcircled{m}_1, \textcircled{m'}_1)$ to constraint graph G ; the length $\gamma(\textcircled{m}_1, \textcircled{m'}_1)$ of this arc represents the time (expressed as a fraction of the period duration) between event \textcircled{m}_1 and event $\textcircled{m'}_1$. We may assume w.l.o.g. $\gamma(\textcircled{m}_1, \textcircled{m'}_1) \in [0, 1)$. Therefore, we can force the events \textcircled{m}_1 and $\textcircled{m'}_1$ to occur simultaneously by replacing constraints (6.34)–(6.35) with:

$$0 \leq \gamma(\textcircled{m}_1, \textcircled{m'}_1) \leq s_{m,m'}.$$

Note that adding this arc to constraint graph G does not increase the cyclomatic number $d := |A| - |V| + \nu(G)$ of this graph; the number of arcs $|A|$ increases by one and the number of connected components $\nu(G)$ decreases by one.

Stability of the signal group diagram

We need to ensure that each pedestrian crossing and each cyclist crossing can handle the amount of traffic arriving at it. Therefore, for each movement $m \in \mathcal{M}^{\text{pc}}$ we require its effective green intervals to span at least a fraction $\beta\rho_m$ of the period duration, where $\rho_m := \max_{q \in \mathcal{Q}_m^{\text{pc}}} \lambda_q / \mu_q$:

$$\sum_{k \in \bar{\mathcal{K}}_m} \gamma(\textcircled{m}_k, \textcircled{m}_k) \geq \beta\rho_m, \quad m \in \mathcal{M}^{\text{pc}}.$$

This ensures that for each of the queues $q \in \mathcal{Q}_m^{\text{pc}}$ the average amount of traffic arriving at this queue is sustainable.

Furthermore, we ensure that each arrival lane is able to handle the amount of traffic arriving at it. Consider a lane l of leg i . The (average) rate at which traffic following lane-use arrow $j \in \mathcal{A}_i$ arrives at lane l equals $\lambda_{i,l,j}$. Lane l requires an effective green fraction of at least $\lambda_{i,l,j} / \mu_{i,l,j}$ to handle (only) the traffic that arrives for lane-use arrow

j . Therefore, to handle all arriving movements, lane l needs to be effective green for at least a fraction $\sum_{j \in \mathcal{A}_i} \lambda_{i,l,j} / \mu_{i,l,j}$. With the following constraint we ensure that the effective green fraction of each arrival lane is large enough:

$$\sum_{k \in \bar{\mathcal{K}}_m} \gamma(\textcircled{m}_k, \textcircled{m}_k) \geq \sum_{j \in \mathcal{A}_i} \frac{\lambda_{i(m),l,j}}{\mu_{i(m),l,j}} - L(1 - \Delta_{i(m),l,j(m)}), \quad m \in \mathcal{M}^v \quad l \in \mathcal{Q}_{i(m)}^v.$$

Note that the left-hand side of this constraint is the effective green fraction of movement m ; when $\Delta_{i(m),l,j(m)} = 1$, this is also the effective green fraction of lane l of leg $i(m)$. Whenever lane l of leg $i(m)$ accommodates movement m ($\Delta_{i(m),l,j(m)} = 1$), this left-hand side is forced to exceed the fraction $\sum_{j \in \mathcal{A}_i} \lambda_{i(m),l,j} / \mu_{i(m),l,j}$ that is required for lane l of leg $i(m)$ to be stable. Since, each arrival lane accommodates at least one movement, the above constraints guarantee stability for each of these arrival lanes.

Remark 6.10. *The proposed formulation automatically distributes the traffic arriving for the vehicular movements amongst the different arrival lanes such that the optimized layout (plus the optimized signal group diagram) can handle all growth factors $\beta < \beta^{\max}$; the growth factor β^{\max} is either fixed or maximized during optimization. A question however is: can the optimized layout of the intersection (plus the optimized signal group diagram) indeed handle all growth factors $\beta < \beta^{\max}$; or do the vehicles distribute themselves amongst the different arrival lanes in a different manner for which some growth factor $\beta < \beta^{\max}$ is not sustainable. In Appendix F.1 we motivate that this is not the case: all growth factors $\beta < \beta^{\max}$ are indeed sustainable.*

Remark 6.11. *Busses often have to meet tight time schedules. Large and uncertain waiting times at signalized intersections can make it impossible to meet these tight schedules. To reduce this waiting time and uncertainty, a signalized intersection can be equipped with exclusive bus lanes. In Appendix F.2 we elaborate on how to include these exclusive bus lanes in the optimization.*

Remark 6.12. *Consider again the T-junction for which we have visualized three layout variants in Figure 6.1. Another interesting layout variant of this T-junction is the one in Figure 6.7. For this layout variant, leg 3 has two departure lanes. The right-turn movement of leg 1 (movement 1) is guided towards the right most departure lane of leg 3, while the through movement of leg 2 (movement 5) is guided towards the left most departure lane of leg 3. As a consequence, for this layout variant movements 1 and 5 are not conflicting; it is safe for the right-turn movement of leg 1 to receive a green indication simultaneous with the through movement of leg 3. Note that movement 1 may then always receive a green indication as this movement has no conflicts. As a consequence, we can omit its traffic light from the layout of the intersection. The optimization problem proposed in this chapter considers the set of conflicting movements $\Psi_{\mathcal{M}}$ to be given (and independent of the layout of the intersection). Therefore, with the proposed optimization problem it*

is not possible to simultaneously optimize over the three layout variants in Figure 6.1 and the layout variant in Figure 6.7. However, in Appendix F.3 we show that with some modifications it is also possible to include layout variants like the one in Figure 6.7 in the optimization. In the remaining part of this chapter we do however not consider the optimization over such layout variants.

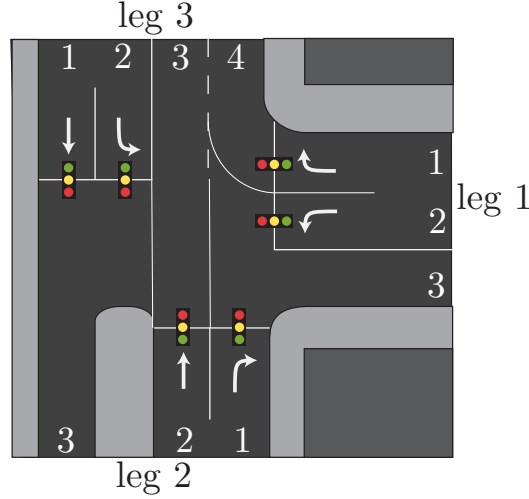


Figure 6.7: A layout variant of a T-junction. For this layout variant the right-turn movement of leg 1 may be green simultaneously with the through movement of leg 2; leg 3 has sufficient lanes to allow these traffic movements simultaneously in a safe manner.

6.4 Numerical results

In this section we test the proposed MILP formulation. To this end, we use the ten intersections that we have given in (Fleuren and Lefeber, 2016e). These examples correspond to the intersections S1, S2, S3, S4, S5, M1, M2, M3, L1 and L2 from (Fleuren and Lefeber, 2016a). The examples M4, L3 and L4 from (Fleuren and Lefeber, 2016a) are not considered for the following reasons. Example M4 is a layout variant of example M3 and, therefore, optimizing the layout of example M4 is the same as optimizing the layout of example M3. The examples L3 and L4 concern a network of intersections instead of a single intersection.

We consider two different objective functions: maximization of the capacity of the intersection and minimizing the number of lanes at the intersection. For the former problem we maximize the growth factor β , i.e., the objective function is:

$$\text{maximize } \beta.$$

For the latter problem we fix the growth factor β to one and consider the following objective function:

$$\text{minimize } \sum_{i \in \mathcal{L}} a_i + e_i.$$

Objective	$\max \beta$			$\min \sum_{i \in \mathcal{L}} \mathbf{a}_i + \mathbf{e}_i$		
Ad. real	0	2	4	0	2	4
S1	1.54	1.54	1.54	13	13	13
S2	1.70	1.70	1.70	13	13	13
S3	2.15	2.15	2.15	11	11	11
S4	3.00	3.00	3.00	7	7	7
S5	2.17	2.17	2.17	7	7	7
M1	1.93	1.93	1.93	8	8	8
M2	1.72	1.72	1.72	8	8	8
M3	1.84	1.84	1.84	10	10	10
L1	1.49	1.49	1.49	17	17	17
L2	1.49	1.49	1.49	17	17	17

Table 6.2: The objective values for the test cases in the numerical study. We have distinguished between two objective functions, three types of intersections (small, medium and large), and we have varied the number of movements that is allowed to have an additional realization, see the second row of this table. If this number of movements equals $k = 0, 2, 4$, the k movements with the largest loads are allowed to have an additional realization; for each of these movements, the optimization decides whether this movement should have one or two realizations.

Moreover, we vary the number of realizations that each signal group is allowed to have. We fix the minimum number of realizations \underline{K}_m , $i \in \mathcal{S}$ to one, i.e., each signal group must have at least one realization. We do however vary the maximum number of realizations \overline{K}_m , $m \in \mathcal{M}$. To this end, we define the load of each movement $m \in \mathcal{M}$. For each movement $m \in \mathcal{M}^{\text{pc}}$ we have $\rho_m := \max_{q \in \mathcal{Q}_m^{\text{pc}}} \lambda_q / \mu_q$. For each vehicular movement we have $\rho_m := \frac{\lambda_{i(m),j(m)}}{\mu_{i(m),j(m),1}}$, which is the load of a lane that accommodates (only) all traffic of movement m ; the saturation flow rate $\mu_{i(m),j(m),l}$ is independent of the lane l for the examples that we consider. We consider three variants for the maximum number of realizations \overline{K}_m , $m \in \mathcal{M}$: we allow the zero, two or four most heavily loaded movements to have an additional realization. Thus we have a total of $10 \times 2 \times 3 = 60$ different test cases. In Table 6.2 we give the objective values of the $10 \times 2 \times 3 = 60$ different test cases. The returned objective value does not depend on the number of movements allowed to have an additional realization. In Table 6.3 we give the (geometric) average computation times. The solver CPLEX requires the least time to solve the optimization problems, while the solver SCIP results in the largest computation times. These computation times increase drastically in the number of movements that is allowed to have an additional realization.

6.5 Summary

In this chapter we have extended the optimization framework proposed in Chapter 4 to also optimize the layout of the intersection, i.e., in this chapter we also optimize the number of lanes that each leg of the intersection has, which of these lanes is an arrival lane, which lane is a departure lane, and which lane-use arrows are marked on each of

Objective		$\max \beta$			$\min \sum_{i \in \mathcal{L}} \mathbf{a}_i + \mathbf{e}_i$		
Ad. real.		0	2	4	0	2	4
CPLEX	Small	0.18	0.29	0.48	0.14	0.14	0.18
	Medium	0.23	0.72	1.47	0.19	0.18	0.26
	Large	6.98	37.99	82.12	2.07	8.51	19.89
GUROBI	Small	0.07	0.14	0.69	0.02	0.05	0.14
	Medium	0.17	0.46	2.60	0.07	0.16	0.25
	Large	18.79	97.07	311.74	6.03	24.97	34.02
SCIP	Small	0.59	1.32	3.19	0.13	0.21	0.54
	Medium	1.54	5.64	21.54	0.40	0.76	1.61
	Large	139.51	519.84	3379.20	15.04	63.45	147.26

Table 6.3: The (geometric) average computation times (in seconds) for the test cases in the numerical study. We have distinguished between two objective functions, three types of intersections (small, medium and large), three types of solvers (CPLEX 12.6.1.0, GUROBI 6.0.5. and SCIP 3.2.0), and we have varied the number of movements that is allowed to have an additional realization (Ad. real.). If 'Ad. real.' equals $k = 0, 2, 4$, the k movements with the largest loads are allowed to have an additional realization; for each of these movements, the optimization decides whether this movement should have one or two realizations.

the arrival lanes. With the proposed MILP problem we can answer questions like: what should the lane-use arrows be such that the capacity of the intersection is maximized and what is the minimum number of lanes that is required for the intersection to have sufficient capacity? The layout determines which signal group diagrams are feasible and which ones not. Therefore, in order to answer the aforementioned questions, we have to simultaneously optimize the layout of the intersection and the signal group diagram that specifies when each of the traffic lights at the intersection is green, yellow and red. In this chapter we have considered two objective functions. The first one is the maximization of the capacity of the intersection. With this objective function we find the layout of the intersection that can handle the largest increase in arrival rates. The second objective function is the minimization of the number of lanes that are present at the intersection. With this objective function we can find the smallest intersection that can handle some specified growth factor.

In the next chapter we elaborate on how the proposed optimization framework can be used in practice and, moreover, we elaborate on some issues that might be relevant in practice, but are not yet touched upon in this thesis. Subsequently, in the final chapter of this thesis we give our concluding remarks and recommendations.

Usage in practice

7.1 Introduction

In this thesis we have given an optimization framework that can be used to optimize (integral) signal group diagrams; such a signal group diagram visualizes when each traffic light is green, yellow and red for a pre-timed controller. With this optimization framework it is for example possible to search for the (integral) signal group diagram that minimizes the period duration T , maximizes the capacity of the intersection (i.e., maximizing the growth factor of the arrival rates), or minimizes the average delay that road users experience. Furthermore, we are able to simultaneously optimize a signal group diagram and the layout of the intersection. This makes it possible to for example maximize the capacity of the intersection by optimizing the lane-use arrows or find the smallest intersection that has sufficient capacity. In this chapter we discuss how this optimization framework can assist the practicing traffic engineers in the design process of a traffic light controller. To this end, we discuss some practical issues that were not yet considered in this thesis.

In practice a signal group diagram is often not used directly to control the traffic lights at an intersection. However, from such a signal group diagram we can obtain a phase diagram which may form the basis of a (semi)actuated controller. Such a (semi)actuated controller uses detector information to control (some of) the traffic lights. The goal of a (semi)actuated controller is to improve the performance of a single intersection or to improve the performance of a network of intersections. The design process of a (semi)actuated controller may include the following steps:

- A signal group diagram is designed (or optimized) for each of the signalized intersections in isolation; this results in a pre-timed controller for each intersection. A signal group diagram can, for example, be obtained with the software package COCON (DTV consultants, 2015), VRIGen (Muller and de Leeuw, 2006; Salomons, 2008) or LISA+ (Scholthauer & Wauer, 2016).
- To improve the overall performance of a network of signalized intersections, the pre-timed controllers of these intersections can be coordinated. In other words, we can

optimize the relative timing between the pre-timed controllers that were obtained for each of these signalized intersections in isolation; optimizing these relative timings may create green waves and improve the overall network performance. To coordinate the different pre-timed controllers we can use, for example, the software package LISA+ (Scholthauer & Wauer, 2016) or TRANSYT (Robertson, 1969).

- Subsequently, from the pre-timed controllers a (semi)actuated controller can be designed. Usually this (semi)actuated controller is based on a phase diagram, which can be obtained from the pre-timed controller. Such a (semi)actuated controller uses detector information to control the traffic lights. As a consequence, this actuated controller adapts to the current traffic situation. An actuated controller can be designed with for example the software package RWS C-regelaar (Rijkswaterstaat WVL, 2014) or Toolkit CCOL (Peters and Prinsen, 2001).
- Finally, the (semi)actuated controller can be tested thoroughly with for example the micro-simulation of software package VISSIM (PTV AG, 2015).

This thesis contributes to the improvement of the first step of this design process. In this chapter we elaborate on topics such as: how to include additional constraints that may be desired in practice, how to obtain multiple signal groups instead of only the optimal one, and what to do whenever input parameters (for example clearance times) depend on the optimization outcome.

7.2 Additional constraints

In this section we elaborate on some additional constraints that may be relevant in practice, but that are not yet considered in this thesis.

7.2.1 Synchronous start of the green indication

Consider the intersection in Figure 7.1. The left-turn movement of leg 2 (leg 4) has a conflict with the through movement of leg 4 (leg 2). The signal groups that correspond to these movements are signal group 5 and signal group 11. A traffic engineer may permit signal groups 5 and 11 to receive a green indication simultaneously; for example when the amount of traffic making a left turn is relatively small for these signal groups. Permitting signal groups 5 and 11 to receive a green indication at the same time, reduces the number of conflicting signal groups at the intersection and increases the class of feasible signal group diagrams. This may have a positive effect on the performance of the intersection, e.g., the capacity of the intersection or the average delay that road users experience. However, these signal groups should be permitted to receive right-of-way simultaneously only if this results in a safe situation. Consider for example the (partial) signal group diagram in Figure 7.2a. The vehicles that depart at the start of the green interval of

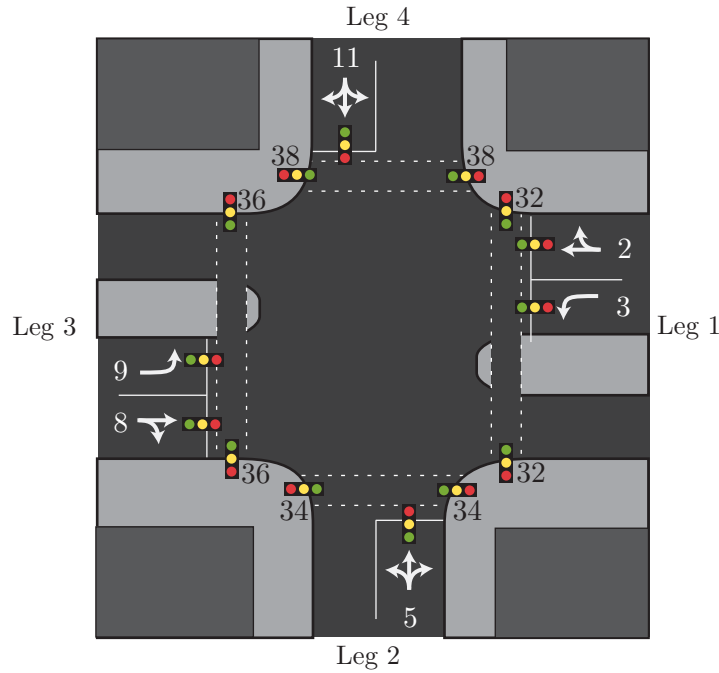
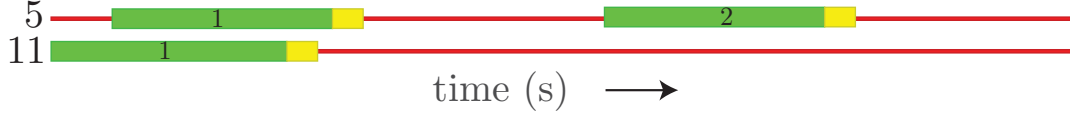


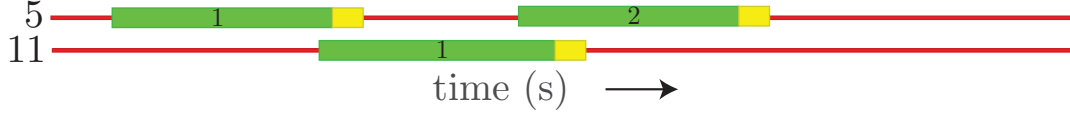
Figure 7.1: An example of an intersection for which possibly additional constraints have to be satisfied.

signal group 11 are unaware of the green interval that starts at signal group 5. As a consequence these vehicles from signal group 11 do not expect to encounter conflicting traffic from signal group 5, which may result in a dangerous situation. Similarly, the (partial) signal group diagram in Figure 7.2a may also result in an unsafe situation; for this signal group diagram, vehicles of signal group 11 that depart between the two realizations of signal group 5 do not expect to encounter conflicting traffic from signal group 5. On the contrary, the (partial) signal groups of Figure 7.2c and Figure 7.2d may result in a safe situation. For the signal group of Figure 7.2c, the first realization of signal group 5 starts simultaneously with the green interval of signal group 11. As a consequence, the left-turning vehicles of both signal group 5 and signal group 11 departing during these realizations are aware of the green indication of the opposing through movement. For the signal group of Figure 7.2d, the left-turning traffic from signal group 5 (signal group 11) does not encounter any conflicting traffic from signal group 11 (signal group 5). This motivates for each realization k of signal group 5 and each realization k' of signal group 11 to satisfy one of the following two statements:

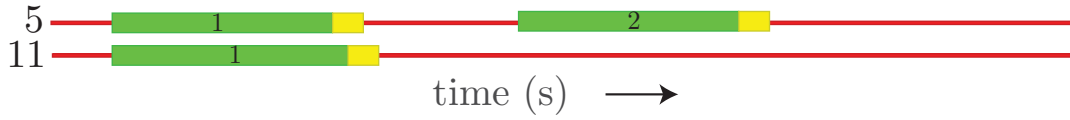
- The realization k of signal group 5 starts simultaneously with realization k' of signal group 11. The left-turning vehicles (of both signal group 5 and signal group 11) departing during these realizations are then aware of the green indication of the opposing through movement.
- The realization k of signal group 5 is considered to be conflicting with realization k' of signal group 11; clearance times must be satisfied between these green intervals and, as a consequence, the left-turning traffic that departs during the k th (k' th)



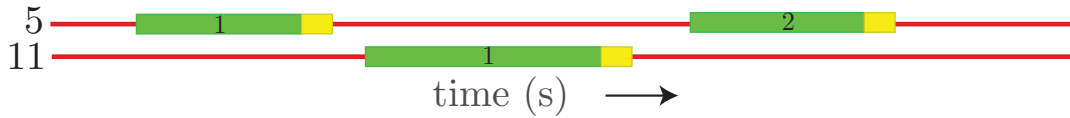
(a) Left-turning vehicles that depart at the start of the realization of signal group 11 do not expect to encounter conflicting traffic from signal group 5, which they may encounter. As a consequence this signal group may result in an unsafe situation.



(b) Left-turning vehicles of signal group 11 that depart between the realizations of signal group 5 do not expect to encounter conflicting traffic from signal group 5, which they may encounter. As a consequence this signal group may result in an unsafe situation.



(c) The first realization of signal group 5 starts simultaneously with the green interval of signal group 11; as a consequence, the left-turning vehicles of signal groups 5 and 11 are aware of the green indication of the opposing through movement. Therefore, this signal group results in a safe situation.



(d) Traffic from signal group 5 and signal group 11 making a left-turn is not expected to encounter any traffic from the opposing through movement. As a consequence, this signal group results in a safe situation.

Figure 7.2: Signal group diagrams for signal groups 5 and 11 of the intersection in Figure 7.1 that may result in a safe or an unsafe situation.

realization of signal group 5 (signal group 11) is not expected to encounter any traffic that departs during the k' th (k th) realization of signal group 11 (signal group 5).

Formulating the constraint

In this section we elaborate on how to include the aforementioned constraint in the optimization framework proposed in this thesis. Consider a realization k of signal group 5 and consider a realization k' of signal group 11; we denote these realizations by realization $(5, k)$ and realization $(11, k')$. For reasons of simplicity assume that $k \leq \underline{K}_5$ and $k' \leq \underline{K}_{11}$. We introduce a binary variable \mathbf{b} to indicate if these realizations switch to green simultaneously ($\mathbf{b} = 1$) or if these realizations are considered to be conflicting ($\mathbf{b} = 0$). Consider these realizations to be conflicting ($\mathbf{b} = 0$). We then have the usual constraints

on the clearance fractions $\gamma(\textcircled{5}_k, \textcircled{11}_{k'})$ and $\gamma(\textcircled{11}_{k'}, \textcircled{5}_k)$, see also Chapter 4. First, both realizations must be scheduled within one period, which implies

$$\gamma(\textcircled{5}_k, \textcircled{5}_k) + \gamma(\textcircled{5}_k, \textcircled{11}_{k'}) + \gamma(\textcircled{11}_{k'}, \textcircled{11}_{k'}) + \gamma(\textcircled{11}_{k'}, \textcircled{5}_k) = 1. \quad (7.1)$$

Second, we have the constraints regarding the clearance times. However, these constraints have to be satisfied only when $\mathbf{b} = 0$:

$$\begin{aligned} c_{5,11} \mathbf{T}' - \mathbf{b}L &\leq \gamma(\textcircled{5}_k, \textcircled{11}_{k'}) \\ c_{5,11} \mathbf{T}' - \mathbf{b}L &\leq \gamma(\textcircled{11}_{k'}, \textcircled{5}_k) \end{aligned}$$

where L is a large (positive) number. Third, we have well-posedness constraints, which become redundant in case $\mathbf{b} = 1$:

$$\begin{aligned} \gamma(\textcircled{5}_k, \textcircled{5}_k) + \gamma(\textcircled{5}_k, \textcircled{11}_{k'}) &\geq \epsilon - \mathbf{b}L \\ \gamma(\textcircled{11}_{k'}, \textcircled{11}_{k'}) + \gamma(\textcircled{11}_{k'}, \textcircled{5}_k) &\geq \epsilon - \mathbf{b}L \end{aligned}$$

When $\mathbf{b} = 1$, green interval $(5, k)$ should start simultaneously with green interval $(11, k')$. This can be forced with the following constraint:

$$-L_1(1 - \mathbf{b}) \leq \gamma(\textcircled{5}_k, \textcircled{5}_k) + \gamma(\textcircled{5}_k, \textcircled{11}_{k'}) + (l_5^s - l_{11}^s) \mathbf{T}' \leq L_2(1 - \mathbf{b}). \quad (7.2)$$

where L_1 and L_2 are large (positive) numbers and where l_5^s and l_{11}^s are starting lost times. This constraint forces effective green interval $(11, k')$ to start $l_{11}^s - l_5^s$ seconds after the start of effective green interval $(5, k)$. As a consequence, their corresponding green intervals start at the same time.

Thus, when $\mathbf{b} = 0$, the minimum clearance times and the well-posedness constraints should be satisfied; the realizations $(5, k)$ and $(11, k')$ are then considered to be conflicting. When $\mathbf{b} = 1$, constraint (7.2) forces the realizations $(5, k)$ and $(11, k')$ to start their green interval simultaneously.

Remark 7.1. When \mathbf{b} equals one, circuital constraint (7.1) forces the equality:

$$\gamma(\textcircled{11}_{k'}, \textcircled{11}_k) + \gamma(\textcircled{11}_{k'}, \textcircled{5}_k) + (l_{11}^s - l_5^s) = 1.$$

This equality implies that a period is comprised of effective green interval k' of signal group 11, the clearance time towards effective green interval k of signal group 5, plus the time between the start of effective green interval k of signal group 5 and the start of effective green interval k' of signal group 11 (which is forced to equal $(l_{11}^s - l_5^s)$). Note that all other constraints on $\gamma(\textcircled{11}_{k'}, \textcircled{5}_k)$ are redundant and allow this equality.

Remark 7.2. Assume that $k > \underline{K}_5$ and/or $k' > \underline{K}_{11}$. In Chapter 4 we can find all inequality constraints that have to be satisfied by the clearance fractions $\gamma(\textcircled{5}_k, \textcircled{11}_{k'})$ and $\gamma(\textcircled{11}_{k'}, \textcircled{5}_k)$ when the realizations $(5, k)$ and $(11, k')$ are considered to be conflicting ($\mathbf{b} =$

0). These constraints have to be satisfied only when $\mathbf{b} = 0$. We can make all these constraints redundant when $\mathbf{b} = 1$ by subtracting $\mathbf{b}L$ from either the left-hand side or the right-hand side of this inequality, where L is some large (positive) number. Note that it is however not necessary to make constraints (4.8s)–(4.8t) redundant; these constraints relate the clearance times when signal group 5 has no k th realization ($\mathbf{b}_{5,k} = 0$) or when signal group 11 has no k' realization ($\mathbf{b}_{11,k'} = 0$). Furthermore, we may assume w.l.o.g. that \mathbf{b} equals zero whenever $\mathbf{b}_{5,k} = 0$ or $\mathbf{b}_{11,k} = 0$; in that situation no constraints between realization $(5, k)$ and realization $(11, k')$ have to be satisfied. We can force \mathbf{b} to equal zero when $\mathbf{b}_{5,k} = 0$ or $\mathbf{b}_{11,k} = 0$ by including the constraints $\mathbf{b} \leq \mathbf{b}_{5,k}$ and $\mathbf{b} \leq \mathbf{b}_{11,k}$.

Remark 7.3. For each of the \bar{K}_{11} realizations of signal group 11 we have a binary variable that indicates whether this realization starts simultaneously with realization k of signal group 5. Since realization k of signal group 5 can switch to green simultaneously with at most one of these realizations, we may restrict the sum of these \bar{K}_{11} binary variables to be at most one.

Remark 7.4. The constraint graph G associated with this problem has the same arcs as when signal groups 5 and 11 are conflicting. Therefore, we can find an integral cycle basis of the constraint graph with the method from Chapter 4.

7.2.2 Coordination constraints

When coordinating the pre-timed controllers of different intersections, it is desirable that these pre-timed controllers have a common period duration. Such a common period duration can be found by optimizing the signal group diagrams (pre-timed controllers) of these intersections simultaneously. In that case the constraint graph G consists of multiple connected components; it has one such connected component for each intersection. It may also be desirable to assess the possibility of creating green waves between these intersections. Consider a signal group i and a signal group j (at another intersection) to which a large part of the traffic from signal group i goes. Suppose this traffic experiences a green wave whenever signal group j becomes effective green c seconds after signal group i switches to effective green. We can then coordinate the k th realization of signal group i and the k' th realization of signal group j by adding the arc $((i)_k, (j)_{k'})$ to constraint graph G and forcing the associated design variable $\gamma((i)_k, (j)_{k'})$ to equal $c\mathbf{T}'$. By adding such coordination constraints we alter the constraint graph. In the next chapter we show how to find an integral cycle basis of the resulting constraint graph.

Finding an integral cycle basis

Suppose we have added n coordination constraints. Let a_1, \dots, a_n denote the corresponding arcs added to the constraint graph G . Let G be the original constraint graph as defined in (4.6); this constraint graph does not include the arcs a_1, \dots, a_n . Let graph \tilde{G}

be the graph that is obtained by adding arcs a_1, \dots, a_n to graph G . With Algorithm 2 we can find an integral cycle basis of the constraint graph \tilde{G} from an ICB of the original constraint graph G .

Algorithm 2 Let graph \tilde{G} be the graph that is obtained by adding the arcs a_1, \dots, a_n to some graph G . With this algorithm we can find an ICB of graph \tilde{G} from an ICB of graph G .

Input: Graph G and the additional arcs a_1, \dots, a_n .

1: **Initialize:** Set \mathcal{B} to an ICB of graph G .

2: **for** $i=1, \dots, n$ **do**

3: Try to find a cycle \mathcal{C} in graph \tilde{G} that uses arc a_i and does not use arcs a_1, \dots, a_{i-1} . If such a cycle exists include it in \mathcal{B} , i.e., $\mathcal{B} := \mathcal{B} \cup \{\mathcal{C}\}$.

Lemma 7.1. *Algorithm 2 returns an integral cycle basis (ICB) of graph \tilde{G} .*

Proof. Via induction we prove that this algorithm returns an ICB of graph \tilde{G} . Let \mathcal{B}_0 be the integral cycle basis of graph G and let \mathcal{B}_l be the set of cycles \mathcal{B} that Algorithm 2 has found after iteration l . Moreover, let G_l denote the graph that is obtained by adding arcs a_1, \dots, a_l to graph G and define d_l as the cyclomatic number of the graph G_l . We assume that the set \mathcal{B}_{l-1} , $l \geq 1$ is an ICB of graph G_{l-1} (this trivially holds for $l = 1$) and prove that \mathcal{B}_l is then an ICB of graph G_l . Proving this induction step would prove this lemma as $G_n = \tilde{G}$.

Assume that we have found a cycle \mathcal{C} during iteration l . With respect to graph G_{l-1} , graph G_l then has the same number of vertices, the same number of connected components and one additional arc. Therefore, $d_l = d_{l-1} + 1$. Cycle \mathcal{C} is the only cycle in \mathcal{B}_l that uses arc a_l . Therefore, the cycle-arc incidence matrix associated with \mathcal{B}_l has some nonsingular $d_l \times d_l$ submatrix X_l of the form:

$$X_l := \begin{bmatrix} 1 & x \\ \mathbf{0} & X_{l-1} \end{bmatrix}$$

where x is some row vector and X_{l-1} is a nonsingular $d_{l-1} \times d_{l-1}$ submatrix of the cycle-arc incidence matrix of ICB \mathcal{B}_{l-1} ; the first row of the matrix X_l is indexed by cycle \mathcal{C} and the first column of this matrix is indexed by arc a_l . From this expression it follows that $\det(X_l) = \det(X_{l-1}) = \pm 1$. As a consequence, Theorem 5.2 proves that \mathcal{B}_l is an ICB of graph G_l .

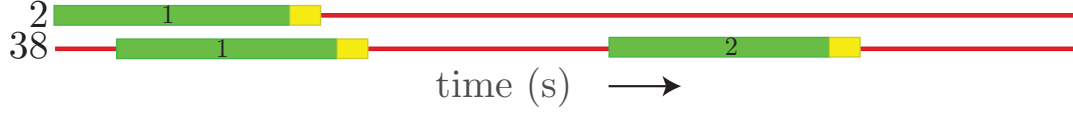
Assume that no cycle was found during iteration l . Arc a_l then connects two different connected components of graph G_{l-1} . Therefore, with respect to graph G_{l-1} , graph G_l has the same number of vertices, one less component and one additional arc. This implies $d_{l-1} = d_l$. Therefore, \mathcal{B}_l is also an integral cycle basis of the graph G_l when no cycle was found during iteration l . This concludes this proof. \square

7.2.3 Relation between the switches to green of permitted movements

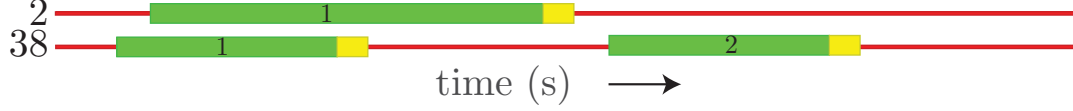
Consider signal group 2 and signal group 38 of the intersection in Figure 7.1. The right-turning vehicles of signal group 2 conflict with the pedestrians of signal group 38. However, these signal groups may be permitted at the same time (as long as it results in a safe situation); see for example Figure 7.3 for a (partial) signal group diagram that result in a safe situation and a (partial) signal group diagram that results in an unsafe situation. Consider the (partial) signal group diagram in Figure 7.3a. For this pre-timed controller, signal group 38 switches to green during a green interval of signal group 2. The vehicles from signal group 2 that depart before signal group 38 becomes green, might not expect to encounter any pedestrians; this may result in an unsafe situation. For the signal group diagram in Figure 7.3b, signal group 38 becomes green a few seconds before signal group 2 receives a green indication. Since, the vehicles waiting at signal group 2 have a clear view of one of the traffic lights of signal group 38, the right-turning vehicles from signal group 2 have seen the green indication of signal group 38 and are alert for pedestrians that cross leg 4. Moreover, the second realization of signal group 38 starts a few seconds after the realization of signal group 2 has ended. As a consequence, the right-turning vehicles of signal group 2 have passed the pedestrian crossing before this second realization of signal group 38 starts.

The two examples of Figure 7.3 motivate the following constraint: for each effective green interval k of signal group 2, prohibit signal group 38 to switch to effective green during the interval $(t(\textcircled{2}_k) - I_1, t(\textcircled{2}_k) + I_2)$, where I_1 and I_2 are input parameters. This constraint is very similar to a conflict between signal group 2 and signal group 38; in that case for each effective green interval k of signal group 2, signal group 38 may not be effective green (instead of 'switch to effective green') during the interval $(t(\textcircled{2}_k) - \underline{c}_{38,2}, t(\textcircled{2}_k) + \underline{c}_{2,38})$. We can formulate the desired constraint by considering signal groups 2 and 38 to be conflicting and setting the parameters (that concern this conflict) as described below. When signal group 2 and signal group 38 are conflicting, the following constraints should be satisfied amongst other constraints:

- The minimum amount of time between signal group 2 becoming effective red and signal group 38 becoming effective green is $\underline{c}_{2,38}$ seconds. For the desired constraint that prohibits signal group 38 to switch to green during the interval $(t(\textcircled{2}_k) - I_1, t(\textcircled{2}_k) + I_2)$, this minimum amount of time should be I_2 . Therefore, we then have $\underline{c}_{2,38} := I_2$.
- The minimum amount of time between signal group 38 switching to effective red and signal group 2 becoming effective green is $\underline{c}_{38,2}$ seconds. For the desired constraint we have no such lower bound. Therefore, we then set $\underline{c}_{38,2} := -\bar{T}$; then no constraint has to be satisfied on the time between signal group 38 switching to effective red and signal group 2 becoming effective green.



(a) The right-turning vehicles from signal group 2 that depart before signal group 38 becomes green may not expect to encounter any pedestrians from signal group 38. As a consequence this signal group diagram may result in an unsafe situation.



(b) The vehicles waiting at signal group 2 have a clear view of one of the traffic lights of signal group 38. Therefore, the right-turning vehicles from signal group 2 have seen the green indication of signal group 38 and are alert for pedestrians that cross leg 4. Moreover, the right-turning vehicles from signal group 2 are not expected to encounter any pedestrians that depart during the second realization of signal group 38. As a consequence this signal group diagram may result in a safe situation.

Figure 7.3: Signal group diagrams for signal groups 2 and 38 of the intersection in Figure 7.1 that may result in safe or unsafe situations.

- The well-posedness constraints (4.7f)–(4.7g) force the time between signal group 38 switching to effective green and signal group 2 switching to effective green to be at least $\epsilon > 0$ seconds, where ϵ is some (small) positive value. For the desired constraint, this minimum amount of time is I_1 (instead of ϵ) seconds.

Remark 7.5. We can use the standard methodology of Chapter 4 to find an integral cycle basis.

7.3 Obtaining multiple signal group diagrams

In practice, returning a single signal group diagram may not be enough (even if this signal group diagram is optimal); traffic engineers may want to compute multiple signal group diagrams so that they can compare these signal group diagrams and assess their quality by using their expertise. In this section we show that we can exclude certain solutions during optimization. We can then for example compute the k best signal group diagrams (w.r.t. some objective function) by first computing the optimal signal group diagram, subsequently computing the second best signal group diagram by again solving the optimization problem but this time excluding the best signal group diagram, et cetera.

We first show that we can reformulate each of the proposed mixed-integer programming problems by using only real-valued design variables \mathbf{x} and binary variables \mathbf{b} . To this end, we rewrite each integral-valued design variable z as a weighted sum of binary variables. First we obtain a (finite) lower bound \underline{z} and a (finite) upper bound \bar{z} on the value of each integral-valued design variable z . Note that each integral-valued design variable is a

variable z_C associated with a cycle periodicity constraint:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2) = z_C.$$

Each of the variables $\gamma(\varepsilon_1, \varepsilon_2)$ is restricted to some finite interval, e.g., $\gamma(\textcircled{i}_k, \textcircled{i}_k)$ is included in the interval $[0, 1)$. Therefore, from this cycle periodicity constraint we can obtain a lower bound and an upper bound on the value of z_C . As a consequence, we can write each such variable z as a weighted sum of binary variables b_i , $i = 1, \dots, k$:

$$z := \underline{z} + 2^0 b_1 + 2^1 b_2 + \dots + 2^{k-1} b_k.$$

for some k . This weighted sum is subject to the constraint $\underline{z} + 2^0 b_1 + 2^1 b_2 + \dots + 2^{k-1} b_k \leq \bar{z}$. For example, an integral variable z for which $\underline{z} := -3$ and $\bar{z} = 2$ can be written as $z := -3 + b_1 + 2b_2 + 4b_3$. Note that this weighted sum can attain all integral values $-3, -2, \dots, 4$; we can restrict this range by including the constraint $-3 + b_1 + 2b_2 + 4b_3 \leq 2$.

As we have proved, we can assume w.l.o.g. that the optimization problem is formulated with only real-valued design variables \mathbf{x} and binary-valued design variables \mathbf{b} . Consider some solution that we want to exclude. Let \mathcal{B}^+ be the set of binary variables that have a value of one for this solution and let \mathcal{B}^- be the set of binary variables that have a value of zero for this solution. We can then exclude any signal group diagram for which the binary variables have the same values as the vector \mathbf{b} by including the following constraint:

$$\sum_{b \in \mathcal{B}^+} b - \sum_{b \in \mathcal{B}^-} b \leq |\mathcal{B}^+| - 1.$$

Note that the above constraint excludes only the solution for which all binary variables in \mathcal{B}^+ have a value of one and all binary variables in \mathcal{B}^- have a value of zero. We can exclude multiple vectors of binary values \mathbf{b} by including one such constraint for each such vector.

7.4 Optimize only a part of the signal group diagram

Traffic engineers may want to have more influence in the design process of a signal group diagram. For example, by filling in a partial signal group diagram and let the optimization procedure fill in the blanks, or by changing (by hand) an optimized signal group diagram, fixing the part of the signal group diagram that seems satisfactory, and again optimizing the part of the signal group diagram that is not satisfactory yet. To this end, we consider the following problem. Consider an intersection and let $G = (V, A)$ be the associated constraint graph. For this intersection we are given a partial signal group diagram. In other words, we are given a period duration T and a set of events V_f , where for each event $\varepsilon \in V_f$ we are given a time $t(\varepsilon)$ at which this event must be scheduled. The objective is to optimize a signal group diagram for which each event $\varepsilon \in V_f$ occurs at its given time $t(\varepsilon)$.

Define $f(\varepsilon) := t(\varepsilon)/T$, $\varepsilon \in V_f$ as the fraction at which event ε must take place. Consider an arc $(\varepsilon_1, \varepsilon_2)$ of the constraint graph G for which $\varepsilon_1, \varepsilon_2 \in V_f$. Recall that by definition:

$$\gamma(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2) \quad (7.3)$$

for some integer $z(\varepsilon_1, \varepsilon_2)$. The variable $\gamma(\varepsilon_1, \varepsilon_2)$ is subject to some lower bound, which we denote by $\underline{\gamma}(\varepsilon_1, \varepsilon_2)$. For example, when the variable $\gamma(\varepsilon_1, \varepsilon_2)$ corresponds to an effective green interval of signal group i , then $\underline{\gamma}(\varepsilon_1, \varepsilon_2) := \underline{g}_i/T$. We can fix $\gamma(\varepsilon_1, \varepsilon_2)$ to the smallest value that exceeds $\underline{\gamma}(\varepsilon_1, \varepsilon_2)$ and satisfies (7.3); this fixes the relative timing between the periodic event ε_1 and the periodic event ε_2 .

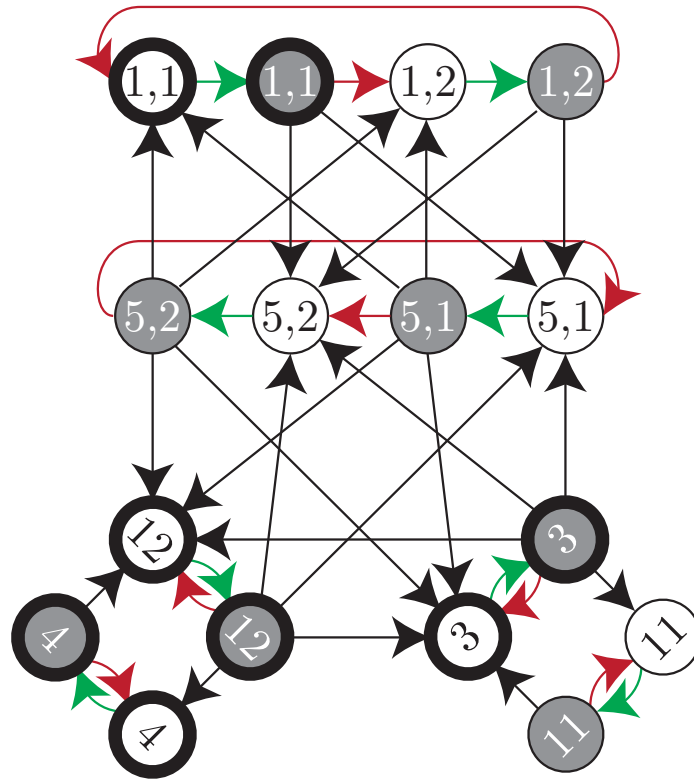
Fixing these variables $\gamma(\varepsilon_1, \varepsilon_2)$ for each arc may not be enough. Consider for example the constraint graph G visualized in Figure 7.4. For this example we have:

$$V_f = \{\textcircled{1}_1, \textcircled{1}_1, \textcircled{3}_1, \textcircled{3}_1, \textcircled{4}_1, \textcircled{4}_1, \textcircled{12}_1, \textcircled{12}_1\}.$$

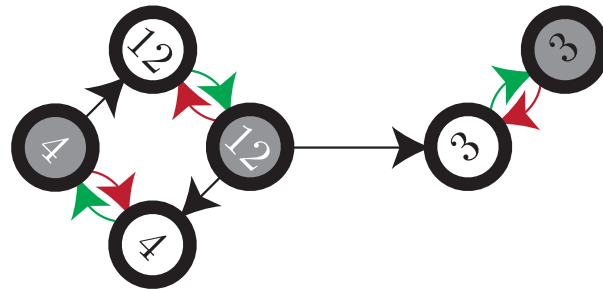
In Figure 7.4b we have visualized the subgraph of the constraint graph G restricted to only the vertices $v \in V_f$. This subgraph consists of two connected components, which are the components:

$$V_f^1 := \{\textcircled{1}_1, \textcircled{1}_1\} \text{ and } V_f^2 := \{\textcircled{3}_1, \textcircled{3}_1, \textcircled{4}_1, \textcircled{4}_1, \textcircled{12}_1, \textcircled{12}_1\}.$$

While the relative timing between each two events $\varepsilon_1 \in V_f$ and $\varepsilon_2 \in V_f$ that are in the same connected component of graph G has been fixed, this relative timing is not fixed for two events that are in different connected components. In general, we can also fix the relative timing between two events that are in different connected components by repeating the following process. Consider two connected components of the subgraph that is restricted to only the vertices V_f . Add an arc $(\varepsilon_1, \varepsilon_2)$ that connects these two connected components. With this additional arc we associate the variable $\gamma(\varepsilon_1, \varepsilon_2)$, which we fix to the value $f(\varepsilon_2) - f(\varepsilon_1)$. The resulting subgraph has one component less. By repeating this process until the subgraph has a single connected component, we can fix the relative timings of each two events in V_f . We can find an integral cycle basis of the associated constraint graph G with Algorithm 2. Subsequently, we can optimize (part of) the signal group diagram with the optimization framework proposed in this thesis. From this solution we can obtain the desired signal group diagram as follows. For some event $\varepsilon \in V_f$ fix the time at which this event occurs to the given time $t(\varepsilon)$ and use the iterative scheme given in Section 3.2.1 to find all other event times $t(\varepsilon)$, $\varepsilon \in V$.



(a) Constraint graph G associated with the T-junction introduced in Section 3.2.1 for which $\overline{K}_1 = \overline{K}_5 = 2$ and $\overline{K}_i = 1$ for all other signal groups i . The events in V_f are visualized in bold.



(b) Subgraph of the constraint graph G restricted to only the vertices in V_f .

Figure 7.4: A constraint graph G and the subgraph restricted to only the vertices $V_f = \{(1)_1, (1)_1, (3)_1, (3)_1, (4)_1, (4)_1, (12)_1, (12)_1\}$. The white (grey) vertex with the text i, k denotes the event $(i)_k$ ($(i)_k$). The white (grey) vertex with the text i denotes the event $(i)_1$ ($(i)_1$).

7.5 Intersection analysis

In the Netherlands it is common to perform an analysis before designing a traffic light controller, see Section 8.5 of (Wilson, 2014). During this step, the capacity of the intersection is assessed and a sensitivity analysis is performed. To this end, the arrival rates are varied by applying some growth factor. As a function of this growth factor, a lower bound is computed on the period duration T that is at least required by any feasible integral signal group diagram. The purpose of this lower bound is to quickly perform a capacity analysis and a sensitivity analysis before designing an integral signal group diagram; if the capacity of the intersection is insufficient, other measures may be needed before designing a traffic light controller.

In Section 8.5 of (Wilson, 2014) it is mentioned that the accuracy of this lower bound is not always up to par. With the framework proposed in this thesis it is also possible to quickly perform this analysis in a more rigorous manner. To this end, for each scaling factor β we compute the signal group diagram with the smallest period duration amongst the signal group diagrams that satisfy $\text{Prop}(\beta)$. As we have seen in Chapter 5, this gives a lower bound on the period duration of any integral signal group diagram that can handle a growth factor of β . Moreover, usually we can obtain an integral signal group diagram (with the same period duration) from this signal group diagram by solving a rounding MILP problem.

In Figure 7.5 we give the computed lower bound on the period duration as a function of the growth factor β for an example; this example is the intersection L1 from (Fleuren and Lefebvre, 2016a). This figure can be obtained as follows. Let $\underline{T}(\beta)$ be the smallest period duration for which a feasible signal group diagram exists that satisfies $\text{Prop}(\beta)$. Note that $\underline{T}(\beta)$ is a non-decreasing and piecewise constant function of the growth factor β , see also Figure 7.5. We exploit this property to iteratively find the values $\underline{T}(\beta)$, $\beta = 1, 1 + \epsilon, 1 + 2\epsilon, 1 + 3\epsilon, \dots$, where $\epsilon > 0$ denotes the accuracy of the resulting figure; the accuracy ϵ equals 0.001 for Figure 7.5. Initially we set the growth factor to one, i.e., $\beta := 1$. Subsequently we iteratively increase this growth factor by the accuracy ϵ , i.e., $\beta := \beta + \epsilon$. In each iteration we find the value for $\underline{T}(\beta)$ and a corresponding signal group diagram; we denote this signal group diagram by $\text{sgd}(\beta)$. We can find $\underline{T}(\beta)$ and the corresponding signal group diagram $\text{sgd}(\beta)$ by scaling the arrival rates with a growth factor of β and using Algorithm 1. Before optimizing over all (feasible and integral) signal groups, we can also first check if a signal group diagrams exists that satisfies $\text{Prop}(\beta)$ and for which:

- the period duration equals $\underline{T}(\beta - \epsilon)$, and
- the values of the integral-valued and binary-valued design variables equal that of signal group diagram $\text{sgd}(\beta - \epsilon)$.

From the monotonicity of $\underline{T}(\beta)$ it follows that $\underline{T}(\beta) \geq \underline{T}(\beta + \epsilon)$. Therefore, if such a signal group diagram exists this implies that $\underline{T}(\beta) = \underline{T}(\beta + \epsilon)$. We can check this by

solving a linear programming problem; for this linear programming problem the integral-valued design variables, the binary-valued design variables and the period duration are fixed to the mentioned values. If such a signal group diagram does not exist, we use (the relatively slow) Algorithm 1 to find $\underline{T}(\beta)$ and the corresponding signal group diagram. The algorithm that is proposed to find the values $\underline{T}(\beta)$, $\beta = 1, 1 + \epsilon, 1 + 2\epsilon, \dots$ terminates if in some iteration we find that $\underline{T}(\beta)$ is infinite, i.e., when for some β no feasible (and integral) signal group diagram exists; then also for all larger growth factors no feasible (and integral) signal group diagram exists.

Depending on the solver that is used, Figure 7.5 can be obtained by solving between 222 and 229 linear programming problems and between 26 and 35 mixed-integer programming problems. When using the solver GUROBI (Gurobi Optimization, Inc., 2015), CPLEX (International Business Machines Corp, 2015) respectively SCIP (Achterberg, 2009), solving these optimization problems takes ~ 3 , ~ 18 and ~ 16 seconds on a computer with specifications: Intel i5-4300U CPU @1.90GHZ with 16.0GB of RAM.

Remark 7.6. *Note that we can without loss of generality set the lower bound $\underline{T} := \underline{T}(\beta - \epsilon)$ when searching for $\underline{T}(\beta)$, $\beta > 1$. As a consequence, the number of LP problems and MILP problems that needs to be solved to find $\underline{T}(\beta)$ may decrease. The mentioned computation times and number of LPs and MILPs that need to be solved to obtain Figure 7.5, all were obtained by using this tighter lower bound \underline{T} .*

7.6 Input parameters that depend on the optimization outcome

The optimization framework formulated in this thesis is based on some assumptions. For example, during an effective green interval of signal group $i \in \mathcal{S}$ traffic is assumed to depart from queue $q \in \mathcal{Q}_i$ at a constant rate of μ_q (the saturation flow rate) unless queue q is empty. However, this maximum departure rate is not constant at all; it depends on the signal group diagram. We explain this using an example. Consider signal group 11 of the intersection in Figure 7.1. We motivate that the maximum departure rate at the queue $q \in \mathcal{Q}_{11}$ differs between the partial signal group diagram in Figure 7.2d and the partial signal group diagram in Figure 7.2c. In the former situation the traffic from signal group 11 can cross the intersection without being hindered. However, in the latter situation this departure process might not go as fluently; left-turning vehicles from signal group 11 have to yield the right-of-way to vehicles (going straight) of signal group 5, which effects the maximum departure rate at queue $q \in \mathcal{Q}_{11}$. Thus, the maximum rate at which traffic can depart is not constant; it depends on the signal group diagram and, as a consequence, it depends on the optimization outcome.

Also other parameters that are assumed to be fixed and given may depend on the optimization outcome. For example, the minimum clearance times are assumed to be

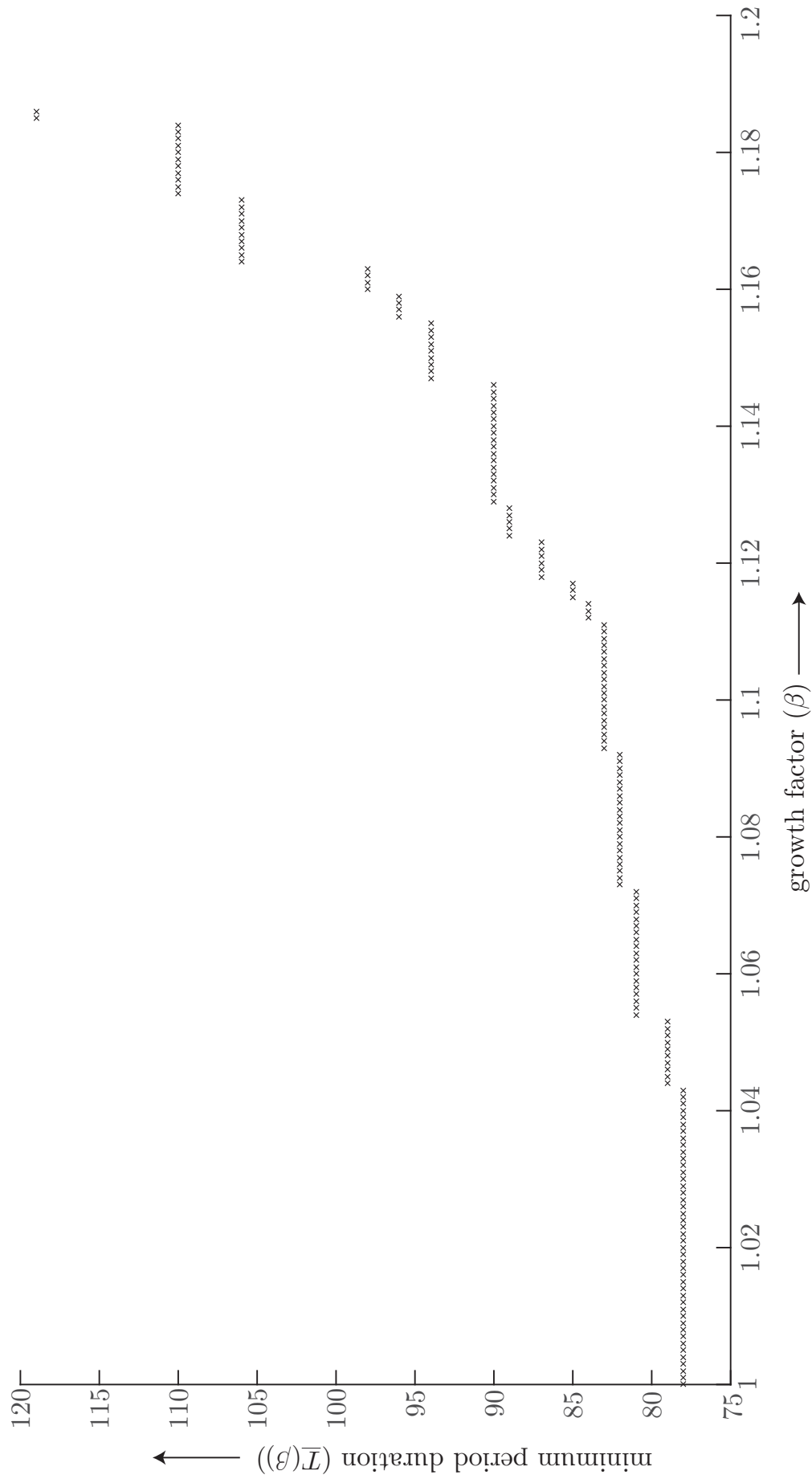


Figure 7.5: The minimum period duration of any signal group diagram satisfying $\text{Prop}(\beta)$ as a function of the growth factor β for the intersection $L1$ from (Fleuren and Lefeber, 2016a) when $\underline{K}_i = \bar{K}_i = 1$ for each signal group $i \in S$.

given for the optimization framework proposed in Chapter 6. However, in that chapter we simultaneously optimize the layout of the intersection and a corresponding signal group diagram, while the minimum clearance times may depend on the layout of the intersection. Even the arrival rates may depend on the optimized outcome as road users tend to avoid an intersection for which the waiting times are expected to be large.

When the input parameters indeed depend on the optimization outcome, we can take a more 'heuristic' and 'engineering' approach by performing multiple design iterations. First, the parameters are approximated. Subsequently, these parameters are considered to be fixed and the optimization procedure is performed. In the next design iteration, the values of the parameters are reevaluated and another optimization procedure is performed. These design iterations are performed until a satisfactory result is obtained.

Remark 7.7. *The above procedure might not converge. Therefore, after some finite number of design iterations this procedure may have to be aborted.*

7.7 Summary

The design process of a (semi)actuated controller consists of multiple steps. This thesis contributes to the following design step: the design (or optimization) of a signal group diagram for an intersection in isolation. We have elaborated on several topics that may be relevant during this design process. First, we have elaborated on some constraints that might be relevant in practice, but that were not yet considered in this thesis. We have shown how to include these constraints in the mixed-integer linear programming formulation that is proposed in this thesis. Subsequently, we have shown how to find multiple signal group diagrams, instead of returning only the optimal one; traffic engineers may want to compute multiple signal group diagrams so that they can compare these signal group diagrams and assess their quality by using their expertise. Subsequently, we considered the problem of optimizing only the part of a signal group diagram that is not satisfactory yet, while maintaining the part of the signal group diagram that is satisfactory. We have also shown how to (quickly) perform a sensitivity analysis. It is common to do such a sensitivity analysis in the Netherlands. For this sensitivity analysis we require the minimum period duration required by any integral signal group diagram as a function of the growth factor of the arrival rates. Finally, we note that some parameters that we consider to be fixed and given, actually depend on the optimization outcome. For example, when we optimize the layout of an intersection, the clearance times may depend on the optimized layout. We propose an engineering approach that iteratively updates these input parameters to find a satisfactory solution. In the next (and final) chapter we give our conclusions and recommendations.

Conclusions and recommendations

8.1 Conclusions

Modern day society has a major demand for road transport. On the one hand, for our personal transportation and, on the other hand, for the transportation of freight. This demand for road transport is expected to increase even further in the upcoming years due to factors as population growth, economic growth, increasing income levels, and rising car ownership. These large demands for road transport come at a cost. Congestion of roads is a serious problem with significant consequences; it affects our daily living, increases the time we spend on the road, the quality of the air we breathe, the price tag on the products we buy, the costs of the trips we make, et cetera. With this thesis we contribute to the more efficient use of existing road infrastructure and, as a consequence, to the mitigation of this congestion problem. To be more specific, this thesis concerns the optimization of pre-timed control at isolated conventional intersections. Such a pre-timed controller may form the basis for a (semi)actuated controller, which is a type of traffic light control that is often used in practice.

First, we have shown in Chapter 2 how to mathematically model an intersection. We have modelled each traffic light that visualizes three colors (green, yellow and red) with only two modes: effective green and effective red. Furthermore, we have modelled the traffic that is waiting at the intersection by using (first-in-first-out) queues. For each of these queues we can approximate the average delay that road users experience at this queue with one of the many known approximations, e.g., the one from (van den Broek et al., 2006).

Subsequently, in Chapter 3 we have used this mathematical model of the intersection to formulate a novel approach to optimize signal group diagrams for isolated intersections. In that chapter we assume that the number of realizations of each signal group equals one and we simultaneously optimize: the period duration of the signal group diagram, when each effective green intervals starts, and when each effective green interval ends. Possible objective functions of the optimization framework are: minimizing the period duration of the signal group diagram, maximizing the capacity of the intersection, and minimizing the

average delay that road users experience at the intersection. We have compared this novel approach with the currently existing group-based approaches. For the currently existing approaches, the binary-valued design variables are possibly defined ambiguously, i.e., multiple solutions may be associated with the same signal group diagram. The integral-valued design variables of the novel formulation aggregate the binary-valued design variables of the currently existing group-based approaches. As a consequence, the novel formulation does not possess this undesirable property, i.e., the design variables are defined unambiguously for each signal group diagram. With a large numerical study we compare the computation times of the novel approach with the currently existing approaches. The novel approach is superior. The difference in computation times is especially large for the minimization of the average delay that road users experience at the intersection; for the large intersections L1, L2, L3 and L4 from (Fleuren and Lefebvre, 2016a) we have seen an average speed up of a factor 18 when minimizing the average delay that road users experience at these intersections.

In the subsequent chapter we have extended this optimization framework to allow a signal group to have multiple realizations. With this extension we can simultaneously optimize: the period duration of the signal group diagram, the number of realizations of each signal group, when each of these realizations start, and when these realizations end. For some realizations the optimization decides whether to include this realization in the signal group diagram or not. For each of these realizations we use a binary-valued design variable to 'switch' this realization on or 'switch' it off, i.e., with this binary-valued design variable the optimization can decide whether to include this realization or not. These binary variables enable us to also optimize the number of realizations that each signal group has. In this chapter we have considered the same objective functions as the ones considered in the previous chapter. In the previous chapter, we could use the approximate formulae of for example (Miller, 1963; van den Broek et al., 2006; Webster, 1958) to approximate the delay that road users experience at the intersection. However, all these formulae assume that a signal group receives only one realization. Therefore, in Chapter 4 we have extended these approximations, in a straightforward manner, to allow a signal group to have multiple realizations. We have tested the optimization framework for a large test set. Allowing some signal groups to receive multiple realizations seems to have little (to no) effect on the objective value when minimizing the period duration and when maximizing the capacity of the intersection. However, when minimizing the delay that road users experience at the intersection, the average delay can decrease substantially by allowing several signal groups to have multiple realizations; for some intersections we have seen a decrease of over 10 percent when we allow only the four most heavily loaded signal groups to have multiple (at most two) realizations. By using either expertise or a trial-and-error approach to determine which signal groups should receive multiple realizations, an even larger improvement may be possible.

In Chapter 5, we have considered the optimization of integral signal group diagrams. For such an integral signal group diagram the period duration is integral and all switches

to green, yellow and red are scheduled at an integral second. These integral signal group diagrams are desired in practice because they are clear, presentable and easy to work with. We split the optimization of integral signal group diagrams in two steps. In the first step we search for the optimal signal group diagram that satisfies some structural property. This structural property is also satisfied by all integral signal group diagrams.

This first step does not necessarily return an integral signal group diagram. Therefore, in the second step we solve a (relatively simple) MILP problem, which we call the *rounding MILP problem*. This rounding MILP problem attempts to find an integral signal group diagram by rounding each green time and each red time (of the signal group diagram returned by the first step) down or up. We have proved that in some situations this two-step approach will always find the optimal integral signal group diagram, e.g., when each signal group is allowed to have only one realization. For the large numerical study that we have performed, this rounding MILP problem was always feasible. For the test cases that minimize the period duration or maximize the capacity of the intersection, we were always able to find the optimal integral signal group diagram. When minimizing the delay that road users experience, we were always able to find a signal group diagram that is either optimal or close to optimality.

For the two-step approach that minimizes the average (weighted) delay that road users experience we have proposed two different approaches to search for the optimal integral signal group diagram. The difference between these two approaches is the first step of the two-step approach. The first approach is a brute force approach. This brute force approach solves one MILP problem for each integral period duration in the interval $[\underline{T}, \overline{T}]$. The other approach is a heuristic approach. The heuristic method is much faster than the brute force algorithm as it only has to solve a MILP problem for a fraction of the integral period duration in the interval $[\underline{T}, \overline{T}]$. Moreover, the results that are returned by this heuristic method are very promising.

In Chapter 6 we considered the optimization of the layout of the intersection. In that chapter we optimize, besides a signal group diagram, also the number of lanes at each leg, which of these lanes are arrival lanes, and which lane-use arrows are present at each of the arrival lanes. With the proposed MILP problem we can answer questions like: what should the lane-use arrows be such that the capacity of the intersection is maximized and what is the minimum number of lanes that the intersection requires to have sufficient capacity? To answer these questions we have to simultaneously optimize the layout of the intersection and a signal group diagram for this intersection. To optimize the signal group diagram, we use the same formulation as the one proposed in Chapter 4. We introduce additional binary variables to optimize the layout of this intersection. These binary variables are also used to force two vehicular movements to have the same indication whenever they are both permitted on the same arrival lane.

We have considered two different objective functions. The first one is the maximization of the capacity of the intersection. With this objective function we find the layout of the intersection that can handle the largest increase in arrival rates. The second objective

function is the minimization of the number of lanes present at the intersection. With that objective function we can find the smallest intersection that can handle some specified growth factor.

Chapter 7 concerns some practical issues that were not yet addressed in this thesis. First, we have elaborated on some constraints that might be relevant in practice, but were not yet considered in previous chapters. Thereupon we have considered the problem of finding multiple signal group diagrams, instead of only the optimal one; traffic engineers may want to compute multiple signal group diagrams so that they can compare these signal group diagrams and assess their quality by using their expertise. Moreover, traffic engineers might want to change parts of the signal group diagram by hand. To assist the traffic engineer in this design process, we have considered the problem of optimizing only the part of a signal group diagram that is not satisfactory yet, while maintaining the part of the signal group diagram that is satisfactory. We have also shown how to (quickly) perform a sensitivity analysis. This sensitivity analysis requires (a lower bound on) the period duration of any integral signal group diagram as a function of the growth factor of the arrival rates. Such a sensitivity analysis is commonly performed in the Netherlands. Finally, we note that some parameters that we consider to be fixed and given in this thesis, may actually depend on the optimization outcome. We propose an engineering approach that iteratively updates these input parameters to find a satisfactory solution.

With the different chapters of this thesis we provide an insightful and comprehensive overview on the optimization of pre-timed controllers and their usefulness. This thesis can be used to help practicing traffic engineers in the design process of traffic light controllers (and intersections).

8.2 Recommendations

In this section we elaborate on some recommendations for further research.

Quality of the proposed delay formula In Chapter 4 we have allowed a signal group to have multiple realizations. The delay that road users experience at a traffic light under pre-timed control can be approximated with some of the well-known approximations. However, the currently existing approximate formulae assume that each signal group has only one realization. Therefore, we have extended these approximate formulae, in a straight forward manner, to allow each signal group to have multiple realizations. We recommend to assess the quality of the proposed extended formulae. In what situations do these approximations perform very well? Are there situations for which the performance is insufficient? Perhaps this insight results in a better approximation. Recall however that the proposed extension has the following desirable properties. First, the resulting approximation is convex in the variables \mathbf{T}' and γ . Second, we can break this extended formula into different terms. When the period duration is fixed, each of these terms

depends only on one design variable (or on a sum of design variables). Hence, we need relatively few piecewise linear segments to approximate this formula by using piecewise linear functions. As a consequence, we can (efficiently) formulate the minimization of the average weighted delay that road users experience as a mixed-integer programming problem.

Improving the objective function of the rounding MILP problem In Section 5.7 we have formulated a *rounding MILP problem*. With this rounding MILP problem we can obtain an integral signal group diagram from a signal group diagram that is not integral. In that section we have also formulated an objective function so that we can find the rounding that minimizes the average (weighted) delay that road users experience. To formulate this objective function we have assumed that each queue is emptied during an effective green interval. This property is satisfied by the signal group diagram that we try to round. This property does however not necessarily hold for the integral signal group diagram that is returned by the two-step approach. It is possible to improve this objective function so that we do take into account that a queue (possibly) is not emptied during an effective green interval. To this end, we reformulate the deterministic term of each signal group $i \in \mathcal{S}$ with multiple realizations, i.e., $K_i > 1$ as follows. Consider some signal group $i \in \mathcal{S}$ with multiple realizations, i.e., $K_i > 1$. We split the deterministic delay term d_i^{det} in K_i terms (one for each realization):

$$d_i^{\text{det}} = d_{i,1}^{\text{det}} + \dots + d_{i,K_i}^{\text{det}},$$

For reasons of simplicity we assume that the effective green (effective red) mode coincides with the green (red) indication, i.e., the lost times are zero. For the moment we restrict our attention to the term $d_{i,k}^{\text{det}}$. For ease of notation we define $d := d_{i,k}^{\text{det}}$. Let \tilde{g} (\tilde{r}) be the k th green time (the k th red time) of signal group $i \in \mathcal{S}$ for the signal group diagram that we attempt to round; the red time \tilde{r} precedes the green time \tilde{g} . Let \mathbf{g} (\mathbf{r}) be the k th green time (the k th red time) of signal group $i \in \mathcal{S}$ for the integral signal group diagram that we find by solving the rounding MILP problem. We have the following inclusions: $\mathbf{r} \in \{\lfloor \tilde{r} \rfloor, \lceil \tilde{r} \rceil\}$ and $\mathbf{g} \in \{\lfloor \tilde{g} \rfloor, \lceil \tilde{g} \rceil\}$. Let r_j be defined such that $r_j := \lfloor \tilde{r} \rfloor + j$; note that the value for j defines whether the k th red time is rounded down ($j = 0$) or rounded up ($j = 1$). Similarly, we define $g_j := \lfloor \tilde{g} \rfloor + j$; the value for j defines whether the k th green time is rounded down ($j = 0$) or rounded up ($j = 1$). Let the delay $d_{j,l}$ be an approximation for the deterministic term d for the situation that $\mathbf{r} = r_j$ and $\mathbf{g} = g_l$. We can approximate the deterministic delay term d by using piecewise linear functions. To this end we use an auxiliary variable \mathbf{d} . We would like the auxiliary variable \mathbf{d} to equal $d_{j,l}$ when $\mathbf{r} = r_j$ and $\mathbf{g} = g_l$. Recall that the binary-valued design variable $\mathbf{u}_{i,k}$ equals zero whenever $\mathbf{r} = r_0$ and this binary-valued design variable equals one whenever $\mathbf{r} = r_1$. We can include the following linear constraint:

$$\mathbf{d} \geq \frac{d_{0,1} - d_{0,0}}{g_1 - g_0} \mathbf{g} + \frac{d_{0,0}g_1 - d_{0,1}g_0}{g_1 - g_0} - L\mathbf{u}_{i,k}.$$

where L is a large positive number. This inequality implies that the deterministic delay term d must be at least $d_{0,0}$ when $\mathbf{r} = r_0$ and $\mathbf{g} = g_0$, and it must be at least $d_{0,1}$ when $\mathbf{r} = r_0$ and $\mathbf{g} = g_1$. This constraint becomes redundant whenever $\mathbf{u}_{i,k} = 1$. Similarly, we include the following inequality:

$$\mathbf{d} \geq \frac{d_{1,1} - d_{1,0}}{g_1 - g_0} \mathbf{g} + \frac{d_{1,0}g_1 - d_{1,1}g_0}{g_1 - g_0} - L(1 - \mathbf{u}_{i,k}).$$

where L is a large positive number. This equality implies that the deterministic delay term d must be at least $d_{1,0}$ when $\mathbf{r} = r_1$ and $\mathbf{g} = g_0$, and it must be at least $d_{1,1}$ when $\mathbf{r} = r_1$ and $\mathbf{g} = g_1$. This constraint becomes redundant whenever $\mathbf{u}_{i,k} = 0$. These two additional constraints allow the auxiliary variable \mathbf{d} to equal $d_{j,l}$ (but not less than $d_{j,l}$) when $\mathbf{r} = r_j$, $j = 0, 1$ and $\mathbf{g} = g_l$, $l = 0, 1$. Note that these additional constraints are linear. Furthermore, note that when the values of the binary variables in \mathbf{u} are fixed, the objective function can be written as an (ordinary) linear objective function. As a consequence, we can still find an integral signal group diagram by solving the rounding MILP problem with the novel objective function.

What remains is to find expressions for $d_{0,0}$, $d_{0,1}$, $d_{1,0}$ and $d_{1,1}$. We only show how to find $d_{0,0}$ the other ones can be obtained analogously. To distinguish between the k th realization and the $k + 1$ st realization of signal group i , we let \tilde{r}_k denote the k th red time and \tilde{g}_k denote the k th green time of signal group i for the signal group diagram that we try to round. Moreover, we let \mathbf{r}_k denote the k th red time and \mathbf{g}_k denote the k th green time of signal group i for the signal group diagram returned by the rounding MILP problem. The value for $d_{0,0}$ then corresponds to $\mathbf{r}_k = \lfloor \tilde{r}_k \rfloor$ and $\mathbf{g}_k = \lfloor \tilde{g}_k \rfloor$. Signal group $i \in \mathcal{S}$ controls the access to the intersection of the queues $q \in \mathcal{Q}_i$. We can write the term $d_{0,0}$ as $d_{0,0} = \sum_{q \in \mathcal{Q}_i} w_q d_{0,0}^q$, where w_q is a weight factor of queue q and $d_{0,0}^q$ is the average delay that road users experience at queue $q \in \mathcal{Q}_i$. Consider such a queue $q \in \mathcal{Q}_i$. We show how to obtain an expression for $d_{0,0}^q$. We distinguish between two different situations: All traffic that arrives at queue q during the k th effective red interval can depart during the k th effective green interval, and all traffic that arrives at queue q during the k th effective red interval cannot depart during the k th effective green interval. In Figure 8.1a we visualize the former situation. The value for $d_{0,0}^q$ then equals the gray area divided by $1/(\lambda_q T)$, which gives:

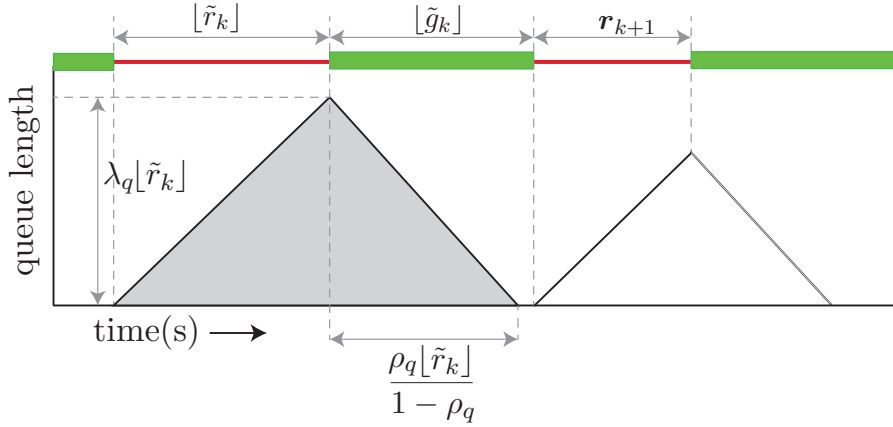
$$d_{0,0}^q := \frac{\lfloor \tilde{r}_k \rfloor^2}{2T(1 - \rho_q)}. \quad (8.1)$$

In Figure 8.1b we visualize the situation that the queue is not emptied during the k th green interval (we do, however, assume that this queue is then emptied during the $k + 1$ st green interval). The value for $d_{0,0}^q$ then equals the sum of the two gray areas divided by $1/(\lambda T)$, which gives:

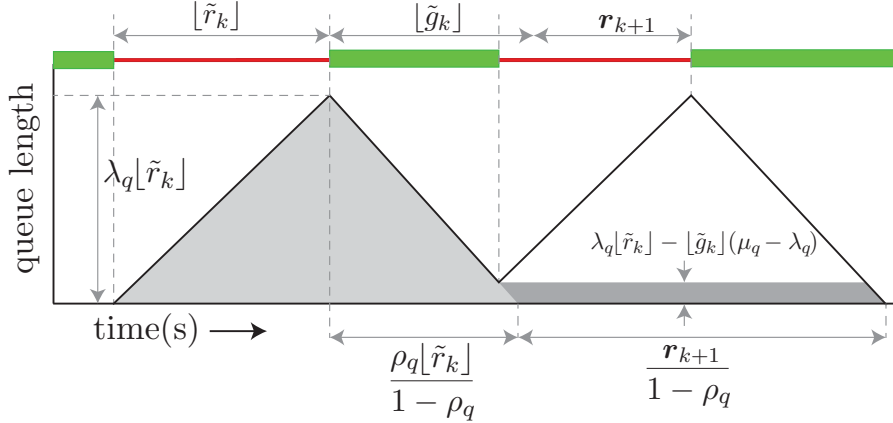
$$d_{0,0}^q := \frac{\lfloor \tilde{r}_k \rfloor^2}{2T(1 - \rho_q)} + \frac{\mathbf{r}_{k+1}(\lambda_q \lfloor \tilde{r}_k \rfloor - \lfloor \tilde{g}_k \rfloor)(\mu_q - \lambda_q)}{T(1 - \rho_1)}. \quad (8.2)$$

We do not yet know the value for \mathbf{r}_{k+1} . However, we know that it is restricted to the interval $[\lfloor \tilde{r}_{k+1} \rfloor, \lceil \tilde{r}_{k+1} \rceil]$. Therefore, we approximate its value by $0.5(\lfloor \tilde{r}_{k+1} \rfloor + \lceil \tilde{r}_{k+1} \rceil)$. Note that the white area in Figure 8.1a and Figure 8.1b corresponds to the deterministic delay term of the $k + 1$ st realization of signal group $i \in \mathcal{S}$.

We have now shown how to formulate the piecewise linear objective function. This objective function does take into account that a queue is possibly not emptied during an effective green interval. A question is: what is the influence of this newly formulated objective function on the result from the integral signal group diagram. How much better is the resulting integral signal group diagram?



(a) Queue length evolution when the queue is emptied during the k th effective green interval.



(b) Queue length evolution when the queue is not emptied during the k th effective green interval.

Figure 8.1: Queue length evolution of queue q when the queue is emptied during the k th effective green time and when the queue is not emptied during the k th effective green interval.

Comparing lane-based approaches In Chapter 6 we have proposed an optimization formulation to simultaneously optimize the layout of an intersection together with a signal group diagram for this intersection. Some methods from the literature are already able to perform this simultaneous optimization, e.g., (Wong and Heydecker, 2011; Wong and Wong, 2003; Wong, 1996; Yan et al., 2014). These existing methods are referred to as lane-based approaches. Some differences with the proposed formulation are that we extend

the novel formulation given in Chapter 4 of this thesis, whereas the other methods extent the currently existing group based approaches; in Chapter 3 we have seen that the novel formulation is superior. Moreover, we also optimize the number of lanes at each leg of the intersection (and the number of realizations of each movement). We recommend to compare the novel formulation to optimize the layout of the intersection with the currently existing lane-based method; to this end we have to fix the number of lanes at each leg of the intersection (and we have to set the number of movements of each movement to one).

Speeding up computations In (Fleuren, 2016) we elaborate on several techniques to reduce the computation time needed to solve optimization problem (4.8), see also Remark 4.5. In that report we also give many recommendations for the further speedup of the optimization problem.

Online updating In practice, actuated controllers often serve the signal groups in a predefined order specified by a phase diagram. Such a phase diagram can be derived from a pre-timed controller and is usually based on forecasted (or historical) arrival rates. If the optimization of pre-timed control is fast enough, we can also compute such a phase diagram and an associated actuated controller in an online manner, i.e., depending on the current traffic situation. We may do so by first computing a pre-timed controller based on the current traffic situation. Subsequently, we can automatically obtain a phase diagram with the method from Appendix A.1. Thereupon, we can automatically derive a set of rules defining when to change the indication of each traffic light. These switching rules may for example be defined as follows. Take the green times specified by the pre-timed controller to be maximum green times. We can end a green interval when its maximum green time is reached or when the associated queue is emptied (whichever one occurs first). A green interval included in phase k of the phase diagram may start whenever the conflicting green intervals included in phase $k - 1$ have ended (and the associated minimum clearance times have elapsed).

We recommend to investigate such online algorithms. Interesting questions that arise are: how should we derive the set of rules (in the third step)? What objective function should we consider in the first step (optimization of the signal group diagram), e.g., should we maximize the capacity of the intersection? Another possible objective function is to maximize the throughput of the intersection during some finite time horizon, e.g., during the next few period durations; to favor the traffic lights with large queues we could consider the weighted throughput, where the weight of each signal group increases proportional to the associated queue length.

Network of signalized intersections In this thesis we have focussed mainly on isolated intersections. To find a pre-timed controller for a network of signalized intersection, we suggest to use the following approach. First, we can optimize a pre-timed controller for

each intersection in isolation. Subsequently, these pre-timed controllers can be synchronized, for example by solving an MILP problem (Gartner et al., 1975; Wunsch, 2008)) or by using an evolutionary algorithm such as the swarm intelligence algorithm from (García-Nieto et al., 2012) or the genetic algorithm proposed in (Zhou and Cai, 2014); besides the offset between the different signal group diagrams, the evolutionary algorithms also optimize the duration of each phase (the sequence of the phases is fixed for each intersection and can be obtained from the pre-timed controller).

Bibliography

- Achterberg, T. (2009). SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41. <http://mpc.zib.de/index.php/MPC/article/view/4>.
- Allsop, R. E. (1971a). Delay-minimizing settings for fixed-time traffic signals at a single road junction. *Journal of the Institute of Mathematics and Its Applications*, 8(2):164–185.
- Allsop, R. E. (1971b). Sigset: A computer program for calculating traffic capacity of signal-controlled road junctions. *Traffic Engineering & Control*, 12:58–60.
- Allsop, R. E. (1972). Estimating traffic capacity of a signalized road junction. *Transportation Research*, 6(3):245–255.
- Allsop, R. E. (1981). Computer program sigset for calculating delay-minimising traffic signal timings description and manual for users. Technical report, University College of London, Transport Studies Group.
- Amaldi, E., Liberti, L., Maculan, N., and Maffioli, F. (2004). Efficient edge-swapping heuristics for finding minimum fundamental cycle bases. In *Experimental and Efficient Algorithms*, pages 14–29. Springer.
- Amitran project (2014). D3. 1: Methodology for classification of ITS. Technical report. <http://www.amitran.eu/assets/02-March-2015/AmitranD31Methodology-for-classification-of-ITS.pdf>.
- Apelberg, B. J., Buckley, T. J., and White, R. H. (2005). Socioeconomic and racial disparities in cancer risk from air toxics in maryland. *Environmental Health Perspectives*, pages 693–699.
- Babisch, W. (2000). Traffic noise and cardiovascular disease: Epidemiological review and synthesis. *Noise and Health*, 2(8):9.
- Babisch, W. (2008). Road traffic noise and cardiovascular risk. *Noise and Health*, 10(38):27.
- Babisch, W., Ising, H., Gallacher, J. E. J., Sweetnam, P. M., et al. (1999). Traffic noise and cardiovascular risk: The caerphilly and speedwell studies, third phase-10-year follow up. *Archives of Environmental Health: An International Journal*, 54(3):210–216.
- Barth, M. and Boriboonsomsin, K. (2009). Traffic congestion and greenhouse gases. *ACCESS Magazine*, 1(35).
- Basner, M., Babisch, W., Davis, A., Brink, M., et al. (2014). Auditory and non-auditory effects of noise on health. *The Lancet*, 383(9925):1325–1332.
- Basner, M., Müller, U., and Elmenhorst, E.-M. (2011). Single and combined effects of air, road, and rail traffic noise on sleep and recuperation. *Sleep*, 34(1):11.
- Cantarella, G. E. and Improta, G. (1988). Capacity factor or cycle time optimization for signalized junctions: A graph theory approach. *Transportation Research Part B: Methodological*, 22(1):1–23.
- Chhajed, D. and Lowe, T. J. (2008). *Building intuition: Insights from basic operations management models and principles*, volume 115, page 82. Springer Science & Business Media.
- Chiou, Y.-C. and Huang, Y.-F. (2013). Stepwise genetic fuzzy logic signal control under mixed traffic conditions. *Journal of Advanced Transportation*, 47(1):43–60.

- Colville, R. N., Hutchinson, E. J., Mindell, J. S., and Warren, R. F. (2001). The transport sector as a source of air pollution. *Atmospheric Environment*, 35(9):1537–1565.
- Department of main roads, Q. g. (2006). *Road planning and design manual*, chapter 13, pages 33–35.
- Dey, K. C., Yan, L., Wang, X., Wang, Y., Shen, H., Chowdhury, M., Yu, L., Qiu, C., and Soundararaj, V. (2016). A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC). *IEEE Transactions on Intelligent Transportation Systems*, 17(2):491–509.
- Djahel, S., Doolan, R., Muntean, G.-M., and Murphy, J. (2015). A communications-oriented perspective on traffic management systems for smart cities: challenges and innovative approaches. *IEEE Communications Surveys & Tutorials*, 17(1):125–151.
- Dratva, J., Phuleria, H. C., Foraster, M., Gaspoz, J.-M., et al. (2012). Transportation noise and blood pressure in a population-based sample of adults. *Environmental Health Perspectives: Journal of the National Institute of Environmental Health Sciences*, 120:50–55.
- DTV consultants (2015). Cocon 9.0 [computer software]. http://www.dtvconsultants.nl/Producten/-_COCON.aspx.
- EENA (2015). eCall - Everything you wanted to ask, but did not know how... http://www.eena.org/download.asp?item_id=111, last accessed november 2016.
- El-Tantawy, S., Abdulhai, B., and Abdelgawad, H. (2013). Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atc): methodology and large-scale application on downtown toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1140–1150.
- El-Tantawy, S., Abdulhai, B., and Abdelgawad, H. (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, 18(3):227–245.
- European commission (2015). *Transport in Figures –Statistical pocketbook 2015*. Luxembourg: Publications Office of the European Union.
- Fleuren, S. (2016). Speed up the optimization of pretimed control. <http://mn.wtb.tue.nl/~lefeber/SpeedUp.pdf>.
- Fleuren, S. and Lefeber, E. (2016a). Data of real-life intersections for fixed-time traffic light control. <https://pure.tue.nl/ws/files/26651263/DataOfRealLifeIntersections.pdf>.
- Fleuren, S. and Lefeber, E. (2016b). Integral fixed-times schedules for traffic light control.
- Fleuren, S. and Lefeber, E. (2016c). Optimizing fixed-time control at isolated intersections: Part I: A single green interval per traffic light. Technical Report DC 2016.067, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven, The Netherlands. Available at http://mn.wtb.tue.nl/~lefeber/do_download_pdf.php?id=170. Submitted to Transportation Science.
- Fleuren, S. and Lefeber, E. (2016d). Optimizing fixed-time control at isolated intersections: Part II: Optimizing the number of green intervals. Technical Report DC 2016.068, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven, The Netherlands. Available at http://mn.wtb.tue.nl/~lefeber/do_download_pdf.php?id=171. Submitted to Transportation Science.
- Fleuren, S. and Lefeber, E. (2016e). Real-life data for the optimization of the geometric layout of an intersection. <https://pure.tue.nl/ws/files/38953928/OptimizingGeometricLayoutExamples.pdf>.
- Fleuren, S. and Lefeber, E. (in preparation). Lane-based optimization of signal group diagrams using cycle periodicity constraints.
- Floud, S., Blangiardo, M., Clark, C., de Hoogh, K., et al. (2013). Exposure to aircraft and road traffic noise and associations with heart disease and stroke in six european countries: A cross-sectional study. *Environ Health*, 12(1):89.
- Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46.
- Gallivan, S. and Heydecker, B. G. (1988). Optimising the control performance of traffic signals at a single junction. *Transportation Research Part B: Methodological*, 22(5):357–370.

- García-Nieto, J., Alba, E., and Olivera, A. C. (2012). Swarm intelligence for traffic light scheduling: Application to real urban areas. *Engineering Applications of Artificial Intelligence*, 25(2):274–283.
- Gartner, N. H., Little, J. D. C., and Gabbay, H. (1975). Optimization of traffic signal settings by mixed-integer linear programming: Part I: The network coordination problem. *Transportation Science*, 9(4):321–343.
- Geroliminis, N., Haddad, J., and Ramezani, M. (2013). Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):348–359.
- Gillen, D. (2016). The role of intelligent transportation systems (ITS) in implementing road pricing for congestion management. In Tom Rye, S. I., editor, *The Implementation and Effectiveness of Transport Demand Management Measures: An International Perspective*, chapter 4, pages 49–74. Routledge.
- Gregoire, J., Qian, X., Frazzoli, E., De La Fortelle, A., and Wongpiromsarn, T. (2015). Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2(2):164–173.
- Grineski, S. E., Collins, T. W., and Chakraborty, J. (2013). Hispanic heterogeneity and environmental injustice: Intra-ethnic patterns of exposure to cancer risks from traffic-related air pollution in Miami. *Population and Environment*, 35(1):26–44.
- Gurobi Optimization, Inc. (2015). Gurobi optimizer reference manual.
- Halonen, J. I., Vahtera, J., Stansfeld, S., Yli-Tuomi, T., et al. (2012). Associations between night-time traffic noise and sleep: The Finnish public sector study. *Environmental Health Perspectives*, 120(10):1391–1396.
- Han, B. (1990). *Some aspects of optimisation of traffic signal timings for time-varying demand*. PhD thesis, University College London (University of London).
- Han, B. (1996). Optimising traffic signal settings for periods of time-varying demand. *Transportation Research Part A: Policy and Practice*, 30(3):207–230.
- Hennessy, D. A. and Wiesenthal, D. L. (1997). The relationship between traffic congestion, driver stress and direct versus indirect coping behaviours. *Ergonomics*, 40(3):348–361.
- Hoffman, A. J. and Kruskal, J. B. (1956). *Integral boundary points of convex polyhedra*, pages 213–246. Princeton University Press.
- Hoffmann, B., Moebus, S., Möhlenkamp, S., Stang, A., et al. (2007). Residential exposure to traffic is associated with coronary atherosclerosis. *Circulation*, 116(5):489–496.
- Hoffmann, B., Moebus, S., Stang, A., Beck, E.-M., et al. (2006). Residence close to high traffic and prevalence of coronary heart disease. *European Heart Journal*, 27(22):2696–2702.
- Hosseini, S. M. and Orooji, H. (2009). Phasing of traffic light at a road junction. *Applied Mathematical Science*, 3(30):1487–1492.
- Improta, G. and Cantarella, G. E. (1984). Control system design for an individual signalized junction. *Transportation Research Part B: Methodological*, 18(2):147–167.
- INRIX, Inc., and CEBR (2014). The future economic and environmental costs of gridlock in 2030. <http://www.cebr.com/reports/the-future-economic-and-environmental-costs-of-gridlock/>.
- International Business Machines Corp (2015). *IBM ILOG CPLEX Optimization studio cplex users manual*. http://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.3.
- Kamal, M. A. S., Imura, J.-i., Hayakawa, T., Ohata, A., and Aihara, K. (2015). Traffic signal control of a road network using milp in the mpc framework. *International journal of intelligent transportation systems research*, 13(2):107–118.
- Kaur, T. and Agrawal, S. (2014). Adaptive traffic lights based on hybrid of neural network and genetic algorithm for reduced traffic congestion. In *Engineering and Computational Sciences (RAECS), 2014 Recent Advances in*, pages 1–5. IEEE.
- Kavitha, T. and Krishna, K. V. (2009). An improved heuristic for computing short integral cycle bases. *Journal of Experimental Algorithmics (JEA)*, 13:14.
- Kavitha, T., Liebchen, C., Mehlhorn, K., Michail, D., et al. (2009). Cycle bases in graphs characterization, algorithms, complexity, and applications. *Computer Science Review*, 3(4):199–243.

- Koukol, M., Zajíčková, L., Marek, L., and Tuček, P. (2015). Fuzzy logic in traffic engineering: a review on signal control. *Mathematical Problems in Engineering*, 2015.
- Krijger, P. (2013). Traffic light prediction for tom tom devices. Master's thesis, Eindhoven University of Technology, the Netherlands.
- Kumar, P. and Seidman, T. I. (1990). Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Transactions on Automatic Control*, 35(3):289–298.
- Künzli, N., Bridevaux, P.-O., Liu, S., Garcia-Esteban, R., et al. (2009). Traffic-related air pollution correlates with adult-onset asthma among never-smokers. *Thorax*.
- Lämmer, S. and Helbing, D. (2008). Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(04):P04019.
- Liebchen, C. (2003). Finding short integral cycle bases for cyclic timetabling. In *Algorithms - ESA 2003*, volume 2832 of *Lecture notes in computer science*, pages 715–726. Springer.
- Liebchen, C. and Peeters, L. (2002). On cyclic timetabling and cycles in graphs. Technical report, Technical Report 761/2002, TU Berlin.
- Lin, M.-H., Carlsson, J. G., Ge, D., Shi, J., and Tsai, J.-F. (2013). A review of piecewise linearization methods. *Mathematical Problems in Engineering*, 2013.
- Lindgren, A., Strohm, E., Montn  mery, P., Nihl  n, U., et al. (2009). Traffic-related air pollution associated with prevalence of asthma and COPD/chronic bronchitis. a cross-sectional study in southern Sweden. *International Journal of Health Geographics*, 8(1):2.
- Liu, R., Liu, H., Kwak, D., Xiang, Y., Borcea, C., Nath, B., and Iftode, L. (2014). Themis: A participatory navigation system for balanced traffic routing. In *2014 IEEE Vehicular Networking Conference (VNC)*, pages 159–166. IEEE.
- Lu, X.-Y., Varaiya, P., Horowitz, R., Su, D., and Shladover, S. (2011). Novel freeway traffic control with variable speed limit and coordinated ramp metering. *Transportation Research Record: Journal of the Transportation Research Board*, (2229):55–65.
- Maranas, C. D. and Zomorodi, A. R. (2016). *Optimization methods in metabolic networks*. John Wiley & Sons.
- McCann, B. (2014). A review of scats operation and deployment in dublin. In *Proceedings of the 19th JCT Traffic Signal Symposium & Exhibition*. JCT Consulting Ltd.
- Miller, A. J. (1963). Settings for fixed-cycle traffic signals. *Operational Research Quarterly*, 14(4):373–386.
- Mingardo, G. (2013). Transport and environmental effects of rail-based park and ride: evidence from the netherlands. *Journal of Transport Geography*, 30:7–16.
- Modig, L., J  rvholm, B., R  nnmark, E., and Nystr  m, L. o. (2006). Vehicle exhaust exposure in an incident case-control study of adult asthma. *European Respiratory Journal*, 28(1):75–81.
- Morello-Frosch, R. and Jesdale, B. M. (2006). Separate and unequal: Residential segregation and estimated cancer risks associated with ambient air toxics in US metropolitan areas. *Environmental Health Perspectives*, pages 386–393.
- Muller, T. and de Leeuw, M. (2006). New method to design traffic control programs. *Transportation Research Record: Journal of the Transportation Research Board*, 1978:68–75.
- Nachtigall, K. (1994). *A branch and cut approach for periodic network programming*. Inst. f  r Mathematik.
- National Research Council (U.S.) (2009). *Urban stormwater management in the United States*. The National Academies Press, Washington D.C.
- National Research Council (U.S.) (2010). *Highway Capacity Manual*. Transportation Research Board, Washington D.C.
- Ou, Q. (2011). *Fusing Heterogeneous Traffic Data: Parsimonious Approaches using Data-Data Consistency*. PhD thesis, Delft University of Technology.
- Padula, A. M., Mortimer, K., Hubbard, A., Lurmann, F., et al. (2012). Exposure to traffic-related air pollution during pregnancy and term low birth weight: Estimation of causal associations in a semiparametric model. *American Journal of Epidemiology*, page kws148.

- Papageorgiou, M., Ben-Akiva, M., Bottom, J., Bovy, P. H., Hoogendoorn, S. P., Hounsell, N. B., Kotsialos, A., and McDonald, M. (2007). ITS and traffic management. In *Handbooks in Operations Research and Management Science*, volume 14, pages 715–774. Elsevier.
- Peters, F. P. M. and Prinsen, L. H. A. (2001). Procesbesturing ccol - flash handleiding. Technical report, Rijkswaterstaat - Adviesdienst verkeer en vervoer.
- Placzek, B. (2014). A self-organizing system for urban traffic control based on predictive interval microscopic model. *Engineering Applications of Artificial Intelligence*, 34:75–84.
- PTV AG (2015). Vissim 8 [computer software]. <http://vision-traffic.ptvgroup.com/nl/products/ptv-vissim/>.
- Radle, A. L. (2007). The effect of noise on wildlife: A literature review. In *World Forum for Acoustic Ecology Online Reader*.
- Rijkswaterstaat WVL (2014). Rws c-regelaar 1.1 [computer software]. <http://www.rwscregelaar.nl/>.
- Rinaldi, M. and Tampère, C. M. (2015). An extended coordinate descent method for distributed anticipatory network traffic control. *Transportation Research Part B: Methodological*, 80:107–131.
- Robertson, D. I. (1969). TRANSYT: A traffic network study tool. Road Research Laboratory report 253, Great Britain Road Research Laboratory.
- Robertson, D. I. and Bretherton, R. D. (1991). Optimizing networks of traffic signals in real time: The scout method. *IEEE Transactions on Vehicular Technology*, vol. 40 no. 1.
- Rosenlund, M., Berglind, N., Pershagen, G., Hallqvist, J., et al. (2006). Long-term exposure to urban air pollution and myocardial infarction. *Epidemiology*, 17(4):383–390.
- Rouphail, N. M. and Radwan, A. E. (1990). Simultaneous optimization of signal settings and left-turn treatments. *Transportation Research Record*, 1287.
- Sacco, N. (2014). Robust optimization of intersection capacity. *Transportation Research Procedia*, 3:1011–1020.
- Salomons, I. (2008). Optimising cycle times of controlled intersections with vrigen. In *Colloquium Vervoersplanologisch Speurwerk*.
- Scholthauer & Wauer (2016). Lisa+ 5.0.4. [software]. <http://www.scholthauer.de/en/software-systems/lisa/>.
- Serafini, P. and Ukovich, W. (1989a). A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581.
- Serafini, P. and Ukovich, W. (1989b). A mathematical model for the fixed-time traffic control problem. *European Journal of Operational Research*, 42(2):152–165.
- Sharples, S., Shalloe, S., Burnett, G., and Crundall, D. (2016). Journey decision making: the influence on drivers of dynamic information presented on variable message signs. *Cognition, Technology & Work*, 18(2):303–317.
- Silcock, J. P. (1997). Designing signal-controlled junctions for group-based operation. *Transportation Research Part A: Policy and Practice*, 31(2):157–173.
- Singh, A., Agrawal, M., et al. (2007). Acid rain and its ecological consequences. *Journal of Environmental Biology*, 29(1):15.
- Sjodin, A., Persson, K., Andreasson, K., Arlander, B., et al. (1998). On-road emission factors derived from measurements in a traffic tunnel. *International Journal of Vehicle Design*, 20(1-4):147–158.
- Sørensen, M., Hvidberg, M., Andersen, Z. J., Nordsborg, R. B., et al. (2011). Road traffic noise and stroke: A prospective cohort study. *European Heart Journal*, 32(6):737–744.
- Stieb, D. M., Chen, L., Eshoul, M., and Judek, S. (2012). Ambient air pollution, birth weight and preterm birth: A systematic review and meta-analysis. *Environmental Research*, 117:100–111.
- Stokols, D., Novaco, R. W., Stokols, J., and Campbell, J. (1978). Traffic congestion, type A behavior, and stress. *Journal of Applied Psychology*, 63(4):467.
- Stolz, C. and Veroude, B. (2013). Nieuwe rekenmethode (voertuigafhankelijke) verkeersregelingen. In *Bijdrage aan het Nationaal verkeerskundecongres*.
- Sutarto, H. (2016). *Adaptive control for traffic signals using a stochastic hybrid system model*. PhD thesis, Ghent University.

- Taale, H. and Hoogendoorn, S. P. (2013). A framework for real-time integrated and anticipatory traffic management. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 449–454. IEEE.
- Tonne, C., Melly, S., Mittleman, M., Coull, B., et al. (2007). A case-control analysis of exposure to traffic and acute myocardial infarction. *Environmental Health Perspectives*, pages 53–57.
- Tully, I. M. (1966). *Synthesis of sequences for traffic signal controllers using techniques of the theory of graphs*. PhD thesis, University of Oxford.
- van den Broek, M. S., van Leeuwen, J. S. H., Adan, I. J. B. F., and Boxma, O. J. (2006). Bounds and approximations for the fixed-cycle traffic-light queue. *Transportation Science*, 40(4):484–496.
- Van Kempen, E. and Babisch, W. (2012). The quantitative relationship between road traffic noise and hypertension: A meta-analysis. *Journal of Hypertension*, 30(6):1075–1086.
- Van Lint, J., Hoogendoorn, S., and van Zuylen, H. J. (2005). Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C: Emerging Technologies*, 13(5):347–369.
- van Zwieten, D. A. J. (2014). *Fluid flow switching servers: Control and observer design*. PhD thesis, Eindhoven University of Technology.
- Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we were going. *Transportation Research Part C: Emerging Technologies*, 43:3–19.
- Webster, F. V. (1958). Traffic signal settings. Road Research Technical Paper 39, Great Britain Road Research Laboratory, Her Majesty’s Stationery Office, London.
- Wickens, C. M. and Wiesensthal, D. L. (2005). State driver stress as a function of occupational stress, traffic congestion, and trait stress susceptibility. *Journal of Applied Biobehavioral Research*, 10(2):83–97.
- Wilhelm, M., Kay Ghosh, J., Su, J., Cockburn, M., et al. (2012). Traffic-related air toxics and term low birth weight in Los Angeles County, California. *Environmental Health Perspectives*, 120(1):132.
- Willson, R. W. (2015). *Parking Management for Smart Growth*. Island Press.
- Wilson, A. (2014). *Handboek verkeerslichtenregelingen 2014*. CROW.
- Wong, C. K. and Heydecker, B. G. (2011). Optimal allocation of turns to lanes at an isolated signal-controlled junction. *Transportation Research Part B: Methodological*, 45(4):667–681.
- Wong, C. K. and Wong, S. C. (2003). Lane-based optimization of signal timings for isolated junctions. *Transportation Research Part B: Methodological*, 37(1):63–84.
- Wong, S. C. (1996). Group-based optimisation of signal timings using the TRANSYT traffic model. *Transportation Research Part B: Methodological*, 30(3):217–244.
- Wu, G., Boriboonsomsin, K., Barth, M., and Tadi, R. (2015). Different types of high-occupancy vehicle access control: Comparative analysis of empirical capacities. *Transportation Research Record: Journal of the Transportation Research Board*, (2484):149–158.
- Wünsch, G. (2008). *Coordination of traffic signals in networks and related graph theoretical problems on spanning trees*. PhD thesis, Universität Berlin.
- Yan, C., Jiang, H., and Xie, S. (2014). Capacity optimization of an isolated intersection under the phase swap sorting strategy. *Transportation Research Part B: Methodological*, 60:85–106.
- Zaidi, A. A., Kulcsár, B., and Wymeersch, H. (2016). Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):2134–2143.
- Zhou, Z. and Cai, M. (2014). Intersection signal control multi-objective optimization based on genetic algorithm. *Journal of Traffic and Transportation Engineering (English Edition)*, 1(2):153–158.
- Zografos, K. G., Androustopoulos, K. N., and Spiliadakis, V. (2009). Design and assessment of an online passenger information system for integrated multimodal trip planning. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):311–323.

Appendices of Chapter 1

A.1 Automatic generation of a single-ringed phase diagram

A type of traffic light control that is often used in practice is actuated control. Such an actuated controller uses information on the current traffic situation to control the different signal groups. Due to the high computational complexity, it is impossible for the actuated controller to oversee (in real-time) all possible orders in which these signal groups can be served. Therefore, it is common to serve the signal groups in a predetermined order. The order in which the signal groups are served is then specified in a *phase diagram*.

In this appendix we show how to automatically generate a single-ringed phase diagram from a signal group diagram. To this end, we propose an optimization problem (linear programming problem) that is able to find such a single-ringed phase diagram. Before reading this appendix, we strongly recommend to read Chapter 4 of this thesis. In that chapter we formulate an optimization problem to optimize signal group diagrams. The optimization problem proposed in this appendix strongly resembles that optimization problem. Furthermore, we adopt all notation that we introduce in that chapter of this thesis.

Consider a signal group diagram from which we want to find a single-ringed phase diagram. Let K_i be the number of realizations (distinct periodically repeating green intervals) that signal group $i \in \mathcal{S}$ has for this signal group diagram. Number the realizations of each signal group $i \in \mathcal{S}$ according to the periodic order in which they occur, i.e., the realizations of signal group $i \in \mathcal{S}$ are scheduled in the periodically repeating order $1, 2, \dots, K_i$. Moreover, define $\mathcal{K}_i := \{1, \dots, K_i\}$ to be the set of realizations of signal group i . We propose to find a single-ringed phase diagram for this signal group diagram by solving two linear programming problems. The first linear programming problem finds the minimum number of phases required by any single-ringed phase diagram. Thereupon, the second linear programming problem fixes the number of phases to this minimum number of phases and attempts to include each realization in as many phases as

possible. In Section A.1.1 we formulate the linear programming problem. Subsequently, in Section A.1.2 we use this optimization problem to find a signal group diagram. Finally, in Section A.1.3 we apply the proposed method to some examples.

A.1.1 Formulating an optimization problem

In this section we formulate an optimization problem to find a single-ringed phase diagram. First, we introduce the design variables of this optimization problem. Subsequently, we introduce all constraints. Finally, we prove that this linear programming problem finds a single-ringed phase diagram when we fix the number of phases to an integral value.

Design variables

Let P be the number of phases of the single-ringed phase diagram. Each realization k of signal group $i \in \mathcal{S}$ is served by an uninterrupted sequence of phases, e.g., the phases $P-1, P, 1, 2$. Let $p(\textcircled{i}_k) \in \{1, \dots, P\}$ denote the first phase in this sequence and let $p(\textcircled{i}_k) \in \{1, \dots, P\}$ be the last phase in this sequence. Note that \textcircled{i}_k (\textcircled{i}_k) is associated with a switch to green (red) of signal group $i \in \mathcal{S}$, i.e., signal group $i \in \mathcal{S}$ switches to green at the start of phase $p(\textcircled{i}_k)$ and it switches to red at the end of phase $p(\textcircled{i}_k)$.

In this chapter we use the following terminology. Let realization k of signal group $i \in \mathcal{S}$ be served during the phases $P-1, P, 1, 2$. We then say that (realization k of) signal group i is included in the phases $P-1, P, 1, 2$. Moreover, we may say that the event \textcircled{i}_k (\textcircled{i}_k) is included in phase $P-1$ (phase 2). A single-ringed phase diagram is completely specified by the number of phases P , the values for $p(\textcircled{i}_k)$, $i \in \mathcal{S}$, $k \in \mathcal{K}_i$, and the values for $p(\textcircled{i}_k)$, $i \in \mathcal{S}$, $k \in \mathcal{K}_i$. However, we do not optimize these variables directly. We introduce the real-valued design variables below. Define the following set:

$$\mathcal{E} = \{\textcircled{i}_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\} \cup \{\textcircled{i}_k \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}.$$

Define $\chi_P(\varepsilon_1, \varepsilon_2)$, $\varepsilon_1, \varepsilon_2 \in \mathcal{E}$ to be the difference (in number of phases) between the phase that includes the event ε_1 and the phase that includes the event ε_2 , i.e.,

$$\chi_P(\varepsilon_1, \varepsilon_2) := p(\varepsilon_2) - p(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)P. \quad (\text{A.1})$$

where $z(\varepsilon_1, \varepsilon_2)$ is some integer. Define $\gamma_P(\varepsilon_1, \varepsilon_2) := \chi_P(\varepsilon_1, \varepsilon_2)/P$, which is the difference between a phase that includes the event ε_1 and a phase that includes the event ε_2 expressed as a fraction of the number of phases P . The real-valued design variables of the optimization problem are the reciprocal $\mathbf{P}' := 1/P$ and the fractions $\gamma_P(\varepsilon_1, \varepsilon_2)$ that are subject to a constraint. The fractions that are subject to a constraint are the ones that correspond to an arc of the following constraint graph $G = (V, A)$:

$$V = \mathcal{E},$$

$$A = A_g \cup A_r \cup A_c,$$

where,

$$A_g = \{(\textcircled{i}_k, \textcircled{i}_k) : i \in \mathcal{S}, k \in \mathcal{K}_i\},$$

$$A_r = \{(\textcircled{i}_{k-1}, \textcircled{i}_k) : i \in \mathcal{S}, k \in \mathcal{K}_i\},$$

$$A_c = \{(\textcircled{i}_k, \textcircled{j}_{k'}) : \{(i, k), (j, k')\} \in \Psi_R\},$$

$$\Psi_R = \{\{(i, k), (j, k')\} \mid \{i, j\} \in \Psi_S, k \in \mathcal{K}_i, k' \in \mathcal{K}_j\},$$

$$\textcircled{i}_0 := \textcircled{i}_{K_i}.$$

In other words $\gamma_P(\varepsilon_1, \varepsilon_2)$ is subject to a constraint if $(\varepsilon_1, \varepsilon_2) \in A$. Note that this constraint graph equals the one used in Chapter 4 when $\underline{K}_i = \overline{K}_i = K_i$ for all signal groups $i \in \mathcal{S}$. From these real-valued design variables \mathbf{P}' and $\gamma_P(\varepsilon_1, \varepsilon_2)$ we can obtain P and $p(\varepsilon)$, $\varepsilon \in \mathcal{E}$ analogously to the iteration scheme given Section 3.2.1; with that iteration scheme we can find a signal group diagram from the real-valued design variables \mathbf{T}' and $\gamma(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$.

Resemblance with Chapter 4 This appendix has a lot in common with the optimization framework proposed in Chapter 4. In that chapter, each element $\varepsilon \in \mathcal{E}$ can be interpreted as either a switch to effective green (\textcircled{i}_k) or a switch to effective red (\textcircled{i}_k). A signal group diagram is specified by its period duration T and the times $t(\varepsilon)$ at which these events $\varepsilon \in \mathcal{E}$ are scheduled, whereas a phase diagram is defined by P and $p(\varepsilon)$, $\varepsilon \in \mathcal{E}$. Furthermore, in Chapter 4 we optimize the reciprocal of the period duration \mathbf{T}' and the fractions $\gamma(\varepsilon_1, \varepsilon_2)$, whereas, in this appendix, we optimize the reciprocal \mathbf{P}' and the fractions $\gamma_P(\varepsilon_1, \varepsilon_2)$.

Linear constraints

Cycle periodicity constraints We start with the most difficult constraints, which are the *cycle periodicity constraints*. For more detailed information on these constraints we refer to Section 3.3 of this thesis. In that section we have used these constraints to model the periodicity of a signal group diagram. In Chapter 4, these cycle periodicity constraints were formulated for each cycle \mathcal{C} in an integral cycle basis \mathcal{B} of the constraint graph G , i.e., in that section we had the following constraints:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2) = z_{\mathcal{C}}, \quad \mathcal{C} \in \mathcal{B},$$

where z_C is an integral-valued design variable and where \mathcal{C}^+ respectively \mathcal{C}^- is the set of arcs that the cycle \mathcal{C} traverses in the forward direction (from tail to head) and the set of arcs that the cycle \mathcal{C} traverses in the backward direction (from head to tail). The values for z_C , $\mathcal{C} \in \mathcal{B}$ specify the order in which the signal group diagrams receive their green intervals.

In the same manner, these constraints can be used to model the periodicity of a phase diagram. For each cycle \mathcal{C} in some integral cycle basis \mathcal{B} of constraint graph G we have the following constraint:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma_P(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma_P(\varepsilon_1, \varepsilon_2) = z_C, \quad \mathcal{C} \in \mathcal{B}. \quad (\text{A.2a})$$

We would like the phase diagram to resemble the signal group diagram, i.e., it should visualize the order in which the signal groups receive their green intervals for this signal group diagram. Therefore, we fix the value for z_C , $\mathcal{C} \in \mathcal{B}$ to the value associated with the signal group diagram for which we try to find a phase diagram.

Remark A.1. *It is sufficient to formulate the cycle periodicity constraints for an integral cycle basis \mathcal{B} . This cycle periodicity constraint is then automatically satisfied for all cycles in the constraint graph G . Any signal group diagram satisfies the cycle periodicity constraints (4.8o)–(4.8p). This translates to the following equalities being satisfied for our situation:*

$$\gamma_P(\textcircled{i}_k, \textcircled{i}_k) + \gamma_P(\textcircled{i}_k, \textcircled{j}_{k'}) + \gamma_P(\textcircled{j}_{k'}, \textcircled{j}_{k'}) + \gamma_P(\textcircled{j}_{k'}, \textcircled{i}_k) = 1, \quad (\text{A.2b})$$

and

$$\sum_{k \in \mathcal{K}_i} (\gamma_P(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma_P(\textcircled{i}_k, \textcircled{i}_k)) = 1. \quad (\text{A.2c})$$

Equation (A.2b) indicates that a phase diagram is comprised of the phases that include realization k of signal group i , the phases that include realization k' of signal group j plus the phases that include neither of these realizations. Equation (A.2c) indicates that a phase diagram is comprised of the phases that include the realizations of signal group $i \in \mathcal{S}$ plus the phases that do not include any realization of signal group i .

Other linear constraints The number of phases must be strictly negative, i.e.,

$$1/\mathbf{P}' \geq \delta. \quad (\text{A.2d})$$

where δ is some small positive number. Between each two realizations of signal group $i \in \mathcal{S}$, this signal group should not be included in at least one phase, i.e., the number of phases $\chi_P(\textcircled{i}_{k-1}, \textcircled{i}_k)$ between the last phase that includes realization $k-1$ of signal group $i \in \mathcal{S}$ and the first phase that includes realization k of signal group i should be at least two. We can formulate this constraint as follows:

$$\gamma_P(\textcircled{i}_{k-1}, \textcircled{i}_k) \geq 2\mathbf{P}', \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i. \quad (\text{A.2e})$$

where $\textcircled{i}_0 := \textcircled{i}_{K_i}$. The number of phases that include realization $k \in \mathcal{K}_i$ of signal group $i \in \mathcal{S}$ is given by $\chi_P(\textcircled{i}_k, \textcircled{i}_k) + 1$. This number of phases should be at least one. Therefore, we have:

$$\gamma_P(\textcircled{i}_k, \textcircled{i}_k) \geq 0, \quad i \in \mathcal{S}, \quad k \in \mathcal{K}_i. \quad (\text{A.2f})$$

The realizations of two conflicting signal groups should not be included in the same phase. Thus, the number of phases $\chi_P(\textcircled{i}_k, \textcircled{j}_{k'})$ between the last phase that includes realization k of signal group $i \in \mathcal{S}$ and the first phase that includes realization k' of conflicting signal group j should be at least one. This can be modeled with the following constraints:

$$\gamma_P(\textcircled{i}_k, \textcircled{j}_{k'}) \geq \mathbf{P}', \quad \{(i, k), (j, k')\} \in \Psi_R. \quad (\text{A.2g})$$

The circuitual constraint (A.2b), which is implied by the cycle periodicity constraints (A.2a), together with the constraints (A.2g) ensure that the realizations of two conflicting signal groups are not included in the same phase.

Well-posedness

In this section we prove that each variable $\gamma_P(\varepsilon_1, \varepsilon_2)$ is defined unambiguously. To this end, for each real-valued design variable we either prove the inclusion $\gamma_P(\varepsilon_1, \varepsilon_2) \in [0, 1)$ or we prove the inclusion $\gamma_P(\varepsilon_1, \varepsilon_2) \in (0, 1]$. Each real-valued design variable $\gamma_P(\textcircled{i}_k, \textcircled{i}_k)$, $i \in \mathcal{S}$, $k \in \mathcal{K}_i$ is included in the interval $[0, 1)$. The lower bound of zero follows from (A.2f). The upper bound follows from circuitual constraint (A.2c) combined with the strict positivity of the number of phases (A.2d) and the lower bounds (A.2e)–(A.2f). Each real-valued design variable $\gamma_P(\textcircled{i}_k, \textcircled{j}_{k'})$, $\{(i, k), (j, k')\} \in \Psi_R$ is included in the interval $[0, 1)$, which follows from circuitual constraint (A.2b) combined with the strict positivity of the number of phases (A.2d) and the lower bounds (A.2f)–(A.2g). For each real-valued design variable $\gamma_P(\textcircled{i}_{k-1}, \textcircled{i}_k)$ we have the inclusion $\gamma_P(\textcircled{i}_{k-1}, \textcircled{i}_k) \in (0, 1]$. Its lower bound follows from the strictly positive lower bound (A.2e). The upper bound follows from circuitual constraint (A.2c) combined with the strict positivity of the number of phases (A.2d), lower bound (A.2e) and lower bound (A.2f).

Integral solution

In this section we prove that we can obtain a single-ringed phase diagram by fixing the reciprocal \mathbf{P}' to a value such that $P := 1/\mathbf{P}'$ is integral and minimizing some linear objective (in γ_P) subject to the constraints (A.2) with a simplex algorithm; a simplex algorithm is a commonly used algorithm to solve linear programming problems. Before reading this section, we strongly recommend to read Chapter 5 of this thesis.

Denote a solution to the linear constraints (A.2) by (\mathbf{P}', γ_P) , where γ_P is a vector that contains the values for $\gamma_P(\varepsilon_1, \varepsilon_2)$. Each solution (\mathbf{P}', γ_P) for which both $P := 1/\mathbf{P}'$ and $\chi_P := \gamma_P/\mathbf{P}'$ are integral corresponds to a single-ringed phase diagram; we can obtain this

phase diagram (from \mathbf{P}' and γ_P) analogously to the method given in Section 3.2.1. The following lemma states that when we set the reciprocal \mathbf{P}' to a value for which $P := 1/\mathbf{P}'$ is integral, then χ_P is integral for every vertex of the polyhedron that is spanned by the resulting linear constraints (A.2). This implies that each of these vertices corresponds to a single-ringed phase diagram.

Lemma A.1. *Set the reciprocal \mathbf{P}' to a value for which $P := 1/\mathbf{P}'$ is integral and the linear programming problem (A.2) is feasible. Consider the polyhedron that is spanned by the resulting linear constraints (A.2). The elements of $\chi_P := \gamma_P/\mathbf{P}'$ are all integral for each vertex of this polyhedron.*

Proof. Define the polyhedron F to be the polyhedron that is spanned by the linear constraints (A.2). Furthermore, define the polyhedron H as the polyhedron that is obtained from F when we scale each vector in F with a factor $1/\mathbf{P}'$, i.e.,

$$H := \{\gamma_P/\mathbf{P}' \mid \gamma_P \in F\}.$$

Note that each vertex of F maps to a vertex of H , i.e., $\gamma_P/\mathbf{P}' \in H$ is a vertex of H if and only if γ_P is a vertex of F . For ease of notation we use $\chi := \chi_P$ in this proof. We prove that the polyhedron H has integral vertices, which would prove the lemma. We can write the polyhedron H as follows:

$$H := \{\chi \mid M\chi = b, \underline{\chi} \leq \chi \leq \bar{\chi}\},$$

where M is a cycle-arc incidence matrix associated with some integral cycle basis of the constraint graph G , and the vectors b , $\underline{\chi}$, and $\bar{\chi}$ are all integral. The cycle periodicity constraints (A.2a) can be written as $M\chi = b$, where the vector b is comprised of the integral values z_C/\mathbf{P}' , $C \in \mathcal{B}$. Furthermore, the constraints (A.2e)–(A.2g) can be written as $\underline{\chi} \leq \chi$ where each element of $\underline{\chi}$ is integral. We can w.l.o.g. include a (redundant) upper bound of $1/\mathbf{P}' \in \mathbb{Z}$ on each of these variables in χ , which proves that the polyhedron H indeed can be written in the proposed form.

We prove that H has integral vertices. For each graph, a totally unimodular cycle basis $\mathcal{B}' = \{C'_1, \dots, C'_d\}$ exists. Let M' be the cycle-arc incidence matrix associated with such a totally unimodular cycle basis of constraint graph G . Furthermore, let $\mathcal{B} = \{C_1, \dots, C_d\}$ be the integral cycle basis used to formulate the cycle periodicity constraints (A.2a). Since cycle basis \mathcal{B} is integral, for each cycle C'_i , $i = 1, \dots, d$ we can find integral values $\alpha_{i,j}$, $j = 1, \dots, d$ such that its cycle-arc incidence vector C'_i can be written as the following linear combination:

$$C'_i = \sum_{j=1}^d \alpha_{i,j} C_j.$$

Therefore, we can write:

$$M' = UM, \quad \text{where,} \quad U = [\alpha_{i,j}]_{i,j=1,\dots,d} \in \mathbb{Z}^{d \times d}.$$

Hence, we can rewrite polyhedron H to $H := \{\chi \mid UM\chi = Ub, \underline{\chi} \leq \chi \leq \bar{\chi}\} = \{\chi \mid M'\chi \leq b', \underline{\chi} \leq \chi \leq \bar{\chi}\}$, where b' is an integral vector and M' is totally unimodular. From Theorem 5.3 it follows that the polyhedron H has integral vertices, which concludes this lemma. \square

Lemma A.1 states that when we set the reciprocal \mathbf{P}' to a value for which $P := 1/\mathbf{P}'$ is integral, each vertex of the polyhedron spanned by the resulting linear constraints (A.2) corresponds to a single-ringed phase diagram. A simplex algorithm searches only the vertices of a polyhedron to find the optimal solution. Therefore, the following corollary is a direct consequence of Lemma A.1.

Corollary A.2. *Set the reciprocal \mathbf{P}' to a value for which $P := 1/\mathbf{P}'$ is integral. If a solution exists to the resulting linear constraints (A.2), a single-ringed phase diagram exists with P phases. Moreover, such a single-ringed phase diagram can be obtained with a simplex algorithm by minimizing any linear objective function (in γ_P) subject to the constraints (A.2).*

A.1.2 Obtaining the single-ringed phase diagram

In this section we use the formulated linear constraints to find a single-ringed phase diagram by solving two linear programming problems. The first linear programming problem finds the minimum number of phases that is required. The second linear programming problem finds the single-ringed phase diagram that includes each realization in as many phases as possible. At the end of this section we prove that this approach returns a single-ringed phase diagram for any signal group diagram satisfying the constraints of MIP problem (4.8). Moreover, in Section A.1.3 we apply this method to some examples.

Step one

When minimizing the number of phases that is required, we consider the following objective function:

$$\underset{\mathbf{P}', \gamma_P}{\text{maximize}} \mathbf{P}'. \quad (\text{A.3})$$

Solving the MILP problem with linear constraints (A.2) and objective (A.3) does not necessarily return an integral number of phases $1/\mathbf{P}'$. However, the following lemma states that $\lceil 1/\mathbf{P}' \rceil$ is then the smallest number of phases for which a single-ringed phase diagram exists that satisfies (A.2):

Lemma A.3. *Let (\mathbf{P}', γ_P) be the solution that minimizes (A.3) subject to the linear constraints (A.2). Then $P := \lceil 1/\mathbf{P}' \rceil$ is the smallest number of phases for which a single-ringed phase diagram exists that satisfies (A.2).*

Proof. The minimum number of phases required by the single-ringed phase diagram is integral. Therefore, the linear programming problem gave a lower bound of $P := \lceil 1/\mathbf{P}' \rceil$ on this minimum number of phases. We prove that a single-ringed phase diagram exists with P phases, which proves the lemma. Consider the solution $(\alpha \mathbf{P}', \alpha \gamma_P)$ with $\alpha = \frac{1}{\mathbf{P}' \lceil 1/\mathbf{P}' \rceil}$. It holds that the number of phases equals $P := \frac{1}{\alpha \mathbf{P}'}$ for this solution. Furthermore, this solution satisfies the linear constraints (A.2). Therefore, from Corollary A.2 it follows that a single-ringed phase diagram exists with P phases. \square

Step two

In the first step, we have obtained the minimum number of phases P required by any single-ringed phase diagram. During the second step, we fix the reciprocal \mathbf{P}' to $1/P$ and maximize the average weighted number of signal groups that is included in each phase, i.e., the objective is:

$$\underset{\mathbf{P}', \gamma_P}{\text{maximize}} \quad \frac{1}{P} \sum_{i \in \mathcal{S}} \sum_{k \in \mathcal{K}_i} w_i \gamma(\textcircled{i}_k, \textcircled{i}_k),$$

where w_i is the weight (importance) of signal group $i \in \mathcal{S}$. From Corollary A.2 it follows that solving this linear programming problem with a simplex algorithm results in a single-ringed phase diagram. That is as long as this linear programming problem is feasible. In the following lemma we prove that the proposed two-step approach finds a single-ringed phase diagram.

Lemma A.4. *Consider some signal group diagram that satisfies all the linear constraints of MIP problem (4.8). For such a signal group diagram, the two-step approach proposed in this appendix finds a single-ringed phase diagram.*

Proof. Let $(\mathbf{T}', \gamma, \mathbf{z}, \mathbf{b})$ be the solution to the MIP problem (4.8) associated with the signal group diagram for which we want to find a single-ringed phase diagram. We construct a solution (\mathbf{P}', γ_P) that satisfies the linear constraints (A.2). In the first step of the two-step approach we then find the smallest number of phases P for which a single-ringed phased diagram exists satisfying (A.2); this follows from Lemma A.3. From Corollary A.2 it follows that we obtain a single-ringed phase diagram during the second step.

Recall that well-posedness constraints (4.8q)–(4.8r) force the following inequality for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$:

$$\gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{j}_{k'}, \textcircled{j}_{k'}) \geq \epsilon \mathbf{T}'.$$

Define $\zeta := \min\{\epsilon, 0.5 \min_{i \in \mathcal{S}} \{r_i + g_i\}\} > 0$. We prove that the following solution satisfies all linear constraints (A.2):

$$\begin{aligned}
\mathbf{P}' &:= \zeta \mathbf{T}', \\
\gamma_P(\textcircled{i}_k, \textcircled{i}_k) &:= 0, & i \in \mathcal{S}, \quad k \in \mathcal{K}_i, \\
\gamma_P(\textcircled{i}_{k-1}, \textcircled{i}_k) &:= \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k), & i \in \mathcal{S}, \quad k \in \mathcal{K}_i, \\
\gamma_P(\textcircled{i}_k, \textcircled{j}_{k'}) &:= \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) + \gamma(\textcircled{j}_{k'}, \textcircled{j}_{k'}), & \{(i, k), (j, k')\} \in \Psi_R.
\end{aligned}$$

First, we prove that the solution (\mathbf{P}', γ_P) satisfies all cycle periodicity constraints. Without loss of generality assume that the cycle periodicity constraints (A.2a) are formulated with an integral cycle basis \mathcal{B} that is constructed with Theorem 4.1 from (Fleuren, 2016). This integral cycle basis has the property that its cycles only traverse arcs in the forward direction. Furthermore, each such cycle alternates between a forward arc $(\textcircled{i}_k, \textcircled{i}_k)$, $i \in \mathcal{S}$ and either a forward arc $(\textcircled{i}_{k-1}, \textcircled{i}_k)$, $i \in \mathcal{S}$ or a forward arc $(\textcircled{i}_k, \textcircled{j}_{k'})$, $\{(i, k), (j, k')\} \in \Psi_R$. Therefore, from the following equalities

$$\begin{aligned}
\gamma_P(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma_P(\textcircled{i}_k, \textcircled{i}_k) &= \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k), \\
\gamma_P(\textcircled{i}_k, \textcircled{j}_{k'}) + \gamma_P(\textcircled{i}_k, \textcircled{i}_k) &= \gamma(\textcircled{i}_k, \textcircled{j}_{k'}) + \gamma(\textcircled{i}_k, \textcircled{i}_k),
\end{aligned}$$

and the fact that the solution $(\mathbf{T}', \gamma, \mathbf{z}, \mathbf{b})$ satisfies all cycle periodicity constraints, it follows that the proposed solution (\mathbf{P}', γ_P) also satisfies the cycle periodicity constraints. We prove that the proposed solution also satisfies the constraints (A.2d)–(A.2g). The inequality (A.2d) follows from the strict positivity of \mathbf{T}' and ζ . From (4.8c)–(4.8f) it follows that:

$$\begin{aligned}
\gamma_P(\textcircled{i}_{k-1}, \textcircled{i}_k) &:= \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{i}_k), \\
&\geq (\underline{r}_i + \underline{g}_i) \mathbf{T}' \geq (\underline{r}_i + \underline{g}_i) \mathbf{P}' / \zeta \geq 2\mathbf{P}',
\end{aligned}$$

which implies that (A.2e) is satisfied. The inequality on (A.2f) is trivially satisfied by the proposed solution. Furthermore, from the well-posedness constraints (4.8q)–(4.8r) it follows that:

$$\begin{aligned}
\gamma_P(\textcircled{i}_k, \textcircled{j}_{k'}) &:= \gamma(\textcircled{i}_k, \textcircled{i}_k) + \gamma(\textcircled{i}_k, \textcircled{j}_{k'}), \\
&\geq \epsilon \mathbf{T}' \geq \epsilon / \zeta \mathbf{P}' \geq \mathbf{P}',
\end{aligned}$$

which proves that (A.2g) is satisfied for the proposed solution. This concludes this proof. \square

A.1.3 Examples

In this section we apply the proposed method to two intersections. First, we consider a small intersection with 5 signal groups. Thereupon, we consider the intersection L1 from (Fleuren and Lefeber, 2016a).

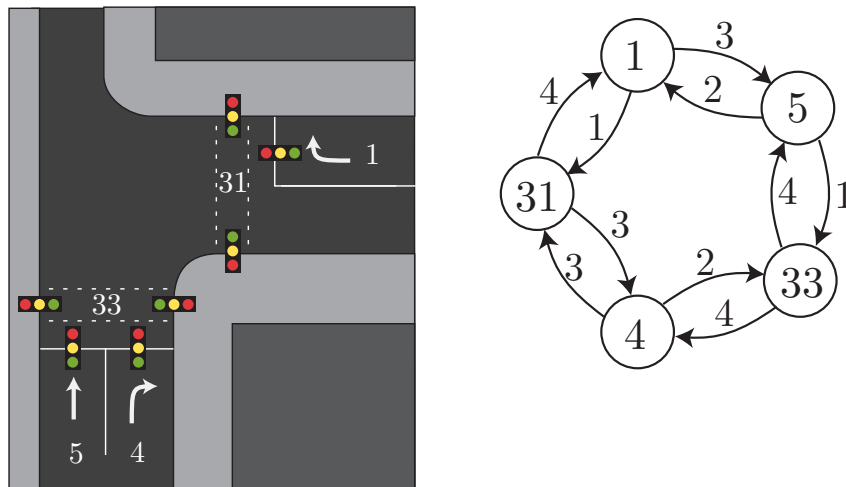
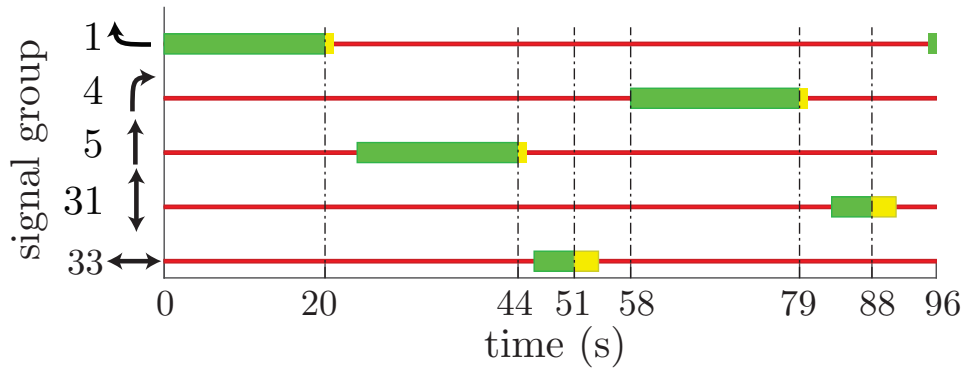


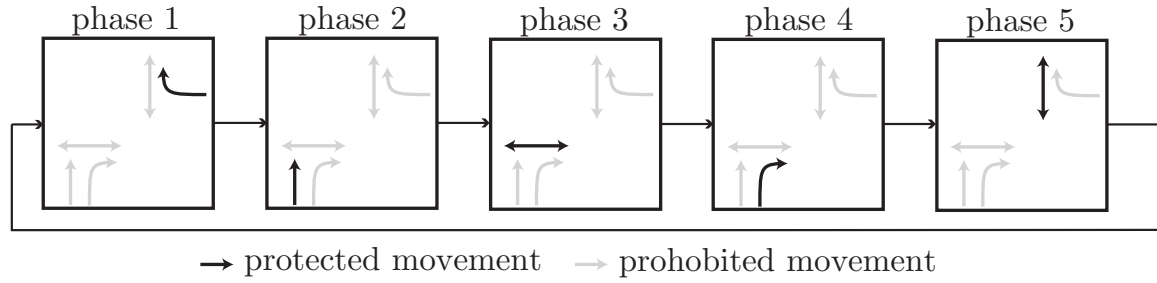
Figure A.1: A T-junction (left) and a graph (right) that visualizes the conflicts. Each two signal groups that are conflicting are connected with a (directed) arc. The value attached to the arc (i, j) , is the minimum amount of time (in seconds) that signal group i has to be red for, before signal group j may switch to a green indication.

A small example

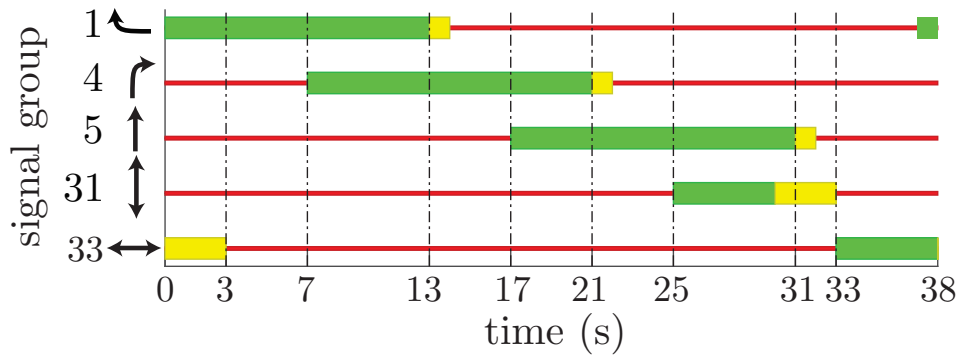
In this section we consider a small intersection. This intersection is visualized in Figure A.1. In Figure A.2 we have visualized four different signal group diagrams plus their associated phase diagrams. For some of these phase diagrams it might seem that some phases are missing. For example, for the signal group diagram in Figure A.2c, the signal groups 1 and 4 are simultaneously green. Whereas its phase diagram, which is visualized in Figure A.2d, has no phase that includes both these signal groups. Nevertheless, the phase diagram is correct. An actuated controller serves the signal groups in the order that is specified by the phase diagram. However, some 'transition' between two subsequent phases is allowed. For example, for the phase diagram in Figure A.2d, the actuated controller can end the green interval of signal group 33 (in phase 1). As a consequence, signal group 4 (in phase 2) can already become green as it has no conflict with signal group 1. Signal group 1 and signal group 4 are then green simultaneously.



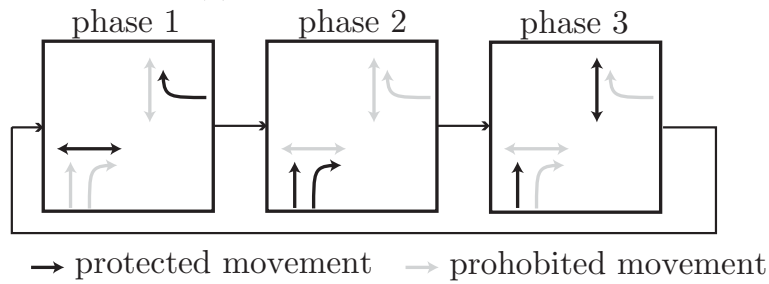
(a) Signal group diagram for the small example of Figure A.1.



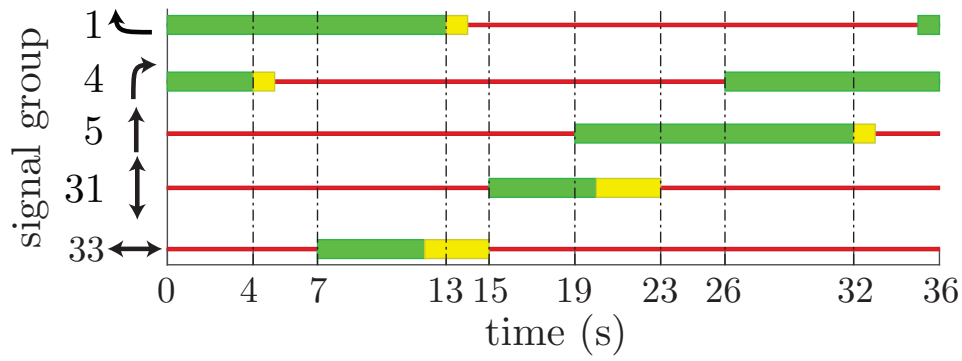
(b) Phase diagram that we have obtained for the signal group diagram in Figure A.2a.



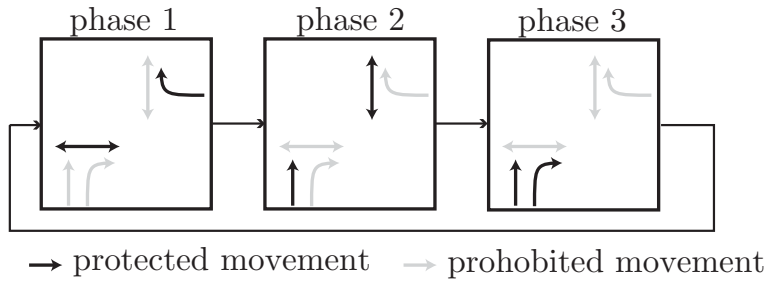
(c) Signal group diagram for the small example of Figure A.1. .



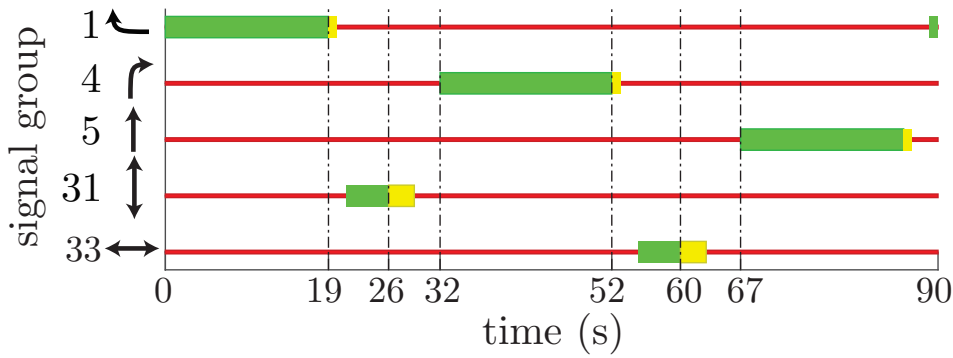
(d) Phase diagram that we have obtained for the signal group diagram in Figure A.2c.



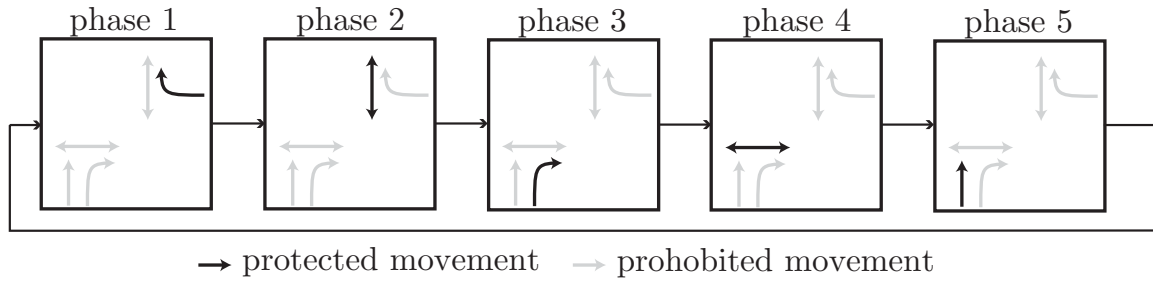
(e) Signal group diagram for the small example of Figure A.1.



(f) Phase diagram that we have obtained for the signal group diagram in Figure A.2e.



(g) Signal group diagram for the small example of Figure A.1.



(h) Phase diagram that we have obtained for the signal group diagram in Figure A.2g.

Figure A.2: Signal group diagrams and their computed single-ringed phase diagrams.

A large example

In this section we consider a larger example: the intersection L1 from (Fleuren and Lefebber, 2016a). This example has 27 signal groups, which includes 8 signal groups for pedestrians and 8 signal groups for cyclists. In Figure A.3 we give a graphical representation of this intersection and in Figure A.4 we give a possible signal group diagram. Finally, in Figure A.5 we give the phase diagram that we have obtained for this signal group diagram. With this large example we conclude this appendix about the automatic generation of a single-ringed phase diagram.

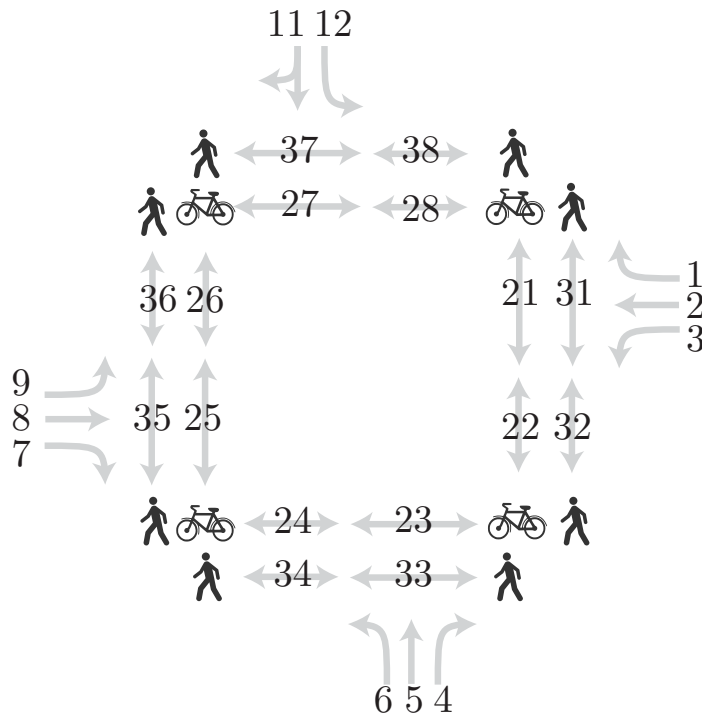


Figure A.3: Graphical representation of the intersection L1 from (Fleuren and Lefebber, 2016a).

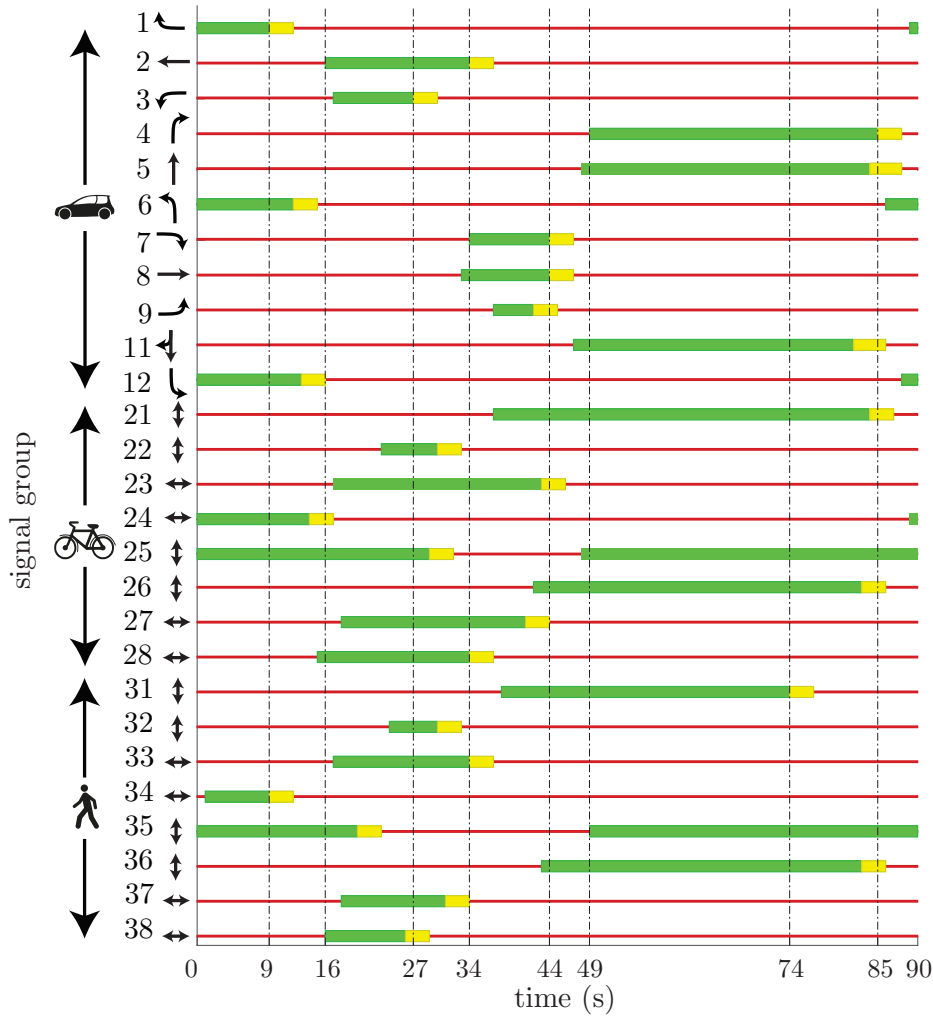


Figure A.4: A signal group diagram for the intersection L1 from (Fleuren and Lefeber, 2016a).

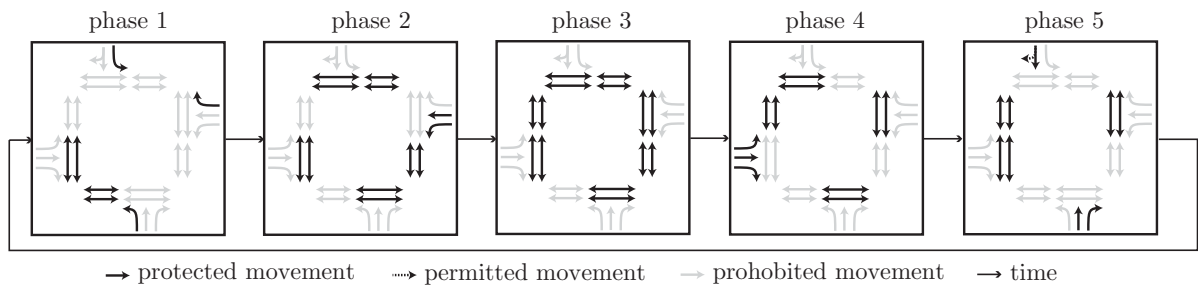


Figure A.5: The phase diagram that is obtained for the signal group diagram in Figure A.4.

A.2 Actuated control may behave as a pre-timed control

In this appendix we show that an actuated controller may behave as a pre-timed controller in some situations. To this end we consider a simple parametric example with two conflicting signal groups; we have visualized this example in Figure A.6. Both signal groups of this intersection control the access to the intersection for a single line of vehicles

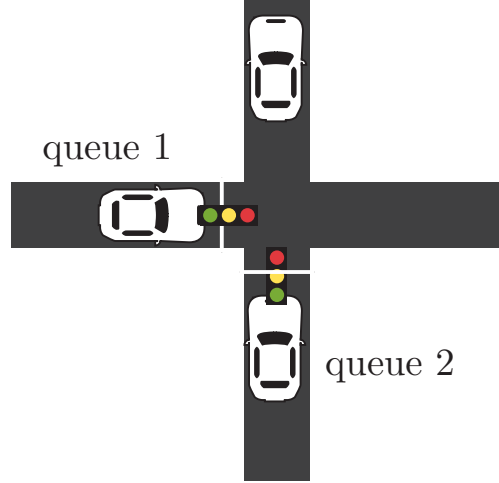


Figure A.6: An example intersection with two conflicting signal groups.

(queue); we refer to these queues as queue 1 and queue 2.

A.2.1 A mathematical model

A traffic light has three indications: green, yellow, and red. However, as we motivate in Chapter 2, mathematically it suffices to have two modes: effective green and effective red. Vehicles depart from the queue whenever its traffic light is effective green and vehicles stop when the traffic light is effective red. Let the departure process at queue q be as follows. During an effective green interval, a vehicle departs every $1/\mu_q$ seconds; we refer to μ_q as the *saturation flow rate* of queue q . Consider an infinite time horizon $[0, \infty)$ and divide this time horizon into constant time intervals of $1/\mu_q$ seconds which we call *slots*. Let $A_{q,k}$ denote the number of arrivals at queue q during slot $k \geq 1$. We assume that $A_{q,k}$ follows some stochastic (or deterministic) distribution; these distributions may differ between slots. Define the average arrival rate at queue q to be ρ_q vehicles per slot, i.e.,

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K E(A_{q,k}) = \rho_q, \quad (\text{A.4})$$

where $E(X)$ is the expected value of X . If queue q is effective green during slot $k \geq 1$, a vehicle departs at the end of this slot (if this queue is not empty). Define $D_{q,k}$ to equal 1 if queue q is effective green during slot k and zero otherwise. The number of vehicles $x_{q,k}$ in queue q at the end of slot k follows the following recursive relation:

$$x_{q,k+1} = \max\{x_{q,k} + A_{q,k+1} - D_{q,k+1}, 0\}, \quad k = 0, 1, 2, \dots \quad (\text{A.5})$$

where $x_{q,0}$ is the initial number of vehicles in queue q .

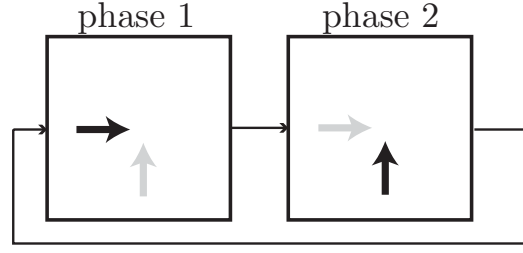


Figure A.7: The phase diagram associated with the actuated controller of the intersection in Figure A.6

A.2.2 The actuated control strategy

The signal groups are controlled with an actuated controller; this controller is based on the phase diagram of Figure A.7. The actuated controller alternates between an effective green interval of signal group 1 and an effective green interval of signal group 2. The actuated controller may end an effective green interval at the end of each slot. If the actuated controller decides to end the effective green interval of queue 1 (2), queue 2 (1) becomes effective green $\underline{c}_{1,2} \geq 0$ ($\underline{c}_{2,1} \geq 0$) slots after signal group 2 (1) became effective red. This time between a signal group becoming effective red and a conflicting signal group becoming effective green is called a *clearance time*; such a clearance time ensures that each traffic stream can safely cross the intersection without encountering conflicting traffic streams. The actuated controller makes its decision to end a green interval as follows. Let queue $q = 1, 2$ be effective green during slot k . The actuated controller ends the effective green interval of queue q at the end of slot k only if at least one of the following two statements is true:

- the queue is empty ($x_{q,k} = 0$) and queue q has been effective green for the minimum duration of \underline{g}_q slots.
- queue q has been effective green for the maximum duration of \bar{g}_q slots.

A.2.3 Assumption on the parameters

For the small parametric example with two signal groups, we assume the following:

Assumption A.1. *The average number of vehicles that arrive at queue 1 (queue 2) during a period of $\bar{g}_1 + \bar{g}_2 + \underline{c}_{1,2} + \underline{c}_{2,1}$ slots exceeds \bar{g}_1 (\bar{g}_2), which is the maximum number of vehicles that can depart during an effective green interval of signal group 1 (signal group 2), i.e.,*

$$\begin{aligned}\bar{g}_1 &< \rho_1(\bar{g}_1 + \bar{g}_2 + \underline{c}_{1,2} + \underline{c}_{2,1}), \\ \bar{g}_2 &< \rho_2(\bar{g}_1 + \bar{g}_2 + \underline{c}_{1,2} + \underline{c}_{2,1}).\end{aligned}$$

A.2.4 The actuated controller behaves as a pre-timed controller

We prove that the introduced actuated controller behaves as a pre-timed controller. To prove this, we require the following definition of an *unbounded queue length*.

Definition A.1 (Unbounded queue length). *A queue length $x(t)$ is unbounded if for each $X \in \mathbb{R}$ with probability one some finite time t^* exists such that $x(t) \geq X$ for all $t \geq t^*$.*

Lemma A.5. *Consider a queue q with a saturation flow rate of μ_q . Let ρ_q be the average number of cars that arrive during a slot of $1/\mu_q$ seconds. Furthermore, let queue q be effective green for a fraction less than ρ_q , i.e.,*

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K E(D_{q,k}) = f_q < \rho_q, \quad (\text{A.6})$$

The queue length $x_{q,k}$ is then unbounded.

Proof. We prove that for all $X \in \mathbb{R}$ it holds that:

$$\lim_{k \rightarrow \infty} P(x_{q,k} \geq X) = 1.$$

This limit would imply the unbounded queue length. Define $A_{q,k}^{\text{cum}} = \sum_{l=1}^k A_{q,l}$ and $D_{q,k}^{\text{cum}} = \sum_{l=1}^k D_{q,l}$. From the recursive relation (A.5) it follows that $A_{q,k}^{\text{cum}} - D_{q,k}^{\text{cum}} \geq X - x_{q,0}$ implies $x_{q,k} \geq X$. Therefore, we have

$$1 \geq \lim_{k \rightarrow \infty} P(x_{q,k} \geq X) \geq \lim_{k \rightarrow \infty} P(A_{q,k}^{\text{cum}} - D_{q,k}^{\text{cum}} \geq X - x_{q,0}).$$

Using (A.4) and (A.6) we find:

$$\begin{aligned} \lim_{k \rightarrow \infty} P(A_{q,k}^{\text{cum}} - D_{q,k}^{\text{cum}} \geq X - x_{q,0}) &= \lim_{k \rightarrow \infty} P\left(\frac{A_{q,k}^{\text{cum}} - D_{q,k}^{\text{cum}}}{k} \geq \frac{X - x_{q,0}}{k}\right), \\ &= P(\rho_q - f_q > 0) = 1, \end{aligned}$$

which proves the lemma. \square

Lemma A.6. *Consider an intersection with two conflicting queues that satisfies Assumption A.1. Let this intersection be controlled with the actuated controller that is introduced in Section A.2.2. Then with probability 1, some finite amount of time exists after which the vehicle actuated controller behaves as a pre-timed controller that alternates between: an effective green interval of signal group 1 of \bar{g}_1 slots, a clearance time of $\underline{c}_{1,2}$ slots, an effective green interval of signal group 2 of \bar{g}_2 slots, a clearance time of $\underline{c}_{2,1}$ slots.*

Proof. Consider an infinite time horizon $[0, \infty)$. Let $g_{q,k}$, $q = 1, 2$ be the duration of the k th green time of queue q . Define g_q to be the average effective green time of queue q over the infinite time horizon:

$$g_q := \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K g_{q,k}.$$

By using Assumption A.1 we find the following inequality on the total effective green fraction of signal groups 1 and 2 combined:

$$\begin{aligned} \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K \mathbb{E}(D_{1,k}) + \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K \mathbb{E}(D_{2,k}) &= \frac{g_1 + g_2}{g_1 + g_2 + c_{1,2} + c_{2,1}} \\ &\leq \frac{\bar{g}_1 + \bar{g}_2}{\bar{g}_1 + \bar{g}_2 + c_{1,2} + c_{2,1}} \\ &< \rho_1 + \rho_2. \end{aligned}$$

As a consequence, from Lemma A.5 it follows that with a probability of one either $x_1(t)$ or $x_2(t)$ is unbounded (or both). Without loss of generality assume that queue $x_1(t)$ is unbounded. This implies that after some finite amount of time t^* , signal group 1 always receives an effective green time of \bar{g}_1 . As a consequence, from Assumption A.1 it follows that during the interval $[t^*, \infty)$ signal group 2 is effective green for at most a fraction:

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K \mathbb{E}(D_{2,k}) = \frac{g_2}{\bar{g}_1 + g_2 + c_{1,2} + c_{2,1}} \leq \frac{\bar{g}_2}{\bar{g}_1 + \bar{g}_2 + c_{1,2} + c_{2,1}} < \rho_2.$$

Hence, it follows from Lemma A.5 that with a probability of one $x_2(t)$ is also unbounded. This implies that (with a probability of one) after some finite amount of time both signal group 1 and signal group 2 receive only effective green intervals with a maximum duration of \bar{g}_1 respectively \bar{g}_2 slots, which implies the lemma. \square

Appendices of Chapter 2

B.1 Propagation of the back of the queue and its effects

In Section 2.2.1 we have calculated the delay d_{det} associated with a deterministic and fluid-like queueing process. In that section we have implicitly assumed that all cars awaiting a green indication do not occupy any space. In other words, we assumed that the back of the queue is positioned at the stop line and cars join this queue when they reach this stop line. In reality this is not the case: cars do occupy space and, as a result, the back of the queue propagates. Consider for example the propagation of the back of the queue during a green indication as depicted in Figure B.1. In this figure we distinguish between *queued cars* and *free flowing cars*; a free flowing car becomes a queued car whenever it reaches the back of the queue, and a queued car again becomes a free flowing car when it has passed the stop line. Between time t_1 and time t_2 the back of the queue propagates backwards. Such a backward propagation might arise because time is needed for the effect of a departing car to propagate (via a shockwave) to the cars at the back of the queue; during this time a car may arrive at the back of the queue. Between time t_3 and time t_4 a forward propagation can be seen, which is caused by the departure of cars.

B.1.1 Effect on the arrival process

The propagation of the back of the queue affects the arrival process of cars to the queue. For example consider the case that each free flowing car has a velocity of v meters per second. Moreover, let the back of the queue propagate forward with the same velocity of v meters per second. Free flowing cars then never arrive at the back of the queue. This example indicates that the propagation of the back of the queue affects the arrival process of cars to this queue.

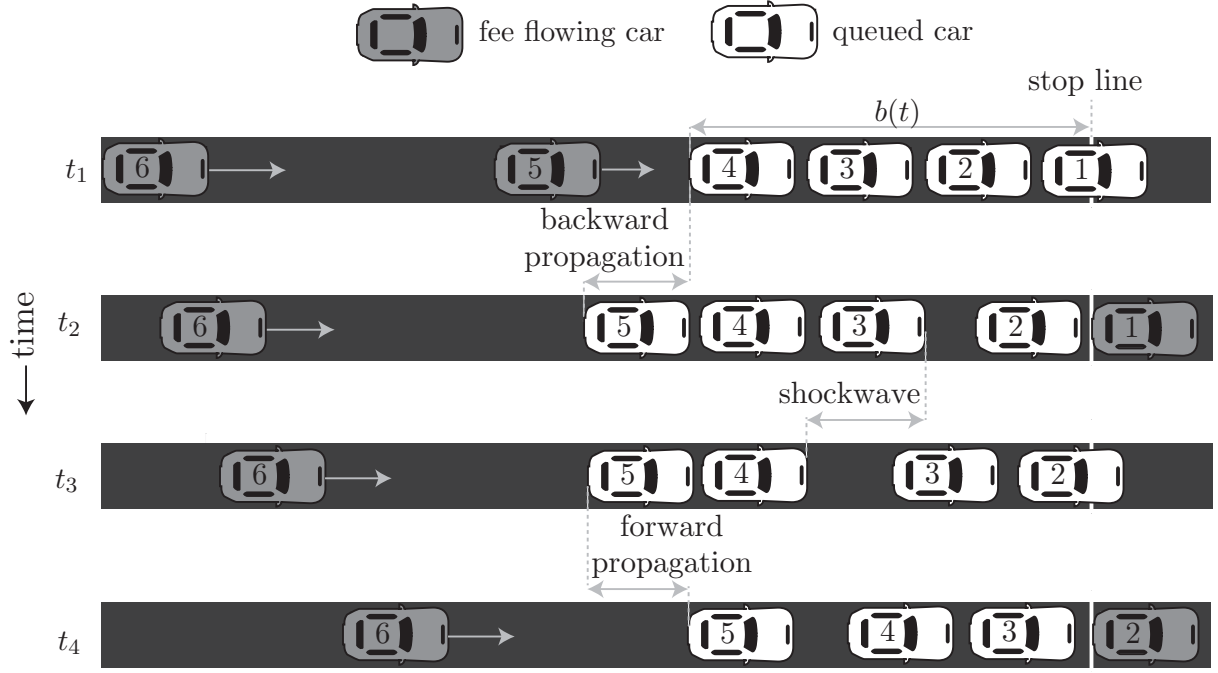


Figure B.1: A possible propagation of the back of the queue during a green indication.

B.1.2 Effect on the delay

The propagation of the back of the queue affects the arrival process of traffic to that queue, however, we show that it does not affect the delay that cars experience. First, we introduce some notation. Consider a lane as seen in Figure B.1. A car has departed whenever the entire car has crossed the stop line, e.g., in Figure B.1, car 1 (2) departs at time t_2 (t_4). Define $t_{f,k}$ as the time at which car k would depart the lane whenever it only experiences free flowing conditions (the car does not experience hinder from a queue) and let $t_{q,k}$ be the time at which car k departs the lane when also taking into account the queueing conditions (possibly experiencing hinder from a queue). The delay d_k that car k experiences is defined as $d_k := t_{q,k} - t_{f,k}$.

Define $b(t)$ to be the distance between the back of the queue and the stop line at time t , i.e., $b(t)$ represents the propagation of the back of the queue. We define $b(t)$ to equal zero whenever the queue is empty. During an effective green interval, a free flowing car becomes a queued vehicle when it reaches the back of the non-empty queue; if a free-flowing vehicle reaches the back of an empty queue (i.e., it reaches the stop line without encountering a queue) during an effective green interval, this vehicle can depart without experiencing any delay. During an effective red interval, a free flowing vehicle becomes a queued vehicle when it arrives at the back of the (possibly empty) queue. We consider two different ways of modeling the queue:

- 1 cars in the queue do occupy space, and, as a result, the back of the queue propagates.
- 2 cars in the queue do not occupy any space, i.e., $b(t) = 0$ for all t . This is what we implicitly have assumed in Section 2.2.1.

We prove that the delay that cars experience is the same for both models. To this end, we prove that the departure time of car k ($t_{q,k}$) is the same for both models, which would prove that the delay $d_k := t_{q,k} - t_{f,k}$ that car k experiences is the same for both models. We assume the following:

Assumption B.1. *The departure time $t_{q,k}$ of car k depends (only) on the previous departure times ($t_{q,k'}, k' < k$), on the timing of the traffic light and on whether or not car k encounters a (non-empty) queue before departure or not.*

Assumption B.2. *The distance between two queued cars, say (the front of) car $k + 1$ and (the back of) car k , equals zero when car k departs from the queue or when car $k + 1$ arrives to the queue.*

The above assumption implies the following. Whenever car k and car $k + 1$ are both in the queue, car $k + 1$ is positioned at the stop line at time $t_{d,k}$. Furthermore, it implies that a car joins the queue whenever its front reaches the back of the queue. In reality the distance between cars is non-zero; to account for this distance we can model the cars to have a length that is equal to the actual size of the car plus some (safety) distance in between the cars. Furthermore, we assume the following:

Assumption B.3. *The time at which a car reaches the stop line under queued conditions, is not smaller than the time needed to reach the stop line under free flow conditions.*

Let $t_{q,k}^i$ be the departure time of car k for model $i = 1, 2$. We prove that these departure times are equal for both models, i.e., $t_{q,k}^1 = t_{q,k}^2$ for all k . Assume to contrary: these departure times are not equal for some car, i.e., $t_{q,k}^1 \neq t_{q,k}^2$ for some k . Let car k be the first such car, i.e., $t_{q,k}^1 \neq t_{q,k}^2$ and $t_{q,k'}^1 = t_{q,k'}^2$ for all $k' < k$. From Assumption B.1 it then follows that car k encounters a non-empty queue for one of the models, but it encounters an empty queue for the other model.

First, consider the case that car k encounters a non-empty queue for model 1 (the queue does occupy space) and encounters an empty queue for model 2 (the queue does not occupy any space). Let t^* be the time at which car k arrives at the stop line (which is the back of the queue) for model 2. Since, the queue is then empty it holds that $t^* > t_{q,k-1}$. From Assumption B.2 it follows that car k reaches the stop line at time $t_{q,k-1}$ for model 1. This implies that car k reaches the stop line sooner under queued conditions than under free flow conditions, which contradicts Assumption B.3.

Now consider the case that car k encounters an empty queue for model 1 and encounters a non-empty queue for model 2. Let $b^i(t)$ be the position of the back of the queue for model $i = 1, 2$ as a function of time. Recall that $b^2(t) := 0$ for all t . Let t^* be the time at which car k arrives at the stop line for model 2. Since, the queue is then non-empty it holds that $t^* \leq t_{q,k-1}$. Since, $b^1(t) \geq b^2(t)$ for all t , car k must have already arrived at the queue at time t^* for model 1. This implies that car k also arrives to a non-empty queue for model 1. This proves that the departure times are the same for both models.

Since, the departure times are the same for both models, also the delay that cars experience is the same delay for both models. Therefore, we may assume w.l.o.g. that the back of the queue is positioned at the stop line when calculating the delay that cars experience.

Appendices of Chapter 3

C.1 A motivation for negative clearance times

In this appendix we motivate the use of negative minimum clearance times. Consider queue 1 and queue 2 of the intersection in Figure C.1. Their traffic streams have a conflict. For safety reasons, the traffic from these two queues must not arrive at this conflict simultaneously. The conflict between these two traffic streams is located close to the stop line of queue 1 but (relatively) far away from the stop line of queue 2. As a consequence, it may be safe for queue 2 to become effective green even before queue 1 becomes effective red, i.e., it may be safe to have a negative minimum clearance time between queue 1 becoming effective red and queue 2 becoming effective green; for such a negative clearance time the traffic from queue 1 may already have passed the conflict before the traffic from queue 2 arrives at this conflict. Below we motivate these negative clearance times in a more formal manner.

Define t_i^s (t_i^e) to be the starting (ending) lost time of queue $i = 1, 2$. Let the minimum clearance time between queue 1 becoming effective red and queue 2 becoming effective green be based on the following worst-case scenario:

- A relatively slow vehicle (vehicle 1) departs from queue 1 at time t_1^R , which is the time at which its traffic light switches to a red indication. Let t_1^{slow} be the time required by vehicle 1 to drive from the stop line of queue 1 to the conflict with queue 2, i.e., vehicle 1 arrives at the conflict with queue 2 at time $t_1^R + t_1^{\text{slow}}$.
- A relatively fast vehicle (vehicle 2) departs from queue 2 at time t_2^G , which is the time at which its traffic light switches to a green indication. Let t_2^{fast} be the time required by vehicle 2 to drive from the stop line of queue 2 to the conflict with queue 1, i.e., vehicle 2 arrives at the conflict with queue 1 at time $t_2^G + t_2^{\text{fast}}$.
- Note that queue 1 becomes effective red at time $t_1^r := t_1^R - t_1^e$ and that queue 2 becomes effective green at time $t_2^g := t_2^G + t_2^s$. For the worst case scenario it holds that the clearance time $t_2^g - t_1^r$ equals the minimum clearance time.

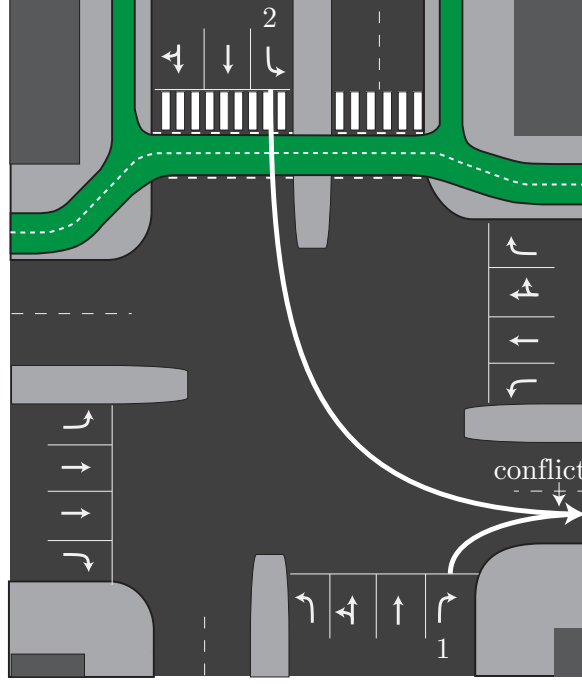


Figure C.1: An intersection for which a negative clearance time between queue 1 and queue 2 is motivated.

The minimum clearance time $t_2^g - t_2^r$ is defined such that vehicle 2 arrives at the conflict (between queue 1 and queue 2) $\delta > 0$ seconds after vehicle 1 arrives at this conflict for the above worst-case scenario, i.e., $t_2^G + t_2^{\text{fast}} = t_1^R + t_1^{\text{slow}} + \delta$. The above worst case scenario implies the following equality on the minimum clearance time $t_2^g - t_1^r$:

$$t_2^g - t_1^r = (t_2^G + t_2^s) - (t_1^R - t_1^e) = (t_2^s + t_1^e + t_1^{\text{slow}} + \delta) - t_2^{\text{fast}}.$$

As queue 1 is positioned much closer to the conflict than queue 2, the value for t_2^{fast} may be much larger than t_1^{slow} and even larger than $t_2^s + t_1^e + t_1^{\text{slow}} + \delta$. This motivates the use of negative clearance times.

C.2 An invalid triangular inequality on the minimum clearance times

In this section we consider the following triangular inequality on the minimum clearance times:

$$\underline{c}_{i_1, i_3} \leq \underline{c}_{i_1, i_2} + \underline{c}_{i_2, i_3}. \quad (\text{C.1})$$

where signal groups i_1, i_2 and i_3 are conflicting with each other. One might believe that this inequality holds for any real-life intersection. However, in Figure C.2 we give an intersection for which this inequality is probably not satisfied. For this example, the conflict between signal group i_1 and signal group i_2 is positioned close to the stop line of signal group i_1 and relatively far away from the stop line of signal group i_2 . Therefore, as motivated by Appendix C.1, the minimum clearance time \underline{c}_{i_1, i_2} is small and possibly even negative. In the same manner, we can motivate that the minimum clearance time \underline{c}_{i_2, i_3} is small or possibly even negative. As a consequence, the right-hand side of (C.1) is small (and possibly negative). On the contrary, we can motivate that the left-hand side of this equation has to be quite large. The conflict between signal group i_1 and signal group i_3 is positioned close to the stop line of signal group i_3 and relatively far away from the stop line of signal group i_1 . Hence, the minimum clearance time \underline{c}_{i_1, i_3} has to be quite large to ensure a safe passage for the traffic from signal group i_3 . Therefore, the triangular inequality (C.1) is probably not satisfied for the intersection in Figure C.2.

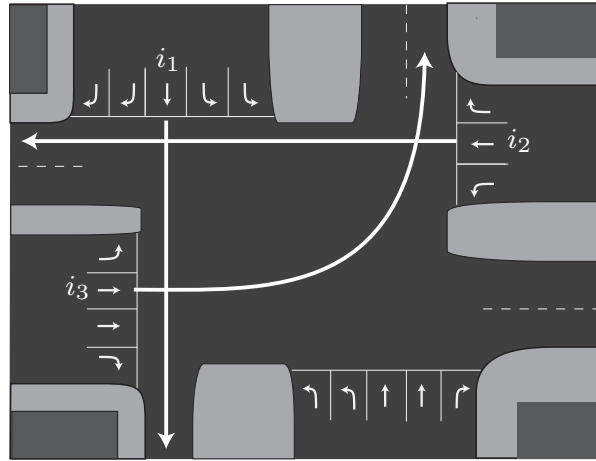


Figure C.2: An intersection for which the triangular inequality and the circutal inequality on the minimum clearance times are possibly not satisfied.

C.3 An invalid circital inequality on the minimum clearance times

In this appendix we consider the following circital inequality on the minimum clearance times:

$$\sum_{k=1}^N \underline{c}_{i_k, i_{k+1}} > 0.$$

where $i_{N+1} := i_1$ and $\{i_k, i_{k+1}\} \in \Psi_{\mathcal{S}}$ for each $k = 1, \dots, N$. One might believe that this inequality holds for any real-life intersection. However, in Figure C.2 we give an intersection for which this inequality does possibly not hold; for this intersection, the circital constraint

$$\underline{c}_{i_1, i_2} + \underline{c}_{i_2, i_3} + \underline{c}_{i_3, i_1} > 0$$

is possibly not satisfied. For this example, the conflict between signal group i_1 and signal group i_2 is positioned close to the stop line of signal group i_1 and relatively far away from the stop line of signal group i_2 . Therefore, as motivated by Appendix C.1, the minimum clearance time \underline{c}_{i_1, i_2} might even be negative. In the same manner, we can motivate the negativity of \underline{c}_{i_2, i_3} and \underline{c}_{i_3, i_1} . Therefore, the circital constraint is possibly not satisfied for this intersection.

We motivate that for $N = 2$ the circital constraint is always satisfied in practice. Consider a pair of conflicting signal groups $\{i, j\} \in \Psi_{\mathcal{S}}$. Consider a queue $1 \in \mathcal{Q}_i$ and a queue $2 \in \mathcal{Q}_j$ whose traffic streams cannot safely cross the intersections, i.e., these two queues have a conflict. As motivated in Appendix C.1, we have:

$$\begin{aligned} \underline{c}_{i,j} &= (t_2^s + t_1^e + t_1^{\text{slow}} + \delta) - t_2^{\text{fast}}, \\ \underline{c}_{j,i} &= (t_1^s + t_2^e + t_2^{\text{slow}} + \delta) - t_1^{\text{fast}}, \end{aligned}$$

where t_1^{slow} (t_2^{slow}) is the time it takes a relatively slow car in queue 1 (queue 2) to drive from the stop line to the conflict with queue 2 (queue 1) when it departs at the start of a green interval. Moreover, t_1^{fast} (t_2^{fast}) is the time it takes a relatively fast car in queue 1 (queue 2) to drive from the stop line to the conflict with queue 2 (queue 1) when it departs at the end of a yellow interval. Moreover, δ is some positive number. From these inequalities it follows that:

$$\underline{c}_{i,j} + \underline{c}_{j,i} = (t_1^s + t_1^e) + (t_2^s + t_2^e) + (t_1^{\text{slow}} - t_1^{\text{fast}}) + (t_2^{\text{slow}} - t_2^{\text{fast}}) + 2\delta > 0.$$

Each of these five terms is very likely to be (strictly) positive, which motivates that for $N = 2$ the circital constraint is always satisfied in practice.

C.4 Convexity of the approximation of Van den Broek

In this appendix we prove convexity of the approximation of (van den Broek et al., 2006). Let μ_q be the saturation flow rate of queue q in passenger car equivalent per second (PCE/s); when traffic contains different types of traffic, e.g., cars, trucks and motorcycles, its arrival rate and capacity can be expressed in a single unit that is called passenger car equivalent units. Van den Broek divides a period into constant time intervals, so called slots; each slot has a length of $1/\mu_q$ seconds. Let ρ_q and σ_q be the average and the standard deviation of the number of road users in passenger car equivalent (PCE) that arrive during a slot at queue q , i.e., on average $\lambda_q := \rho_q \mu_q$ PCE of road users arrive per second at queue q . For ease of notation we define $r'_i := \gamma(\textcircled{i}, \textcircled{i})$. The approximation of van den Broek for a queue $q \in \mathcal{Q}_i$ can be written as:

$$d_q = \frac{r'_i}{2(1 - \rho_q)\rho_q} \left(\frac{\sigma_q^2}{\mu_q(1 - \rho_q)} + \frac{\rho_q r'_i}{\mathbf{T}'} + \frac{r'_i \rho_q^2 \sigma_q^2}{\mu_q(1 - r'_i)^2(1 - r'_i - \rho_q)(1 - \rho_q)} \right).$$

Note that stability constraint (3.8e), the strictly positive red times (3.8d) and circuitual constraint (3.8i) together imply $0 < \rho_q \leq 1 - r'_i < 1$ and note that $\sigma_q, \mathbf{T}' \geq 0$. Let $\rho_q := \alpha_q(1 - r'_i)$, where $0 < \alpha_q \leq 1$. The equation for d_q is convex if $\det(H) \geq 0$ and $\frac{\partial^2 d_q}{\partial \mathbf{T}'^2} \geq 0$, where H is the Hessian of this equation. We show these inequalities below:

$$\begin{aligned} \frac{\partial^2 d_q}{\partial \mathbf{T}'^2} &= \frac{r_i'^2}{\mathbf{T}'^3(1 - \alpha_q + r'_i)} \geq 0, \\ \det(H) &= \frac{\alpha_q \sigma_q^2 r_i'^2 ((1 - \alpha_q) + r'_i)^2 + 2(1 - \alpha_q)^2 r'_i}{\mu_q(1 - \alpha_q)^3(1 - \alpha_q + r'_i)^3(1 - r'_i)^4 \mathbf{T}'^3} \geq 0, \end{aligned}$$

C.5 Proofs of lemmas

C.5.1 Proof of Lemma 3.1

In Lemma 3.1 we use the following definition of the constraint graph G :

$$\begin{aligned} V &= \{\textcircled{i} \mid i \in \mathcal{S}\} \cup \{\textcircled{i} \mid i \in \mathcal{S}\}, \\ A &= A_g \cup A_r \cup A_c, \\ \text{where,} \\ A_g &= \{(\textcircled{i}, \textcircled{i}) \mid i \in \mathcal{S}\}, \\ A_r &= \{(\textcircled{i}, \textcircled{i}) \mid i \in \mathcal{S}\}, \\ A_c &= \{(\textcircled{i}, \textcircled{j}) \mid \{i, j\} \in \Psi_S\}. \end{aligned} \tag{C.2}$$

Lemma 3.1 *Let \mathcal{F}' be a spanning forest of the conflict graph G' (as defined in (3.5)) and let \mathcal{F} be the spanning forest of the constraint graph G (as defined in (C.2)) calculated*

with (3.6). Define $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ to be the SFCB of the constraint graph G defined by spanning forest \mathcal{F} , and let \mathcal{B}' be the set of cycles obtained from \mathcal{B} when, for each conflict $\{i, j\} \notin \mathcal{F}'$, $i < j$ we replace the cycle $\mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i}))$ by the cycle:

$$\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \textcircled{i}), (\textcircled{i}, \textcircled{j}), (\textcircled{j}, \textcircled{j}), (\textcircled{j}, \textcircled{i})\}.$$

The set \mathcal{B}' is an integral cycle basis of the constraint graph G that includes all cycles associated with circutal constraints (3.3g)–(3.3h).

Proof. The SFCB \mathcal{B} obtained from the spanning forest \mathcal{F} includes the cycles associated with circutal constraints (3.3g); each arc $(\textcircled{i}, \textcircled{i})$ is not included in the spanning forest \mathcal{F} and results in one such cycle. However, this SFCB does not include all of the cycles associated with circutal constraint (3.3h). Consider a conflict $\{i, j\} \in \Psi_{\mathcal{S}}$ and assume w.l.o.g. that $i < j$. The cycle associated with circutal constraint (3.3h) of the conflict $\{i, j\} \in \Psi_{\mathcal{S}}$ is the cycle:

$$\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \textcircled{i}), (\textcircled{i}, \textcircled{j}), (\textcircled{j}, \textcircled{j}), (\textcircled{j}, \textcircled{i})\}. \quad (\text{C.3})$$

This cycle is included in the SFCB if and only if $\{i, j\} \in \mathcal{F}'$; this is then the cycle $\mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i}))$. Consider a conflict $\{i, j\} \in \Psi_{\mathcal{S}}$, $i < j$ for which $\{i, j\} \notin \mathcal{F}'$. For each such conflict we replace the cycle $\mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i}))$ by the cycle (C.3). The resulting set \mathcal{B}' includes all cycles associated with circutal constraints (3.3g)–(3.3h). Furthermore, this set is an integral cycle basis; we prove this via induction in the remaining part of this proof.

Let \mathcal{B}_k be a set of cycles that is obtained when we have done the replacement (that is described in the previous paragraph) for k conflicts. We use the induction hypothesis that \mathcal{B}_k is an integral cycle basis and we prove that the set \mathcal{B}_{k+1} is then also an integral cycle basis. Note that $\mathcal{B}_0 := \mathcal{B}$ is a SFCB, which is by definition an integral cycle basis.

The set of cycles \mathcal{B}_{k+1} can be obtained from a set $\mathcal{B}_k := \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ by performing one additional replacement. Let $\{i, j\} \in \Psi_{\mathcal{S}}$, $i < j$ be the conflict for which we perform this additional replacement. For such a conflict it holds that $\{i, j\} \notin \mathcal{F}'$. As a consequence, from the definition of \mathcal{B} it follows that $\mathcal{C}_{\mathcal{F}}((\textcircled{i}, \textcircled{j})) \in \mathcal{B}$ and that $\mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i})) \in \mathcal{B}$. The previous $k - 1$ replacements did not affect these cycles and, as a consequence, $\mathcal{C}_{\mathcal{F}}((\textcircled{i}, \textcircled{j})) \in \mathcal{B}_k$ and $\mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i})) \in \mathcal{B}_k$. Assume w.l.o.g. that $\mathcal{C}_{d-1} := \mathcal{C}_{\mathcal{F}}((\textcircled{i}, \textcircled{j}))$ and that $\mathcal{C}_d := \mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i}))$. We replace the cycle \mathcal{C}_d . From the induction hypothesis it follows that $\mathcal{B}_k := \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ is an integral cycle basis. Therefore, for each cycle \mathcal{C} in the constraint graph G we can find $\alpha \in \mathbb{Z}^d$ such that:

$$\mathcal{C} = \alpha_1 \mathcal{C}_1 + \dots + \alpha_d \mathcal{C}_d.$$

The cycles \mathcal{C}_{d-1} and \mathcal{C}_d are visualized in Figure C.3. Let \mathcal{C}'_d be the cycle associated with circutal constraint (3.3h) of the conflict $\{i, j\} \in \Psi_{\mathcal{S}}$, i.e., \mathcal{C}'_d is the cycle (C.3). Note that the cycle-arc incidence vector of the cycle \mathcal{C}'_d satisfies $\mathcal{C}'_d := \mathcal{C}_{d-1} + \mathcal{C}_d$, see Figure C.3; each arc $a \in \mathcal{C}'_d$ is used in the forward direction by either the cycle \mathcal{C}_{d-1} or by the cycle \mathcal{C}_d

(not both) and each arc $a \notin \mathcal{C}'_d$ used by the cycle \mathcal{C}_{d-1} (\mathcal{C}_d) is used by the cycle \mathcal{C}_d (\mathcal{C}_{d-1}) in the opposite direction. Hence, for each cycle \mathcal{C} in the constraint graph G we can find $\alpha \in \mathbb{Z}^d$ such that:

$$\begin{aligned} \mathcal{C} &= \alpha_1 \mathcal{C}_1 + \dots + \alpha_d \mathcal{C}_d, \\ &= \alpha_1 \mathcal{C}_1 + \dots + \alpha_{d-2} \mathcal{C}_{d-2} + (\alpha_{d-1} - \alpha_d) \mathcal{C}_{d-1} + \alpha_d \mathcal{C}'_d, \\ &= \alpha_1 \mathcal{C}_1 + \dots + \alpha_{d-2} \mathcal{C}_{d-2} + \alpha'_{d-1} \mathcal{C}_{d-1} + \alpha_d \mathcal{C}'_d, \end{aligned}$$

which implies that we can write each cycle \mathcal{C} as an integral combination of the cycles in the set \mathcal{B}_{k+1} ; this implies that \mathcal{B}_{k+1} is an integral cycle basis and concludes the proof.

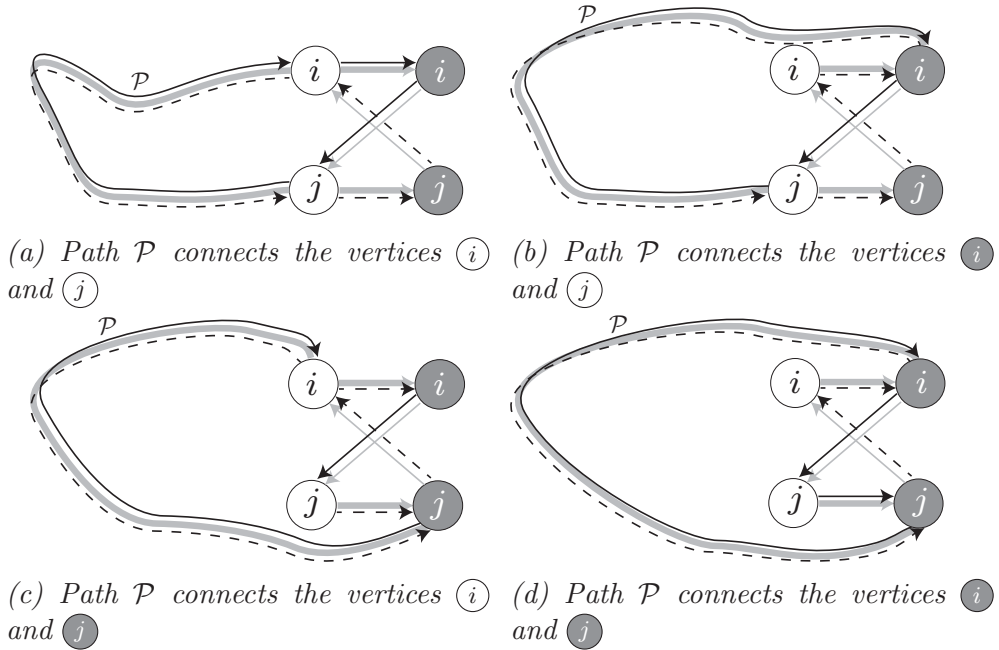


Figure C.3: These figures visualize the cycles $\mathcal{C}_{d-1} := \mathcal{C}_{\mathcal{F}}((\textcircled{i}, \textcircled{j}))$ and $\mathcal{C}_d := \mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i}))$. The solid black line visualizes the cycle \mathcal{C}_{d-1} and the dotted black line visualizes the cycle \mathcal{C}_d . Furthermore, the gray lines visualize the relevant arcs of graph G ; the arcs that are in the spanning forest \mathcal{F} are visualized in bold. Path \mathcal{P} consists of the arcs in the spanning forest \mathcal{F} that both the cycle \mathcal{C}_{d-1} and the cycle \mathcal{C}_d use (in opposite directions). We visualize four different situations for this path \mathcal{P} .

□

C.5.2 Proof of Lemma 3.2

Lemma 3.2 Consider a cycle basis $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ obtained by using Lemma 3.1. Consider a cycle $\mathcal{C} \in \mathcal{B}$ that does not equal the zero-width cycle associated with the circuital constraint (3.8i). For each such cycle, we can relate its integral-valued design variable $z_{\mathcal{C}}$ to the binary design variables $\Omega_{i,j}$ as follows:

$$z_{\mathcal{C}} = \sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}^+} \Omega_{i,j} - \sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}^-} \Omega_{i,j}. \quad (\text{C.4})$$

Proof. For a cycle $\mathcal{C} \in \mathcal{B}$ associated with circuitual constraint (3.8h), the equation to prove, reduces to:

$$z_{\mathcal{C}} = \Omega_{i,j} + \Omega_{j,i} = 1,$$

which is indeed true; for these cycles we fix the value of $z_{\mathcal{C}}$ to one.

Now consider some other cycle $\mathcal{C} \in \mathcal{B}$ that is not associated with some circuitual constraint (3.8i). Recall that the arcs $a \in A_g \subset A$ represent effective green intervals and that the arcs $a \in A_c \subset A$ represent clearance intervals. Assume that the cycle \mathcal{C} alternates between a forward arc in A_g and a forward arc in A_c ; at the end of this proof, we prove that we can do so without loss of generality. With this cycle we can associate a periodic sequence of vertices:

$$(i_1), (i_2), (i_2), (i_3), (i_3), \dots, (i_N), (i_N), \quad (\text{C.5})$$

where $v_N = v_1$ and $N > 3$. The equation to prove (C.4), then reduces to:

$$z_{\mathcal{C}} = \sum_{k=1}^{N-1} \Omega_{i_k, i_{k+1}}.$$

The cycle periodicity constraint (3.8g) of the circuit (C.5) gives the following expression for $z_{\mathcal{C}}$:

$$\begin{aligned} z_{\mathcal{C}} &= \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \gamma(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \gamma(\varepsilon_1, \varepsilon_2) \\ &= \sum_{k=1}^{N-1} \gamma(i_k, i_{k+1})', \end{aligned} \quad (\text{C.6})$$

where,

$$\gamma(i, j)' := \gamma((i), (j)) + \gamma((j), (i)).$$

By combining circuitual constraint (3.8h) with well-posedness constraint (3.8j) we find that:

$$\gamma(i_k, i_{k+1})' \in (0, 1), \quad k = 1, \dots, N-1.$$

Note that $\gamma(i_k, i_{k+1})'$ is the time between a switch to effective green of signal group i_k and the subsequent switch to effective green of signal group i_{k+1} . Define $f(i) := f((i))$, which is the time (expressed as a fraction of the period duration) at which signal group i becomes effective green. Recall that $\Omega_{i,j}$ is defined such that $\Omega_{i,j} = 0$ if $f((i)) \leq f((j))$ and $\Omega_{i,j} = 1$ otherwise. Since, $\gamma(i_k, i_{k+1})' \in (0, 1)$ it holds that:

$$\gamma(i_k, i_{k+1})' = f(i_{k+1}) - f(i_k) + \Omega_{i_k, i_{k+1}},$$

Substituting this equation in (C.6) and using $v_N = v_1$ gives:

$$z_{\mathcal{C}} = \sum_{k=1}^{N-1} \Omega_{i_k, i_{k+1}},$$

which proves the lemma.

What remains is to prove that we can indeed assume w.l.o.g. that the cycle \mathcal{C} alternates between a forward arc in A_g and a forward arc in A_c . Let \mathcal{F} be the spanning forest that is calculated with (3.6). Let $\mathcal{C}_{\mathcal{F}}(a)$, $a \notin \mathcal{F}$ be the unique circuit in $\mathcal{F} \cup \{a\}$ that uses the arc a in the forward direction. By definition of the cycle basis \mathcal{B} , each cycle $\mathcal{C} \in \mathcal{B}$ is equal to $\mathcal{C}_{\mathcal{F}}(a)$ for some arc $a \in A_c$. Consider such a circuit $\mathcal{C}_{\mathcal{F}}(a)$. With this circuit we can associate a sequence of vertices:

$$v_1, v_2, v_3, v_4, \dots, v_N, \quad (\text{C.7})$$

where $(v_1, v_2) = a$, $v_N = v_1$ and $N > 3$. Assume that this sequence does not alternate between a forward arc in A_g and a forward arc in A_c , i.e., the sequence (C.7) does not equal the desired sequence (C.5). Let v_k be the first vertex in the sequence (C.7) that breaks this desired sequence; note that $k > 2$ as $(v_1, v_2) := (\textcircled{i}, \textcircled{j})$ for some conflict $\{i, j\} \in \Psi_{\mathcal{S}}$. Consider the case that $v_{k-1} = \textcircled{i}$ for some signal group $i \in \mathcal{S}$. Since v_{k-2} by definition follows the desired sequence it holds that $v_{k-2} = \textcircled{i}$. Vertex v_k must be some other vertex $v \neq v_{k-2}$ of the constraint graph G that is adjacent to vertex v_{k-1} ; however, all these adjacent vertices satisfy the desired sequence (see Figure C.4), which contradicts the definition of v_k . Therefore, it must hold that:

$$v_{k-1} = \textcircled{i} \quad \text{for some } i \in \mathcal{S}.$$

Since the vertex v_k is the first vertex that breaks the desired sequence (C.5), it holds that:

$$v_{k-2} = \textcircled{j} \quad \text{for some } j \neq i.$$

The vertex v_{k-1} is only connected (via an arc) with vertices that represent a switch to effective red. Therefore, we have:

$$v_k = \textcircled{j'} \quad \text{for some } j'.$$

By definition the vertex v_k does not satisfy the sequence (C.5), which implies $j' \neq i$. Thus, as $v_{k-1} := \textcircled{i}$ and $v_k := \textcircled{j'}$, $j' \neq i$, the cycle $\mathcal{C}_{\mathcal{F}}(a)$ traverses the arc $(\textcircled{j'}, \textcircled{i})$ in the backward direction. Moreover, this implies $(\textcircled{j'}, \textcircled{i}) \in \mathcal{F}$. Furthermore, the sequence (C.7) is associated with a circuit, which implies $j' \neq j$. Recall that spanning forest \mathcal{F} includes each arc that represents an effective green interval, i.e., it includes all the arcs $(\textcircled{i}, \textcircled{i})$, $i \in \mathcal{S}$ (see its definition (3.6)). Therefore, in addition to the arc $(\textcircled{j'}, \textcircled{i})$, the spanning forest \mathcal{F} includes the arcs $(\textcircled{i}, \textcircled{i})$ and $(\textcircled{j'}, \textcircled{j'})$. Since $\mathcal{F} \cup \{a\}$ contains a single circuit this implies that:

$$(\textcircled{i}, \textcircled{j'}) \notin \mathcal{F} \cup \{a\},$$

otherwise $\mathcal{F} \cup \{a\}$ would also contain the following circuit \mathcal{C}'' , which does alternate between a forward arc in A_g and a forward arc in A_c :

$$\mathcal{C}'' = \mathcal{C}''^+ = \{(\textcircled{i}, \textcircled{i}), (\textcircled{i}, \textcircled{j'}), (\textcircled{j'}, \textcircled{j'}), (\textcircled{j'}, \textcircled{i})\}.$$

Note that the circuit \mathcal{C}'' is the cycle associated with circuital constraint (3.8h) of the conflict $\{i, j'\} \in \Psi_S$. Define the spanning forest $\tilde{\mathcal{F}}$ as follows:

$$\tilde{\mathcal{F}} = F \setminus \{(\textcircled{j'}, \textcircled{i})\} \cup \{(\textcircled{i}, \textcircled{j'})\}.$$

Thus, we obtain the spanning forest $\tilde{\mathcal{F}}$ by replacing the arc $(\textcircled{j'}, \textcircled{i})$ of spanning forest \mathcal{F} by the arc $(\textcircled{i}, \textcircled{j'})$. Let C be the cycle-arc incidence vector associated with the circuit $\mathcal{C} := \mathcal{C}_{\mathcal{F}}(a)$, $a = (v_1, v_2)$, and define C' to be the cycle-arc incidence vector associated with the circuit $\mathcal{C}' := \mathcal{C}_{\tilde{\mathcal{F}}}(a)$. The cycle \mathcal{C}' replaces the backward arc $(\textcircled{j'}, \textcircled{i})$ of \mathcal{C} by the forward arc $(\textcircled{i}, \textcircled{j'})$. Note that these cycle-arc incidence vectors satisfy (see also Figure C.5):

$$C' = C + C'',$$

As a consequence,

$$z_{C'} = C' \gamma = (C + C'') \gamma = z_C + z_{C''} = z_C + 1.$$

Furthermore, it holds that:

$$\sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}'^+} \Omega_{i,j} - \sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}'^-} \Omega_{i,j} = \sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}^+} \Omega_{i,j} - \sum_{(\textcircled{i}, \textcircled{j}) \in \mathcal{C}^-} \Omega_{i,j} + 1,$$

which follows from the fact that the cycle \mathcal{C}' uses the arc $(\textcircled{i}, \textcircled{j'})$ in the forward direction instead of the arc $(\textcircled{j'}, \textcircled{i})$ in the backward direction; as a consequence, the term $-\Omega_{i,j'} = \Omega_{i,j'} - 1$ is replaced by the term $\Omega_{i,j'}$. Therefore, circuit $\mathcal{C}_{\mathcal{F}}(a)$ satisfies (C.4) if and only if the circuit $\mathcal{C}_{\tilde{\mathcal{F}}}(a)$ satisfies (C.4). Furthermore, by its definition, the circuit $\mathcal{C}_{\tilde{\mathcal{F}}}(a)$ can be associated with a sequence of vertices:

$$v_1, v_2, \dots, v_{k-1}, v'_k, v'_{k+1}, v'_{k+2}, \dots, v'_N. \quad (\text{C.8})$$

where $(v_1, v_2) = a$, $v'_k = \textcircled{i}$ and $v'_{k+1} = \textcircled{j'}$. Thus, this sequence satisfies the desired sequence (C.5) at least till vertex v'_{k+1} . Therefore, by repeating this process, we can find a cycle that alternates between a forward arc in A_g and a forward arc in A_c . This cycle satisfies (C.4) if and only if the cycle $\mathcal{C} := \mathcal{C}_{\mathcal{F}}(a)$ satisfies (C.4). Therefore, we may assume w.l.o.g. that the circuit \mathcal{C} alternates between a forward arc in A_g and a forward arc in A_c . This concludes this proof. \square

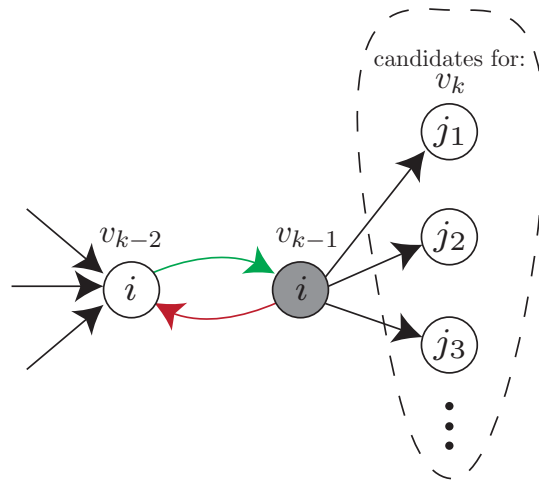
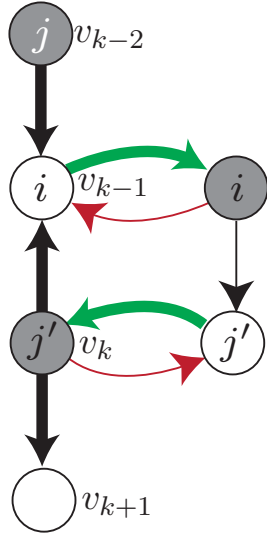
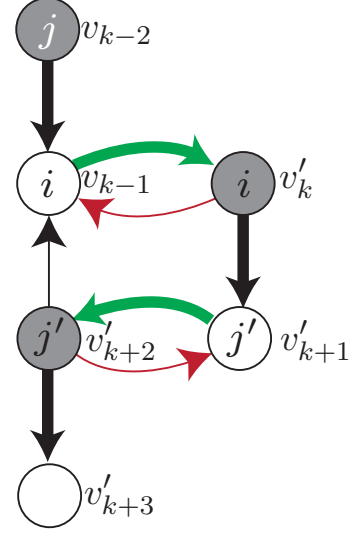


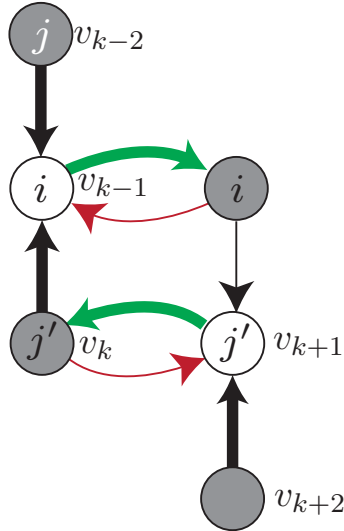
Figure C.4: Visualization of the following statement: for a circuit v_1, v_2, \dots, v_N , $N > 3$ in constraint graph G , the vertex v_k represents a switch to effective green when the vertices v_{k-2} and v_{k-1} represent the switch to effective green respectively the switch to effective red of the same signal group i ; we have only visualized the relevant vertices and arcs of the constraint graph G .



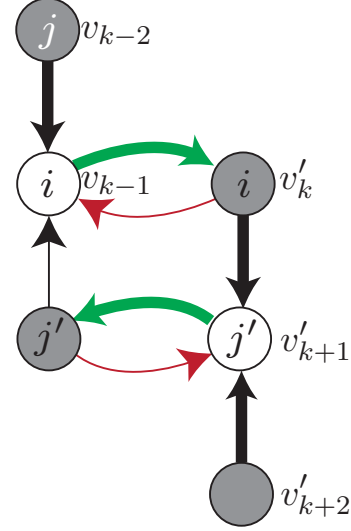
(a) The sequence of vertices v_1, \dots, v_N associated with the only circuit in $\mathcal{F} \cup \{a\}$ if $v_{k+1} \neq \langle j' \rangle$. The arcs in $\mathcal{F} \cup \{a\}$ are visualized in bold.



(b) The alternative sequence of vertices $v_1, \dots, v_{k-1}, v'_k, \dots, v'_{N+1}$ associated with the only circuit in $\tilde{\mathcal{F}} \cup \{a\}$ if $v_{k+1} \neq \langle j' \rangle$. The arcs in $\tilde{\mathcal{F}} \cup \{a\}$ are visualized in bold.



(c) The sequence of vertices v_1, \dots, v_N associated with the only circuit in $\mathcal{F} \cup \{a\}$ if $v_{k+1} = \langle j' \rangle$. The arcs in $\mathcal{F} \cup \{a\}$ are visualized in bold.



(d) The alternative sequence of vertices $v_1, \dots, v_{k-1}, v'_k, \dots, v'_{N+1}$ associated with the only circuit in $\tilde{\mathcal{F}} \cup \{a\}$ if $v_{k+1} = \langle j' \rangle$. The arcs in $\tilde{\mathcal{F}} \cup \{a\}$ are visualized in bold.

Figure C.5: These figures concern the situation that the cycle $\mathcal{C}_{\mathcal{F}}(a)$ does not alternate between an arc in $a \in A_g$ and an arc in $a \in A_c$. This circuit can be represented by a sequence of vertices v_1, \dots, v_N where v_k is the first vertex that breaks the alternating sequence. In these figures, we visualize a part of the sequence v_1, \dots, v_N (Figure C.5a and Figure C.5c) and a part of an alternative sequence $v_1, \dots, v_{k-1}, v'_k, \dots, v'_{N+1}$ (Figure C.5b and Figure C.5d); the alternative sequence is associated with the only circuit in $\tilde{\mathcal{F}} \cup \{a\}$, where $\tilde{\mathcal{F}} := \mathcal{F} \setminus \{(\langle j' \rangle, \langle i \rangle)\} \cup \{(\langle i \rangle, \langle j' \rangle)\}$ is a spanning forest.

Appendices of Chapter 4

D.1 Convexity of the approximation of Van den Broek extended to multiple realizations

Define $r'_{i,k} := \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k)$ and define $r'_i := r'_{i,1} + \dots + r'_{i,K_i}$. When the formula of (van den Broek et al., 2006) is extended, in the straightforward manner that is described in Section 4.2.3, we obtain:

$$d_q := d_q^{\text{stoch}} + \sum_{k \in \mathcal{K}_i} d_{q,k}^{\text{det}},$$

where,

$$d_{q,k}^{\text{det}} := \frac{r'^2_{i,k}}{2\mathbf{T}'(1 - \rho_q)},$$

$$d_q^{\text{stoch}} := \frac{r'_i}{2(1 - \rho_q)\rho_q} \left(\frac{\sigma_q^2}{\mu_q(1 - \rho_q)} + \frac{r'_i \rho_q^2 \sigma_q^2}{\mu_q(1 - r'_i)^2(1 - r'_i - \rho_q)(1 - \rho_q)} \right).$$

First, we prove that $d_{q,k}^{\text{det}}$ is convex in $r'_{i,k}$ and \mathbf{T}' by proving that its Hessian H and its second derivative to \mathbf{T}' are non-negative:

$$\frac{\partial^2 d_{q,k}^{\text{det}}}{\partial \mathbf{T}'^2} = \frac{r'^2_{i,k}}{\mathbf{T}'^3(1 - \rho_q)} \geq 0,$$

$$\det(H) = 0.$$

The stochastic delay term d_q^{stoch} is only a function of r'_i (not of \mathbf{T}'). Hence, we prove its convexity by proving that the second derivative to r'_i is non-negative. The stability constraint (4.1d), the strictly positive effective red times (4.1c), and circuital constraint (4.1j) together imply $0 < \rho_q \leq 1 - r'_i < 1$. Define α_q such that $\rho_q := \alpha_q(1 - r'_i)$, where $0 < \alpha_q \leq 1$ we have:

$$\frac{\partial^2 d_q^{\text{stoch}}}{\partial r_i'^2} = \frac{\alpha_q \sigma_q^2 ((1 - \alpha_q) + r'_i)^2 + 2(1 - \alpha_q)^2 r'_i}{\mu_q (1 - \alpha_q)^3 (1 - \alpha_q + r'_i)^3 (1 - r'_i)^4} \geq 0.$$

The convexity of $d_{q,k}^{\text{det}}$, $k \in \mathcal{K}_i$ and the convexity of d_q^{stoch} imply the convexity of d_q .

D.2 Queue emptying and convexity of the delay formula

Consider a signal group $i \in \mathcal{S}$ and a queue $q \in \mathcal{Q}_i$ that is controlled by signal group i . In Section 4.2.3 we split the delay in a deterministic delay term and a stochastic delay term. This deterministic delay term is associated with a purely deterministic and fluid-like arrival and departure process. In that section we force queue q to be emptied during each effective green interval for this purely deterministic queueing process. The resulting approximation is then convex, see Appendix D.1. In this section we prove that the delay at queue q is not a convex function of the real-valued design variables when we do not force that queue to be emptied during each effective green interval for this purely deterministic queueing process. First, we prove that the deterministic delay term is then not necessarily a convex function. Subsequently we prove that also the approximation of (van den Broek et al., 2006) extended to multiple realizations (as done in Section 4.2.3) is not convex.

D.2.1 Convexity of the deterministic term

We show that the deterministic delay term is not convex by using an example. We consider a queue $q \in \mathcal{Q}_i$ with a load of $\rho = 0.5$ and an arrival rate of $\lambda = 0.25$.

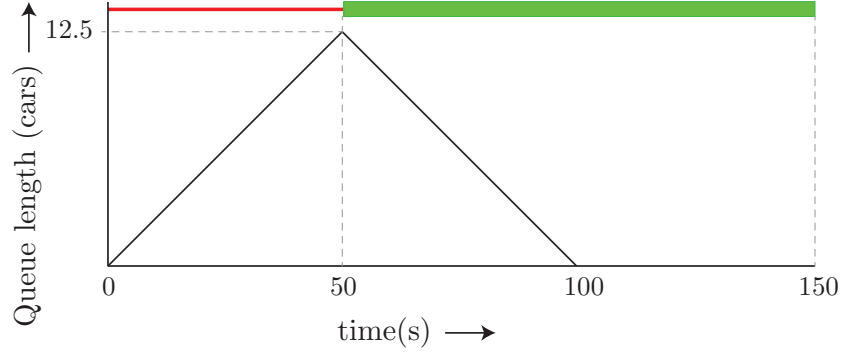
Consider a signal group diagram with a period duration of 150 seconds for which signal group i has two realizations ($K_i = 2$); signal group i is effective green during the interval $[50, 50]$ and during the interval $[50, 150]$, see also Figure D.1a. Define the deterministic delay term d_q^{det} as:

$$d_q^{\text{det}} := \sum_{k=1}^{\mathcal{K}_i} d_{q,k}^{\text{det}}.$$

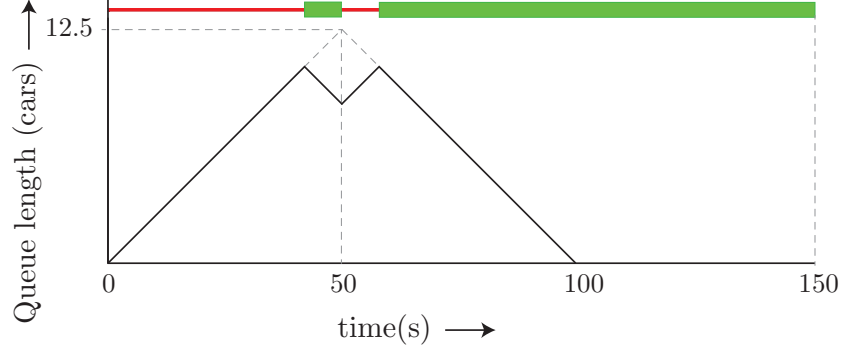
This deterministic delay term can be calculated from the average queue length, which equals $(0.5 \cdot 12.5 \cdot 100)/150 = 25/6$, by applying Little's law (Chhajed and Lowe, 2008); this yields a deterministic delay term d_q^{det} of $100/6$ seconds.

Consider the similar signal group diagram that is visualized in Figure D.1b. Its period duration is again 150 seconds and signal group i again has two realizations ($K_i = 2$); signal group i is effective green during the interval $[42, 50]$ and during the interval $[58, 150]$. For this signal group diagram, the deterministic delay term d_q^{det} can be calculated to be $1186/75$ seconds.

Now take the convex combination of the former two signal group diagrams (each fifty percent). The resulting signal group diagram has a period duration of 150 seconds. Moreover, signal group i is effective green during the intervals $[46, 50]$ and $[54, 150]$. If the deterministic delay term would be a convex function, the deterministic delay term associated with this signal group diagram would be at most $0.5 \cdot 100/6 + 0.5 \cdot 1186/75 = 1218/75$. However, similar to the previous two cases we can determine that its deterministic delay term is larger and equal to $1234/75$ seconds, which contradicts convexity.



(a) Signal group i is effective green during the interval $[50, 50]$ and during the interval $[50, 150]$.



(b) Signal group i is effective green during the interval $[42, 50]$ and during the interval $[58, 150]$.

Figure D.1: A queue length evolution of queue $q \in \mathcal{Q}_i$ for a deterministic and fluid-like queueing process when $\lambda_q = 0.25$ and $\mu_q = 0.5$.

D.2.2 Convexity of the delay

In Section 4.2.3 we have extended the approximation of (van den Broek et al., 2006) in a straightforward manner to the case of multiple realizations. This approximation d_q can be split into a deterministic d_q^{det} term and a stochastic delay term d_q^{stoch} . In this section we prove that this approximation is not convex whenever the queue is not forced to be emptied during each realization. To this end, we consider the same example as in the previous section. Refer to the signal group diagrams of Figure D.1a and Figure D.1b as signal group diagram 1 respectively signal group diagram 2. Furthermore, refer to their convex combination (each 50 percent) as signal group diagram 3. Let d_q^k , $d_q^{\text{det},k}$ and $d_q^{\text{stoch},k}$ be the approximated delay of queue q , the deterministic delay term of queue q , respectively the stochastic delay term of queue q for signal group diagram k . We have already showed that:

$$0.5d_q^{\text{det},1} + 0.5d_q^{\text{det},2} < d_q^{\text{det},3}. \quad (\text{D.1})$$

Furthermore, we can prove that the stochastic delay term is the same for these three signal groups, i.e.,

$$0.5d_q^{\text{stoch},1} + 0.5d_q^{\text{stoch},2} = d_q^{\text{stoch},3}. \quad (\text{D.2})$$

The equations (D.1) and (D.2) together imply:

$$0.5d_q^1 + 0.5d_q^2 < d_q^3,$$

which proves that the approximation d_q is not convex. The equality (D.2) follows from the following observation. The signal group diagrams of Figure D.1a and Figure D.1b (as well as each convex combination of these signal group diagrams) have the same period duration and the same total effective red time. The stochastic delay term depends only on the total effective red time, which implies (D.2).

D.3 Proofs of lemmas

D.3.1 Proof of Lemma 4.1

The lemma below generalizes Lemma 3.1. See Appendix C.5.1 for that lemma. Furthermore, their proofs are very similar. Therefore, we recommend to first read the proof of Lemma 3.1 before reading the proof of the lemma below. In the following lemma we use the following definition of the constraint graph G :

$$\begin{aligned} V &= \{(\textcircled{i}_k) \mid i \in \mathcal{S}, k \in \mathcal{K}_i\} \cup \{(\textcircled{i}_k) \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}, \\ A &= A_g \cup A_r \cup A_c, \end{aligned}$$

where,

$$\begin{aligned} A_g &:= \{((\textcircled{i}_k), (\textcircled{i}_k)) \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}, \\ A_r &:= \{((\textcircled{i}_{k-1}), (\textcircled{i}_k)) \mid i \in \mathcal{S}, k \in \mathcal{K}_i\}, \\ A_c &:= \{((\textcircled{i}_k), (\textcircled{j}_{k'})) \mid \{(i, k), (j, k')\} \in \Psi_R\}, \\ \Psi_R &:= \{\{(i, k), (j, k')\} \mid \{i, j\} \in \Psi_S, k \in \mathcal{K}_i, k' \in \mathcal{K}_j\}. \end{aligned}$$

Lemma 4.1 *Let \mathcal{F}' be a spanning forest of the conflict graph G' and let \mathcal{F} be the spanning forest of the constraint graph G calculated with (4.2). Define $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ to be the SFCB of graph G defined by spanning forest \mathcal{F} , and let \mathcal{B}' be the set of cycles obtained from \mathcal{B} when, for each arc $(\textcircled{i}_k, \textcircled{j}_{k'}) \notin \mathcal{F}$, $i < j$, we replace the cycle $\mathcal{C}_{\mathcal{F}}((\textcircled{j}_{k'}, \textcircled{i}_k))$ by the cycle:*

$$\mathcal{C} = \mathcal{C}^+ = \{((\textcircled{i}_k), (\textcircled{i}_k)), ((\textcircled{i}_k), (\textcircled{j}_{k'})), ((\textcircled{j}_{k'}, (\textcircled{j}_{k'})), ((\textcircled{j}_{k'}, (\textcircled{i}_k)))\}.$$

The set \mathcal{B}' is an integral cycle basis of the constraint graph G that includes all the cycles associated with circuitual constraints (4.1i)–(4.1j).

Proof. The proof goes analogously to the proof of Lemma 3.1. The differences are as follows:

- For each signal group $i \in \mathcal{S}$, the arc $((\textcircled{i}_K), (\textcircled{i}_1))$ (instead of the arc $((\textcircled{i}), (\textcircled{i}))$) is not included in spanning forest \mathcal{F} and results in a cycle associated with (4.1i) (instead of (3.3g)).

- The replacement is performed for each conflict $\{(i, k), (j, k')\} \in \Psi_R$, $i < j$ for which $(\textcircled{i}_k, \textcircled{j}_{k'}) \notin \mathcal{F}$ (instead of for each conflict $\{i, j\} \in \Psi_S$, $i < j$ for which $\{i, j\} \notin \mathcal{F}'$).
- For this replacement we have $\mathcal{C}_{d-1} := \mathcal{C}_{\mathcal{F}}((\textcircled{i}_k, \textcircled{j}_{k'}))$ (instead of $\mathcal{C}_{d-1} := \mathcal{C}_{\mathcal{F}}((\textcircled{i}, \textcircled{j}))$), $\mathcal{C}_d := \mathcal{C}_{\mathcal{F}}((\textcircled{j}_{k'}, \textcircled{i}_k))$ (instead of $\mathcal{C}_d := \mathcal{C}_{\mathcal{F}}((\textcircled{j}, \textcircled{i}))$), and \mathcal{C}'_d is the cycle associated with circuitual constraint (4.1j) of conflict $\{(i, k), (j, k')\} \in \Psi_R$, $i < j$ (instead of circuitual constraint (3.3h) of conflict $\{i, j\} \in \Psi_S$, $i < j$).

□

D.3.2 Proof of Lemma 4.2

Lemma 4.2 *Let MIP_{fix} be the optimization problem with linear constraints (4.1) and let MIP_{var} be the optimization problem (4.8). Consider a signal group diagram for which the number of realizations K_i of each signal group $i \in \mathcal{S}$ satisfies $\underline{K}_i \leq K_i \leq \overline{K}_i$. This signal group diagram satisfies the constraints of MIP_{fix} if and only if it satisfies the constraints of MIP_{var} , i.e., MIP_{fix} has a solution that results in this signal group diagram if and only if MIP_{var} has a solution that results in this signal group diagram.*

Proof. Define $\mathcal{K}_i := \{1, \dots, K_i\}$. Signal group $i \in \mathcal{S}$ has K_i realizations for the signal group diagram that we consider. Therefore, its associated solution to MIP_{var} satisfies:

$$\begin{aligned} b_{i,k} &:= 1, & i \in \mathcal{S}, \quad k \in \overline{\mathcal{K}}_i \setminus \mathcal{K}_i, \\ b_{i,k} &:= 0, & i \in \mathcal{S}, \quad k \in \mathcal{K}_i. \end{aligned}$$

When fixing these binary variables as shown above, then the optimization problem MIP_{var} includes all the constraints of optimization problem MIP_{fix} . Therefore, any signal group diagram that does not satisfy the constraints of MIP_{fix} also does not satisfy the constraints of MIP_{var} . As a consequence, what remains is to prove that each signal group diagram that satisfies the constraints of MIP_{fix} also satisfies the constraints of MIP_{var} . Consider a solution $(\gamma_{\text{fix}}, \mathbf{T}'_{\text{fix}}, \mathbf{z}_{\text{fix}})$ that satisfies the linear constraints of MIP_{fix} . We construct a solution $(\mathbf{T}', \gamma, \mathbf{z}, \mathbf{b})$ to MIP_{var} that results in the same signal group diagram; this would conclude this proof. First, the period duration of both solutions should be the same:

$$\mathbf{T}' := \mathbf{T}'_{\text{fix}}.$$

Second, all effective green and effective red intervals must have the same duration:

$$\begin{aligned} \gamma(\textcircled{i}_k, \textcircled{i}_k) &:= \gamma_{\text{fix}}(\textcircled{i}_k, \textcircled{i}_k), & i \in \mathcal{S}, \quad k \in \mathcal{K}_i, \\ \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) &:= \gamma_{\text{fix}}(\textcircled{i}_k, \textcircled{i}_k), & i \in \mathcal{S}, \quad k \in \mathcal{K}_i. \end{aligned}$$

The remaining effective green and effective red intervals have a duration of zero, which is forced by (4.8j)–(4.8l):

$$\begin{aligned} \gamma(\textcircled{i}_k, \textcircled{i}_k) &:= 0, & i \in \mathcal{S}, \quad k \in \overline{\mathcal{K}}_i \setminus \mathcal{K}_i, \\ \gamma(\textcircled{i}_{k-1}, \textcircled{i}_k) &:= 0, & i \in \mathcal{S}, \quad k \in \overline{\mathcal{K}}_i \setminus \mathcal{K}_i. \end{aligned}$$

We use the definition of Ψ_R as defined in Section 4.3, i.e.,

$$\Psi_R := \{ \{(i, k), (j, k')\} \mid \{i, j\} \in \Psi_S, k \in \bar{\mathcal{K}}_i, k' \in \bar{\mathcal{K}}_j \}.$$

Consider a pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i$ and $k' \in \mathcal{K}_j$. The clearance time from effective green interval k of signal group i to effective green interval k' of signal group j are the same for both solutions. Thus, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i$ and $k' \in \mathcal{K}_j$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) := \gamma_{\text{fix}}(\textcircled{i}_k, \textcircled{j}_{k'}).$$

Consider a pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \notin \mathcal{K}_i$ and $k' \in \mathcal{K}_j$. It holds that effective green interval k of signal group i has a duration of zero seconds and directly follows effective green interval K_i of signal group i . As a result, for each conflict $\{(i, k), (j, k')\} \in \Psi_R$ with $k \notin \mathcal{K}_i$ and $k' \in \mathcal{K}_j$ it holds that:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) := \gamma_{\text{fix}}(\textcircled{i}_{K_i}, \textcircled{j}_{k'}). \quad (\text{D.3})$$

Consider a pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k' \notin \mathcal{K}_j$. Realization k' of signal group j directly follows realization K_j of signal group j . As a result, a clearance time to such an effective green interval k' of signal group j equals the clearance time to effective green interval K_j of signal group j plus the duration of this effective green interval, i.e., for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \in \mathcal{K}_i$, and $k' \notin \mathcal{K}_j$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) := \gamma_{\text{fix}}(\textcircled{i}_k, \textcircled{j}_{K_j}) + \gamma_{\text{fix}}(\textcircled{j}_{K_j}, \textcircled{j}_{K_j}),$$

and for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ with $k \notin \mathcal{K}_i$, and $k' \notin \mathcal{K}_j$ we have:

$$\gamma(\textcircled{i}_k, \textcircled{j}_{k'}) := \gamma_{\text{fix}}(\textcircled{i}_{K_i}, \textcircled{j}_{K_j}) + \gamma_{\text{fix}}(\textcircled{j}_{K_j}, \textcircled{j}_{K_j}).$$

The values of the integral-valued design variables \mathbf{z}_C can be calculated from (4.8n); their values depend on the cycle basis that is used. We can verify that the proposed solution indeed satisfies the constraints of MIP_{var} , which proves this lemma. \square

Appendices of Chapter 5

E.1 Proof of lemmas

E.1.1 Proof of Lemma 5.4

Lemma 5.4 *Algorithm 1 terminates in a finite number of iterations and either returns the signal group diagram with the smallest period duration amongst the signal group diagrams satisfying Prop(1) or it detects that no signal group diagram exists that satisfies Prop(1).*

Proof. Assume that a (feasible) signal group diagram exists that satisfies Prop(1). Define MILP_k as the MILP problem solved during the k th iteration of Algorithm 1, and let T_k be the period duration of the signal group diagram obtained by solving MILP_k . Furthermore, let \underline{T}_k be the lower bound that we have on the period duration for optimization problem MILP_k , i.e., $\underline{T}_{k+1} := \lceil T_k \rceil$ and \underline{T}_1 is the initial lower bound on the period duration. Define \tilde{T} to be the smallest period duration of any signal group diagram that satisfies Prop(1). This proof has the following ingredients:

1. The signal group diagram with the smallest period durations amongst the signal group diagrams satisfying Prop(1), satisfies the constraints of MILP_k for each k . This implies $T_k \leq \tilde{T}$.
2. When $T_{k+1} = T_k$, the algorithm has found a signal group diagram that satisfies Prop(1) during iteration $k + 1$.
3. For each iteration k except for the iteration in which the algorithm terminates it holds that $T_k \geq \underline{T}_1 + k - 1$.

These three statements together imply the lemma. From the third statement combined with the finite upper bound $\tilde{T} < \infty$ it follows that the algorithm terminates in a finite number of iterations. The first statement implies that whenever MILP_k is infeasible for some k , no signal group diagram exists that satisfies Prop(1). Furthermore, from the first two statements combined it follows that the optimal signal group diagram has been found

whenever in some iteration k the optimization problem MILP_k returns a signal group diagram that satisfies $\text{Prop}(1)$. The third statement together with the upper bound $\bar{T} < \infty$ proves that the algorithm terminates in a finite number of iterations.

We now prove the first statement via induction. By definition the optimal signal group diagram (amongst the ones that satisfy $\text{Prop}(1)$) satisfies the constraints of MILP_1 . We prove the induction step: this optimal signal group diagram satisfies the constraints of MILP_{k+1} if it satisfies the constraints of MILP_k . The period duration T_k is the minimum period duration of any signal group diagram that satisfies the constraints of MILP_k . Therefore, by our induction hypothesis it holds that $T_k \leq \tilde{T}$. Since \tilde{T} is integral this implies that $\underline{T}_{k+1} = \lceil T_k \rceil \leq \tilde{T}$. As a consequence, the optimal signal group diagram (amongst the ones that satisfy $\text{Prop}(1)$) remains feasible after the update in line 9. Furthermore, the optimal signal group diagram (amongst the ones that satisfy $\text{Prop}(1)$) satisfies $\text{Prop}(1)$. Therefore, for each signal group $i \in \mathcal{S}$ it satisfies (5.4) with $\underline{T} := \tilde{T}$. Since $\underline{T}_{k+1} \leq \tilde{T}$, this signal group diagram also satisfies (5.4) with $\underline{T} := \underline{T}_{k+1}$. As a result, the optimal signal group diagram remains feasible after the update in line 10. This proves the first statement.

We prove the second statement. By definition we have $T_{k+1} \geq \underline{T}_{k+1} = \lceil T_k \rceil \geq T_k$. Thus, the equality $T_{k+1} = T_k$ implies that $T_{k+1} = \underline{T}_{k+1} = \lceil T_k \rceil = T_k$ and, therefore, also that T_{k+1} is integral. Let G_i be the total green time of signal group $i \in \mathcal{S}$ for the signal group obtained in iteration $k+1$. From the update on line 10, it follows that the signal group diagram obtained during iteration $k+1$ satisfies (5.4) with $\underline{T} := T_{k+1}$ for each signal group $i \in \mathcal{S}$. This together with the integrality of T_{k+1} proves that the signal group diagram obtained during iteration $k+1$ satisfies $\text{Prop}(1)$.

Finally, we prove the third statement. The algorithm terminates whenever T^{k+1} equals T^k . As a consequence, the lower bound \underline{T} increases by at least one in each iteration (in line 9) except for the iteration in which the algorithm terminates. As a result, it holds that $\underline{T}^k \geq \underline{T}^1 + k - 1$. Combining this with the inequality $T^k \geq \underline{T}^k$ gives $T^k \geq \underline{T}^1 + k - 1$ and proves the third statement. \square

E.1.2 Proof of Lemma 5.5

Lemma 5.5 *Let the delay of any queue $q \in \mathcal{Q}$ be approximated with a function that is convex in the variables γ^{eff} and \mathbf{T}' , e.g., the delay of each queue $q \in \mathcal{Q}$ is approximated with the (extended version of the) approximation of (van den Broek et al., 2006). Then any local minimum of the function $D_{\mathbf{z},\mathbf{b}}(T)$ is also a global minimum.*

Proof. We prove this lemma by proving that the function $D_{\mathbf{z},\mathbf{b}}(1/\mathbf{T}')$ is a convex function of the reciprocal of the period duration \mathbf{T}' . This implies that any local minimum of the convex function $D_{\mathbf{z},\mathbf{b}}(1/\mathbf{T}')$ is a global minimum and proves this lemma.

Let $d_q(\gamma^{\text{eff}}, \mathbf{T}')$ be the (convex) approximate formulae used to approximate the delay that road users experience at queue $q \in \mathcal{Q}$. Define the delay $D(\gamma^{\text{eff}}, \mathbf{T}')$ as:

$$D(\gamma^{\text{eff}}, \mathbf{T}') := \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q(\gamma^{\text{eff}}, \mathbf{T}').$$

Note that $D(\gamma^{\text{eff}}, \mathbf{T}')$ is a convex function. Furthermore, recall that $D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}')$ is defined as the objective value returned by MIP problem (4.8) with additional constraint (4.5) and objective function:

$$\underset{\mathbf{T}', \gamma^{\text{eff}}, \mathbf{z}, \mathbf{b}}{\text{minimize}} \quad D(\gamma^{\text{eff}}, \mathbf{T}')$$

when the reciprocal of the period duration is fixed to the value \mathbf{T}' , the integral-valued and binary-valued design variables are fixed to the values in \mathbf{z} respectively \mathbf{b} . Consider two values \mathbf{T}'_1 and \mathbf{T}'_2 for which $D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}'_1)$ and $D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}'_2)$ are finite. We prove that for each two such reciprocals \mathbf{T}'_1 and \mathbf{T}'_2 , and for each $\alpha \in [0, 1]$ it holds that:

$$D_{\mathbf{z}, \mathbf{b}}(1/(\alpha \mathbf{T}'_1 + (1-\alpha) \mathbf{T}'_2)) \leq \alpha D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}'_1) + (1-\alpha) D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}'_2), \quad (\text{E.1})$$

which implies that $D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}')$ is a convex function of the reciprocal \mathbf{T}' . Let $(\gamma_1^{\text{eff}}, \mathbf{T}'_1, \mathbf{z}, \mathbf{b})$ respectively $(\gamma_2^{\text{eff}}, \mathbf{T}'_2, \mathbf{z}, \mathbf{b})$ be the solution to the linear constraints of MIP problem (4.8) associated with $D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}'_1)$ and $D_{\mathbf{z}, \mathbf{b}}(1/\mathbf{T}'_2)$, i.e., we have $D_{\mathbf{z}, \mathbf{b}}(1/T'_i) := D(\gamma_i^{\text{eff}}, T'_i)$, $i = 1, 2$. From the convexity of $D(\gamma^{\text{eff}}, \mathbf{T}')$ it follows that the convex combination of the two solutions $(\gamma_1^{\text{eff}}, \mathbf{T}'_1, \mathbf{z}, \mathbf{b})$ and $(\gamma_2^{\text{eff}}, \mathbf{T}'_2, \mathbf{z}, \mathbf{b})$ satisfies:

$$D(\alpha \gamma_1^{\text{eff}} + (1-\alpha) \gamma_2^{\text{eff}}, \alpha \mathbf{T}'_1 + (1-\alpha) \mathbf{T}'_2) \leq \alpha D(\gamma_1^{\text{eff}}, \mathbf{T}'_1) + (1-\alpha) D(\gamma_2^{\text{eff}}, \mathbf{T}'_2).$$

The solutions $(\gamma_1^{\text{eff}}, \mathbf{T}'_1, \mathbf{z}, \mathbf{b})$ and $(\gamma_2^{\text{eff}}, \mathbf{T}'_2, \mathbf{z}, \mathbf{b})$ satisfy the linear constraints of MIP problem (4.8) and (4.5). As a consequence, also their convex combination satisfies these linear constraints. This implies that:

$$D_{\mathbf{z}, \mathbf{b}}(1/(\alpha \mathbf{T}'_1 + (1-\alpha) \mathbf{T}'_2)) \leq D(\alpha \gamma_1^{\text{eff}} + (1-\alpha) \gamma_2^{\text{eff}}, \alpha \mathbf{T}'_1 + (1-\alpha) \mathbf{T}'_2).$$

We obtain the desired inequality (E.1) by combining the latter two inequalities with the definition $D_{\mathbf{z}, \mathbf{b}}(1/T'_i) := D(\gamma_i^{\text{eff}}, T'_i)$, $i = 1, 2$. This concludes the proof of this lemma. \square

E.2 Description of the polyhedron $P(\mathbf{u})$

Consider a signal group diagram that satisfies $\text{Prop}(\beta)$. Let $\tilde{\chi}^{\text{ind}}$ and \tilde{T}' be the associated values for χ^{ind} respectively \mathbf{T}' . Consider the rounding MILP problem (5.10) of this signal group diagram. Define the polyhedron $P(\mathbf{u})$ to be the polyhedron spanned by the constraints of this MILP problem when the binary-valued design variables are fixed to the values \mathbf{u} . We prove in this appendix that the polyhedron $P(\mathbf{u})$ can be written as follows:

$$P(\mathbf{u}) := \{\chi^{\text{ind}} \mid M_C \chi^{\text{ind}} = b_C, \underline{b}_P \leq M_P \chi^{\text{ind}} \leq \bar{b}_P, \underline{\chi} \leq \chi^{\text{ind}} \leq \bar{\chi}\},$$

where M_C is a cycle-arc incidence matrix associated with some integral cycle basis of the constraint graph G , M_P is a path-arc incidence matrix of the constraint graph G , and the vectors b_C , \underline{b}_P , \bar{b}_P , $\underline{\chi}$, and $\bar{\chi}$ are all integral.

The constraints (5.10a)–(5.10f) can be written as follows:

$$\underline{\chi} \leq \chi^{\text{ind}} \leq \bar{\chi}.$$

To this end, we set the elements of $\underline{\chi}$ and $\bar{\chi}$ as follows. For each signal group $i \in \mathcal{S}$ we have the following (integral) lower and upper bound on its first red time:

$$\begin{aligned} \underline{\chi}(\textcircled{i}_{K_i}, \textcircled{i}_1) &:= \lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \rfloor, \\ \bar{\chi}(\textcircled{i}_{K_i}, \textcircled{i}_1) &:= \min\{\lceil \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \rceil, \\ &\quad \tilde{T} - \sum_{k=2}^{K_i} (\lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) \rfloor + \mathbf{u}_{i,k}) \\ &\quad - \lfloor \sum_{k=1}^{K_i} \tilde{\chi}^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) \rfloor\}. \end{aligned}$$

The second term of the above minimum is an upper bound that follows from (5.10e). For each signal group $i \in \mathcal{S}$ with $K_i > 1$ we have the following (integral) lower and upper bound on its first red time:

$$\begin{aligned} \underline{\chi}(\textcircled{i}_{K_i}, \textcircled{i}_1) &:= \lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \rfloor, \\ \bar{\chi}(\textcircled{i}_{K_i}, \textcircled{i}_1) &:= \lceil \tilde{\chi}^{\text{ind}}(\textcircled{i}_{K_i}, \textcircled{i}_1) \rceil. \end{aligned}$$

Moreover, for each realization $k = 2, \dots, K_i$ of signal group $i \in \mathcal{S}$ we have the following (integral) lower and upper bound on its red time:

$$\begin{aligned} \underline{\chi}(\textcircled{i}_{k-1}, \textcircled{i}_k) &:= \lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) \rfloor + \mathbf{u}_{i,k}, \\ \bar{\chi}(\textcircled{i}_{k-1}, \textcircled{i}_k) &:= \lceil \tilde{\chi}^{\text{ind}}(\textcircled{i}_{k-1}, \textcircled{i}_k) \rceil + \mathbf{u}_{i,k}. \end{aligned}$$

Furthermore, for each signal group $i \in \mathcal{S}$ we have the following (integral) bounds on its k th green time:

$$\begin{aligned} \underline{\chi}(\textcircled{i}_k, \textcircled{j}_k) &:= \lfloor \tilde{\chi}^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_k) \rfloor, \\ \bar{\chi}(\textcircled{i}_k, \textcircled{j}_k) &:= \lceil \tilde{\chi}^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_k) \rceil. \end{aligned}$$

Furthermore, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ we have the following (integral) lower bound:

$$\underline{\chi}(\textcircled{i}_k, \textcircled{j}_{k'}) := \underline{C}_{i,j}.$$

Note that we have no direct upper bound on the variable $\chi^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_{k'})$. However, as this variable is bounded, we can (w.l.o.g.) include a (redundant) and integral upper bound on its value. For example, for each pair of conflicting realizations $\{(i, k), (j, k')\} \in \Psi_R$ we can set the upper bound $\bar{\chi}(\textcircled{i}_k, \textcircled{j}_{k'})$ to the following integral value:

$$\bar{\chi}(\textcircled{i}_k, \textcircled{j}_{k'}) := 1/\tilde{T}' - \underline{\chi}(\textcircled{j}_{k'}, \textcircled{i}_k).$$

Below we prove that this upper bound is indeed redundant. Recall that, as a consequence of the cycle periodicity constraints (5.10h), also the following circuital constraint is satisfied:

$$\chi^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k) + \chi^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_{k'}) + \chi^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_{k'}) + \chi^{\text{ind}}(\textcircled{j}_{k'}, \textcircled{j}_{k'}) = 1/\tilde{T}',$$

which is circuital constraint (4.1j) written in the coordinates χ^{ind} instead of the coordinates γ^{eff} . Combining the above circuital constraint with $\underline{\chi}(\textcircled{i}_k, \textcircled{i}_k) \geq 0$ and $\underline{\chi}(\textcircled{j}_{k'}, \textcircled{j}_{k'}) \geq 0$ implies the upper bound. We have now set all elements of the integral vectors $\underline{\chi}$ and $\bar{\chi}$.

The constraints that remain are the cycle periodicity constraints (5.10h) and the well-posedness constraints (5.10g). Cycle periodicity constraints (5.10h) can be written as:

$$M_C \chi^{\text{ind}} = b_C,$$

where M_C is the cycle-arc incidence matrix associated with some integral cycle basis of the constraint graph G and b_C is an integral vector. The vector b_C contains the values for z_C/\tilde{T}' associated with the cycles $C \in \mathcal{B}$; as z_C and $1/\tilde{T}'$ are both integral, it holds that b_C is indeed an integral vector.

Well-posedness constraints (5.10g) force the duration of a clearance time $\chi^{\text{ind}}(\textcircled{i}_k, \textcircled{j}_{k'})$ plus the preceding green time $\chi^{\text{ind}}(\textcircled{i}_k, \textcircled{i}_k)$ to be at least $\epsilon_{i,j} + \omega_\chi(\textcircled{i}_k, \textcircled{i}_k) + \omega_\chi(\textcircled{i}_k, \textcircled{j}_{k'}) := \epsilon_{i,j} + l_j^s - l_i^s$ seconds, which can be written as:

$$\epsilon_{i,j} + l_j^s - l_i^s \leq \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^+} \chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^-} \chi^{\text{ind}}(\varepsilon_1, \varepsilon_2),$$

where \mathcal{P} is a path in the constraint graph G with:

$$\mathcal{P}^+ := \{(\textcircled{i}_k, \textcircled{i}_k), (\textcircled{i}_k, \textcircled{j}_{k'})\} \text{ and } \mathcal{P}^- := \emptyset.$$

Besides this (integral) lower bound of $\epsilon_{i,j} + l_j^s - l_i^s$ seconds, we can also include a (redundant and integral) upper bound of $\bar{\chi}(\textcircled{i}_k, \textcircled{i}_k) + \bar{\chi}(\textcircled{i}_k, \textcircled{j}_{k'})$ on the length of this path, i.e.,

$$\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^+} \chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{P}^-} \chi^{\text{ind}}(\varepsilon_1, \varepsilon_2) \leq \bar{\chi}(\textcircled{i}_k, \textcircled{i}_k) + \bar{\chi}(\textcircled{i}_k, \textcircled{j}_{k'}).$$

Therefore, the well-posedness constraints and the introduced (redundant) upper bounds can be written as $\underline{b}_P \leq M_P \chi^{\text{ind}} \leq \bar{b}_P$, where \underline{b}_P and \bar{b}_P are integral vectors, and M_P is a path-arc incidence matrix. This proves that indeed the polyhedron $P(u)$ can be written in the proposed form.

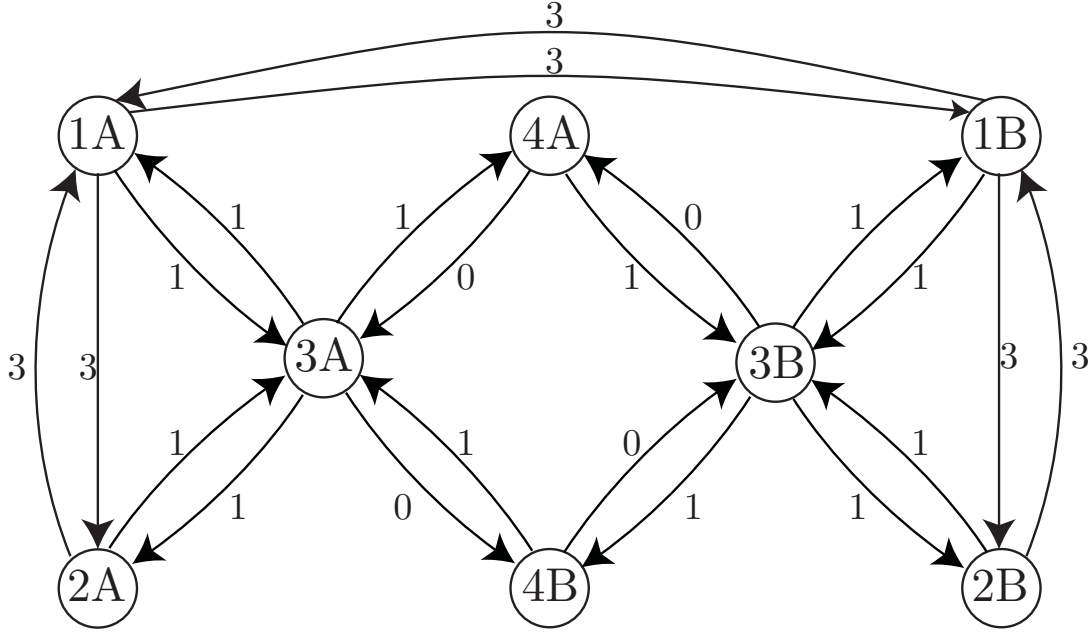


Figure E.1: Visualization of the conflicts and the minimum clearance times for a theoretical example. For this theoretical example, the first step of the two-step approach returns a signal group diagram that cannot be rounded in the second step. Two signal groups are conflicting if they are connected with an arc. Attached to each directed arc (i, j) is the minimum clearance time $c_{i,j}$.

E.3 The two-step approach for a theoretical example

In this appendix we show, by means of a theoretical example, that the rounding MILP problem might not always be able to find an integral signal group diagram. First, we give the data of this theoretical example. Subsequently, we show that for each of the objective functions considered in this thesis, the first step returns a signal group diagram that cannot be rounded in the second step.

E.3.1 Data of the theoretical example

In this section we give the data of the theoretical example. This example consists of the following eight signal groups:

$$\mathcal{S} := \{1A, 1B, 2A, 2B, 3A, 3B, 4A, 4B\}.$$

The conflicts between these signal groups and the minimum clearance times required between their realizations are visualized in Figure E.1.

Lost times and yellow times For each signal group $i \in \mathcal{S}$ the starting lost time (l_i^s), the ending lost time (l_i^e) and the yellow time (Y_i) are all zero. Note that for this theoretical example the effective green (effective red) mode of each signal group coincides with its green (red) indication.

Minimum and maximum period duration The minimum period duration (\underline{T}) equals 1 second and the maximum period duration (\overline{T}) equals 9 seconds.

Minimum and maximum green times For each signal group $i \in \mathcal{S}$, the minimum effective green time (\underline{g}_i) equals 1 second and the maximum effective green time (\overline{g}_i) is infinite.

Minimum and maximum red times For each signal group $i \in \mathcal{S}$, the minimum effective red time (\underline{r}_i) equals 1 second and the maximum effective red time (\overline{r}_i) is infinite.

Loads The loads (ρ_i) of the signal groups $i \in \mathcal{S}$ are as follows:

$$\begin{aligned} \rho_{1A}^{\text{SG}} &:= \frac{1}{10}, & \rho_{2A}^{\text{SG}} &:= \frac{2}{10}, & \rho_{3A}^{\text{SG}} &:= \frac{2}{10}, & \rho_{4A}^{\text{SG}} &:= \frac{3}{10}, \\ \rho_{1B}^{\text{SG}} &:= \frac{1}{10}, & \rho_{2B}^{\text{SG}} &:= \frac{2}{10}, & \rho_{3B}^{\text{SG}} &:= \frac{2}{10}, & \rho_{4B}^{\text{SG}} &:= \frac{3}{10}. \end{aligned}$$

Minimum and maximum number of realizations We consider the number of realizations to be fixed for each signal group diagram, i.e., $\underline{K}_i = \overline{K}_i$ for each signal group $i \in \mathcal{S}$. The signal groups 1A, 1B, 2A and 2B each have a single realization and the signal groups 3A, 3B, 4A and 4B each have two realizations, i.e.,

$$\begin{aligned} \underline{K}_{1A} = \overline{K}_{1A} &= 1, & \underline{K}_{2A} = \overline{K}_{2A} &= 1, & \underline{K}_{3A} = \overline{K}_{3A} &= 2, & \underline{K}_{4A} = \overline{K}_{4A} &= 2, \\ \underline{K}_{1B} = \overline{K}_{1B} &= 1, & \underline{K}_{2B} = \overline{K}_{2B} &= 1, & \underline{K}_{3B} = \overline{K}_{3B} &= 2, & \underline{K}_{4B} = \overline{K}_{4B} &= 2, \end{aligned}$$

E.3.2 Structural properties

Note that for this theoretical example, the structural property $\text{Prop}(\beta)$ reduces to the following one:

Prop(β). *Consider a signal group diagram with a period duration of T seconds. Let G_i be the total green time of signal group $i \in \mathcal{S}$ during one period of T seconds. This signal group diagram satisfies $\text{Prop}(\beta)$ if its period duration T is integral and for each signal group $i \in \mathcal{S}$ it holds that:*

$$G_i \geq \lfloor \beta \rho_i T \rfloor + 1.$$

E.3.3 Result of the first step of the two-step approach

Consider the minimization of the period duration of the signal group diagram. During the first step of the two-step approach we search for the optimal signal group diagram

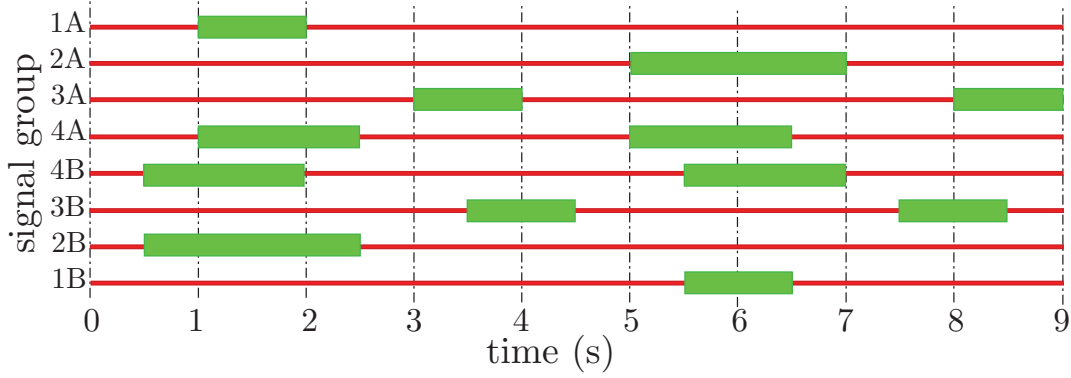


Figure E.2: The only signal group diagram that satisfies Prop(1). This signal group diagram cannot be rounded in the second step of the two-step approach.

amongst the signal group diagrams that satisfy Prop(1). For the proposed theoretical example, only one such signal group diagram exists. This signal group diagram is not integral and is visualized in Figure E.2. In Section E.3.4 we prove that this is indeed the only signal group diagram that satisfies Prop(1). Hence, when minimizing the period duration, the first step of the two-step approach returns the signal group diagram that is visualized in Figure E.2. Note that no integral signal group diagram exists; all integral signal group diagrams also satisfy Prop(1). Therefore, the rounding MILP problem of this signal group diagram is infeasible.

Consider the maximization of the capacity of the intersection. During the first step of the two-step approach we search for the maximum growth factor β^{\max} for which a signal group diagram exists that satisfies Prop(β^{\max}). The signal group diagram in Figure E.2 is the only signal group diagram that satisfies Prop(1). Furthermore, for any growth factor $\beta > 1$ either the signal group diagram in Figure E.2 is the only one satisfying Prop(β) or no signal group diagrams exists that satisfies Prop(β). This implies that the first step returns the signal group diagram in Figure E.2 when maximizing the capacity of the intersection. Therefore, also for this objective function the first step of the two-step approach returns a signal group diagram that cannot be rounded during the second step.

Consider the minimization of the average (weighted) delay that road users experience. The signal group diagram in Figure E.2 is the only one satisfying Prop(1) and the inequalities (4.5). Therefore, also when minimizing the average (weighted) delay that road users experience, the first step of the two-step approach returns a signal group diagram that cannot be rounded during the second step.

Remark E.1. For this example the rounding MILP is infeasible because no feasible integral signal group diagram exists at all. However, the rounding MILP could also be infeasible when an integral signal group diagram exists. For example, suppose we omit the conflict between signal group 1A and signal group 1B. Then also the integral signal group diagram in Figure E.3 might be feasible; this signal group diagram does not satisfy the constraints (4.5) and is, therefore, not feasible when minimizing the delay that road

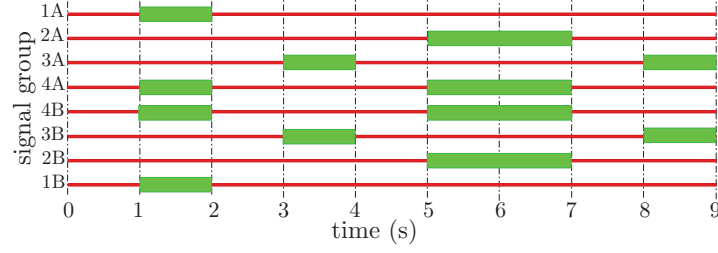


Figure E.3: A feasible integral signal group diagram for the theoretical example, when we omit the conflict between signal group 1A and signal group 1B.

users experience at the intersection. However, the first step can then still return the signal group diagram in Figure E.2, which can still not be rounded by solving rounding MILP problem (5.10); the values for the integral-valued design variables z_C , $C \in \mathcal{B}$ are fixed for this rounding MILP problem, and the signal group diagrams in Figure E.2 and Figure E.3 correspond to different values for the integral-valued design variables z_C , $C \in \mathcal{B}$.

E.3.4 One feasible signal group diagram

What remains to prove is that the signal group diagram in Figure E.2 is the only one satisfying Prop(1). In this section we prove this statement. First we prove that the period duration of any signal group diagram satisfying Prop(1) has a period duration of 9 seconds

E.3.5 Period duration

First we prove that any signal group diagram satisfying Prop(1) has a period duration of at least 9 seconds. From the maximum period duration of 9 seconds it then follows that any signal group diagram satisfying Prop(1) must have a period duration of 9 seconds. Consider the signal groups 1A and 2A. A lower bound on the period duration of any signal group diagram can be obtained as follows. Recall that the signal groups 1A and 2A have a single green interval. The minimum amount of time that is spend on the clearance times between the green interval of signal group 1A and the green interval of signal group 2A equals $c_{1A,2A} + c_{2A,1A} = 6$. Let G_i be the (effective) green time of signal group $i \in \mathcal{S}$. We then have:

$$T \geq G_{1A} + G_{2A} + 6.$$

Using the stability constraint (4.8g) we get:

$$T \geq \frac{6}{1 - \rho_{1A} - \rho_{2A}}, \quad (\text{E.2})$$

$$\geq 8 + \frac{4}{7}. \quad (\text{E.3})$$

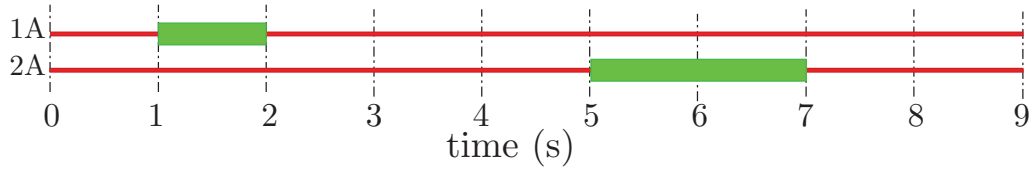
As any signal group diagram that satisfies property Prop(1) has an integral period duration, this implies that the period duration of each such signal group diagram must be at least 9 seconds. From the maximum period duration of 9 seconds it follows that any signal group diagram that satisfies property Prop(1) has a period duration of 9 seconds.

E.3.6 Partial signal group diagram

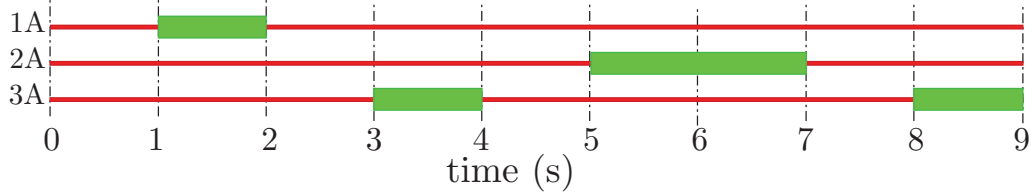
To satisfy Prop(1), the total green time G_i of each signal group $i \in \mathcal{S}$ must satisfy the following inequalities:

$$\begin{aligned} G_{1A} &\geq 1, & G_{2A} &\geq 2, & G_{3A} &\geq 2, & G_{4A} &\geq 3, \\ G_{1B} &\geq 1, & G_{2B} &\geq 2, & G_{3B} &\geq 2, & G_{4B} &\geq 3, \end{aligned}$$

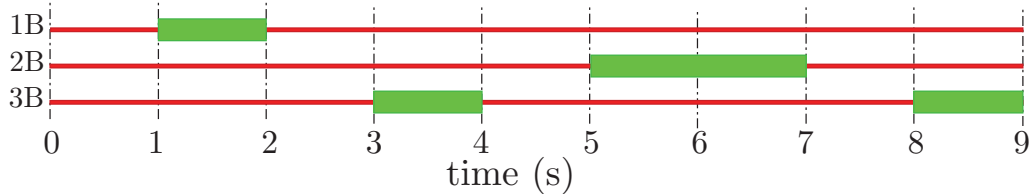
Note that the signal groups 1A and 2A are conflicting and only receive enough green time if their realizations are scheduled (relative to each other) as is shown in Figure E.4a. Signal group 3A is also conflicting with signal groups 1A and 2A. Furthermore, this signal group has two realizations. To receive enough green time, signal group 3A must be scheduled in the gaps between the realizations of signal group 1A and signal group 2A, see Figure E.4b.



(a) Partial signal group diagram for signal groups 1A and 2A.



(b) Partial signal group diagram for signal groups 1A, 2A and 3A.



(c) Partial signal group diagram for signal groups 1B, 2B and 3B.

Figure E.4: Partial signal group diagrams for the theoretical example.

In exactly the same manner, we can motivate that the realizations of the signal groups 1B, 2B and 3B must be scheduled (relative to each other) as is shown in Figure E.4b. We

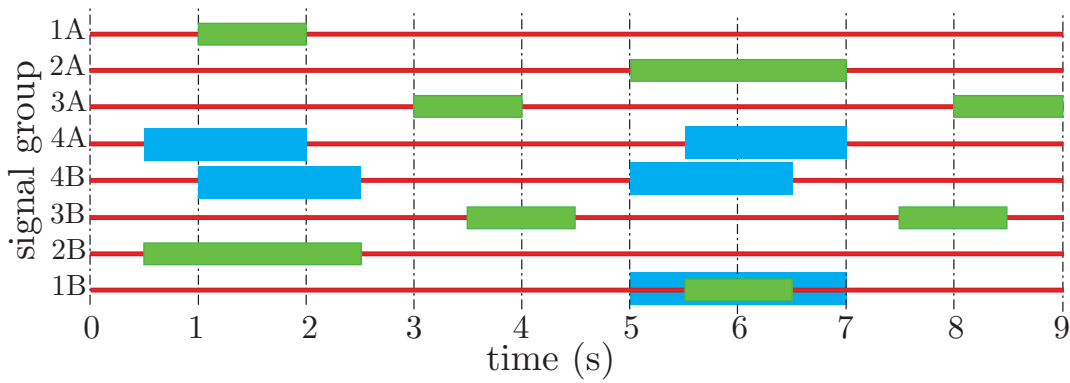


Figure E.5: Partial signal group diagram of the signal groups 1A, 2A, 3A, 1B, 2B, 3B for the introduced theoretical example. In blue we have visualized where signal group 1B is allowed to be green given that signal group 1A is green during the interval $[1, 2]$. Moreover, we have visualized in blue where signal groups 4A and 4B are allowed to be green. The conflict between signal groups 1A and 1B is the only conflict between the signal groups in $\{1A, 2A, 3A\}$ and the signal groups in $\{1B, 2B, 3B\}$. Therefore, we are free to move the timings of the signal groups in $\{1B, 2B, 3B\}$ relative to the timings of the signal groups in $\{1A, 2A, 3A\}$ (as long as the complete green time of signal group 1B stays within the blue interval).

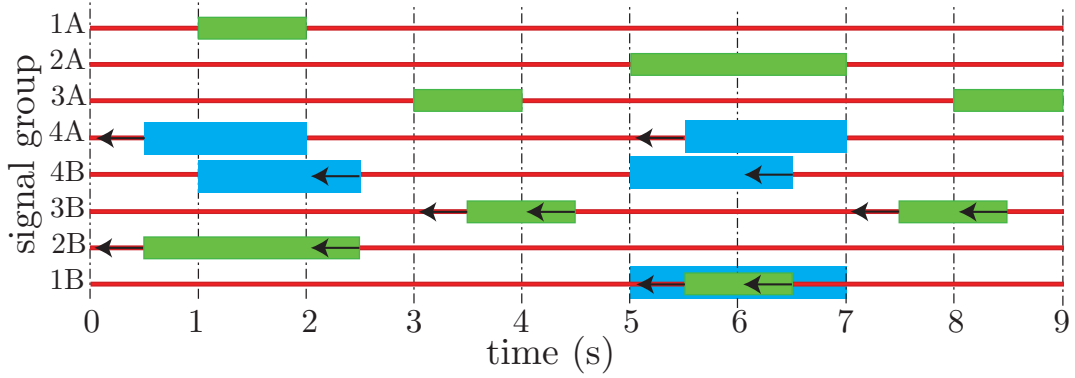
refer to the partial signal group diagram in Figure E.4b (Figure E.4c) as partial diagram A (partial diagram B).

E.3.7 Relative shift of the partial signal group diagrams

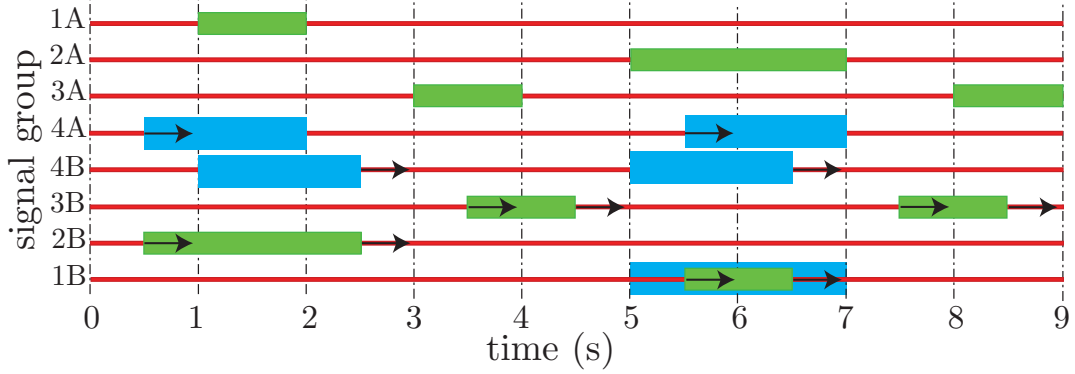
When we combine partial diagram A and partial diagram B, we obtain the partial signal group diagram in Figure E.5. Note that the only conflict that needs to be regarded between these two partial signal groups is the conflict between signal group 1A and signal group 1B; to satisfy the clearance times associated with this conflict we have shifted partial diagram B 4.5 seconds in time (with respect to Figure E.4c). In Figure E.5 we have visualized (in blue) where signal group 1B is allowed to be green (given that signal group 1A is green during the interval $[1, 2]$). Note that we have some room to 'play' with the timing of partial diagram B relative to partial signal group A. Let α denote this relative timing between partial diagram A and partial diagram B. For the signal group diagram in Figure E.5 it holds that this relative timing is zero, i.e., for this signal group diagram we have $\alpha = 0$. If $\alpha > 0$, we shift partial diagram B α seconds forward in time (with respect to partial signal group A) and if $\alpha < 0$, we shift partial diagram B α seconds backwards in time; note that $\alpha \in [-0.5, 0.5]$ because otherwise the minimum clearance times between signal group 1A and signal group 1B are violated.

Signal group 4A and signal group 4B have the same conflicts, which are a conflict with signal group 3A and a conflict with signal group 3B. Signal group 4A and signal group 4B each have to be green for at least three seconds for Prop(1) to be satisfied. In Figure E.5 we have visualized (in blue) where signal group 4A and signal group 4B are allowed to be green when $\alpha = 0$. For $\alpha = 0$ it is (exactly) possible to give both these signal groups enough green time. We prove that it is not possible to satisfy Prop(1) for

any other shift α . Consider the case that $\alpha \in [-0.5, 0)$. In Figure E.6a we visualize the effect of this negative shift on the intervals during which signal group 4A and signal group 4B are allowed to be green. We can see that the total time during which signal group 4B is allowed to be green becomes strictly smaller than 3 seconds for a shift $\alpha \in [-0.5, 0)$. Consider the case that $\alpha \in [0, 0.5)$. In Figure E.6b we visualize the effect of this positive shift on the intervals during which signal group 4A and signal group 4B are allowed to be green. We can see that the total time during which signal group 4A is allowed to be green becomes strictly smaller than 3 seconds for a shift $\alpha \in [0, 0.5)$. This proves that only one shift ($\alpha = 0$) is possible and, therefore, also only one signal group diagram satisfies Prop(1) for the introduced theoretical example.



(a) A negative shift of $\alpha \in (-0.5, 0]$.



(b) A positive shift of $\alpha \in (0, 0.5]$.

Figure E.6: Shift of partial diagram B and its effects on the intervals during which signal group 4A and signal group 4B are allowed to be green (visualized in blue).

Appendices of Chapter 6

F.1 Sustainability of the optimized layout

The optimization problem formulated in Chapter 6 optimizes simultaneously the layout of the intersection and a signal group diagram. During this optimization, also the distribution of traffic amongst the different arrival lanes of the intersection is optimized. The traffic is distributed such that the optimized layout (plus the optimized signal group diagram) can handle all growth factors $\beta < \beta^{\max}$ of these arrival rates; this maximum growth factor β^{\max} is either fixed (for example when finding the smallest intersection that is able to handle all growth factors $\beta < \beta^{\max}$) or is maximized during optimization (for example when finding the layout of the intersection that has the largest capacity). Question however is: can the optimized intersection (plus the optimized signal group diagram) indeed handle all growth factors $\beta < \beta^{\max}$ or do the road users distribute themselves differently amongst the different arrival lanes and, as a result, some growth factor $\beta < \beta^{\max}$ is not sustainable?

In this appendix we motivate that indeed the growth factors $\beta < \beta^{\max}$ are sustainable in reality. In other words, we prove that if the arrival rates (for each of the movements) are scaled with a factor $\beta < \beta^{\max}$ and, in addition, these arrival rates persist over an infinite time horizon, in reality the waiting time at each of the traffic lights remains bounded, i.e., each of the traffic lights at the intersection is *undersaturated*. If the waiting time at a traffic light is not bounded, we call this traffic light *oversaturated*. We assume that the tendency of road users to switch lanes depends on the difference in waiting times between these lanes:

Assumption F.1. *A vehicle switches to an alternative lane (also accommodating the desired movement) if the difference in waiting times (the waiting time at the current lane minus the waiting time at the alternative lane) exceeds some (finite) threshold; this threshold may differ between road users.*

As a consequence of this assumption, road users tend to use the lane with the smallest queue length (when choosing between the lanes that accommodate the desired move-

ment). In particular we can prove that (over an infinite time horizon) the fraction of road users that prefers an *oversaturated* lane (with a waiting time growing to infinity) over an *undersaturated* lane (with a waiting time that remains bounded) is zero. Therefore, some automatic load-balancing-mechanism balances the waiting times of different arrival lanes that provide the same vehicular movement. As a consequence of this load-balancing-mechanism, each traffic light is also undersaturated in reality for any growth factor $\beta < \beta^{\max}$.

Before we give the formal proof, we introduce some notation. Consider the optimization problem formulated in Chapter 6. In this appendix we consider the optimized intersection, the associated optimized signal group diagram, and the optimized distribution of traffic $\lambda \geq 0$ that result from this optimization problem; the vector λ has elements $\lambda_{i,l,j}$, which is the arrival rate at lane l of leg i for lane-use arrow j when the growth factor of the arrival rates equals β^{\max} . Define the scaled distribution of traffic λ^β as $\lambda^\beta := \frac{\beta}{\beta^{\max}} \lambda$, i.e., $\lambda_{i,l,j}^\beta := \frac{\beta}{\beta^{\max}} \lambda_{i,l,j}$; this scaled distribution of traffic corresponds to a growth factor of β , i.e., it satisfies constraints (6.7)–(6.8) with $\beta := \beta$.

Let $\delta^\beta \geq 0$ denote the distribution of traffic amongst the different arrival lanes when the growth factor equals β and the road users may themselves choose which lane to use; the arrival rate $\delta_{i,l,j}^\beta$ is defined as the average amount of traffic arriving per time unit at lane l of leg i for lane-use arrow j when the growth factor equals β . We refer to λ^β as the *optimized distribution of traffic* and we refer to δ^β as the *real distribution of traffic*. We assume that the real distribution of traffic also satisfies constraints (6.7)–(6.8) with $\beta := \beta$. Thus, for both distributions it holds that $\beta \lambda_{i,j}$ is the total arrival rate of traffic that follows lane-use arrow j at leg i :

$$\sum_{l=1}^{\bar{L}_i} \lambda_{i,l,j}^\beta = \sum_{l=1}^{\bar{L}_i} \delta_{i,l,j}^\beta = \beta \lambda_{i,j}, \quad i \in \mathcal{L}, \quad j \in \mathcal{A}_i. \quad (\text{F.1})$$

and if lane l of leg i is not equipped with lane-use arrow j , the corresponding arrival rate equals zero:

$$\lambda_{i,l,j}^\beta = \delta_{i,l,j}^\beta = 0 \text{ when } \Delta_{i,l,j} = 0. \quad (\text{F.2})$$

Recall that each traffic light is undersaturated for the optimized distribution of traffic λ^β , $\beta < \beta^{\max}$. We prove that each traffic light is also undersaturated for the real distribution of traffic δ^β when $\beta < \beta^{\max}$. As a consequence, for each growth factor $\beta < \beta^{\max}$ the waiting time at each of the traffic lights remains bounded in reality. Before proving that each traffic light is undersaturated for the real distribution of traffic δ^β , we first prove some preliminary lemmas.

Lemma F.1. *Consider a growth factor $\beta < \beta^{\max}$. Consider two lanes l and l' of leg i that are both equipped with lane-use arrow j . If lane l is oversaturated then, under Assumption F.1, at least one of the following two statements holds:*

- the arrival rate $\delta_{i,l,j}^\beta$ is zero.

- lane l' of leg i is also oversaturated for this real distribution of traffic.

Proof. Assume that the latter statement is not satisfied: lane l' of leg i is not oversaturated for the real distribution of traffic δ^β . The waiting time at lane l' of leg i then remains bounded whereas the waiting time at lane l of leg i grows to infinity; the difference in the waiting times between these two lanes is then subject to a positive drift. As a consequence, after some finite amount of time the difference in waiting times between these two lanes has become sufficiently large. From that time on, under Assumption F.1, each vehicle that follows lane use arrow j prefers lane l' over lane l . As a consequence, over the infinite time horizon, the (average) arrival rate of traffic following lane-use arrow j at lane l ($\delta_{i,l,j}^\beta$) is zero, which concludes this proof. \square

Corollary F.2. *Consider a growth factor $\beta < \beta^{\max}$. Assume that at least one lane at leg i is oversaturated for this real distribution of traffic; let lane l_{right} be the right most lane of leg i that is oversaturated, i.e., all lanes $l < l_{\text{right}}$ of leg i are undersaturated. Each lane-use arrow j that satisfies $\delta_{i,l_{\text{right}},j}^\beta > 0$ is then not positioned on any lane $l < l_{\text{right}}$ of leg i .*

Proof. Direct consequence of Lemma F.1; otherwise lane l would also be oversaturated, which would contradict the definition of l_{right} . \square

Let L_i be the number of lanes of leg $i \in \mathcal{L}$. We refer to lane $l = 1$ (lane $l = L_i$) as the right most (left most) lane of leg i . Let $l_{\text{right}}(i, j)$ ($l_{\text{left}}(i, j)$), $j \in \mathcal{A}_i$ be the right most (left most) lane of leg i that is equipped with lane-use arrow j . Furthermore, define $j_{\text{right}}(i, l)$ ($j_{\text{left}}(i, l)$) as the lane-use arrow associated with the sharpest turn to the right (left) that is allowed on lane l of leg i .

Lemma F.3. *Consider a growth factor $\beta < \beta^{\max}$. If lane l of leg i is oversaturated and $\delta_{i,l,j_{\text{right}}(i,l)}^\beta < \lambda_{i,l,j_{\text{right}}(i,l)}^\beta$, it must hold that: lane l of leg i accommodates multiple movements, i.e., $j_{\text{right}}(i, l) < j_{\text{left}}(i, l)$, and the sharpest left-turn allowed on lane l of leg i satisfies $\delta_{i,l,j_{\text{left}}(i,l)}^\beta > \lambda_{i,l,j_{\text{left}}(i,l)}^\beta$.*

Proof. Since lane l of leg i is oversaturated for the real distribution δ^β but, by definition, not for the optimized distribution λ^β , it holds that:

$$\sum_{j=j_{\text{right}}(i,l)}^{j_{\text{left}}(i,l)} \frac{\delta_{i,l,j}^\beta}{\mu_{i,l,j}} > \sum_{j=j_{\text{right}}(i,l)}^{j_{\text{left}}(i,l)} \frac{\lambda_{i,l,j}^\beta}{\mu_{i,l,j}}. \quad (\text{F.3})$$

When traffic is distributed amongst the different arrival lanes according to distribution δ^β (λ^β), the left-hand (right-hand) side of this inequality is the minimum fraction of time that lane l of leg i has to be effective green for to ensure stability. Assume that $j_{\text{right}}(i, l) = j_{\text{left}}(i, l)$. Inequality (F.3) then implies $\delta_{i,l,j_{\text{right}}(i,l)}^\beta > \lambda_{i,l,j_{\text{right}}(i,l)}^\beta$, which contradicts the inequality $\delta_{i,l,j_{\text{right}}(i,l)}^\beta < \lambda_{i,l,j_{\text{right}}(i,l)}^\beta$. Hence, it must hold that $j_{\text{right}}(i, l) < j_{\text{left}}(i, l)$.

From the ordering of the lane-use arrows (6.3) it follows that lane-use arrows $j_{\text{right}}(i, l) + 1, \dots, j_{\text{left}}(i, l) - 1$ are all only present at lane l of leg i . Therefore, all traffic following these lane-use arrows must use lane l and, as a result, from (F.1) it follows that:

$$\delta_{i,l,j}^\beta = \lambda_{i,l,j}^\beta = \beta \lambda_{i,j}, \quad j = j_{\text{right}}(i, l) + 1, \dots, j_{\text{left}}(i, l) - 1. \quad (\text{F.4})$$

Combining (F.3) and (F.4) gives:

$$\frac{\delta_{i,l,j_{\text{right}}(i,l)}^\beta}{\mu_{i,l,j_{\text{right}}(i,l)}} + \frac{\delta_{i,l,j_{\text{left}}(i,l)}^\beta}{\mu_{i,l,j_{\text{left}}(i,l)}} > \frac{\lambda_{i,l,j_{\text{right}}(i,l)}^\beta}{\mu_{i,l,j_{\text{right}}(i,l)}} + \frac{\lambda_{i,l,j_{\text{left}}(i,l)}^\beta}{\mu_{i,l,j_{\text{left}}(i,l)}}.$$

Therefore, $\delta_{i,l,j_{\text{right}}(i,l)}^\beta < \lambda_{i,l,j_{\text{right}}(i,l)}^\beta$ indeed implies $\delta_{i,l,j_{\text{left}}(i,l)}^\beta > \lambda_{i,l,j_{\text{left}}(i,l)}^\beta$. \square

Lemma F.4. *Consider a growth factor $\beta < \beta^{\max}$. Furthermore, consider some lane-use arrow $j \in \mathcal{A}_i$ of leg i . If lane $l_{\text{right}}(i, j)$ of leg i is oversaturated and $\delta_{i,l_{\text{right}}(i,j),j}^\beta > \lambda_{i,l_{\text{right}}(i,j),j}^\beta$ then, under Assumption F.1, it must hold that:*

1. *Some other lane of leg i is equipped with lane-use arrow j , i.e., $l_{\text{right}}(i, j) < l_{\text{left}}(i, j)$,*
2. *lane $l_{\text{left}}(i, j)$ is oversaturated for the real distribution of traffic δ^β , and*
3. *$\delta_{i,l_{\text{left}}(i,j),j}^\beta < \lambda_{i,l_{\text{left}}(i,j),j}^\beta$.*

Proof. Assume that $l_{\text{right}}(i, j) = l_{\text{left}}(i, j)$. All traffic following lane-use arrow j must then use lane $l_{\text{right}}(i, j)$ of leg i . Equation (F.1) then implies:

$$\delta_{i,l_{\text{right}}(i,j),j}^\beta = \lambda_{i,l_{\text{right}}(i,j),j}^\beta = \beta \lambda_{i,j},$$

which contradicts the inequality $\delta_{i,l_{\text{right}}(i,j),j}^\beta > \lambda_{i,l_{\text{right}}(i,j),j}^\beta$. Therefore, it must hold that $l_{\text{right}}(i, j) < l_{\text{left}}(i, j)$, which proves the first statement.

From the ordering of the lane-use arrows (6.3) it follows that the lanes $l_{\text{right}}(i, j) + 1, \dots, l_{\text{left}}(i, j) - 1$ are also equipped with lane-use arrow j . The inequality $\delta_{i,l_{\text{right}}(i,j),j}^\beta > \lambda_{i,l_{\text{right}}(i,j),j}^\beta$ together with the non-negativity of the arrival rate $\lambda_{i,l_{\text{right}}(i,j),j}^\beta \geq 0$ implies $\delta_{i,l_{\text{right}}(i,j),j}^\beta > 0$. Since lanes $l = l_{\text{right}}(i, j) + 1, \dots, l_{\text{left}}(i, j)$ are all equipped with lane-use arrow j , it follows from Lemma F.1 that each of these lanes is oversaturated for the real distribution of traffic δ^β . This proves the second statement: lane $l_{\text{left}}(i, j)$ is oversaturated for the real distribution of traffic δ^β .

All traffic that arrives at leg i for lane-use arrow j must be distributed amongst the lanes $l_{\text{right}}(i, j), \dots, l_{\text{left}}(i, j)$. Therefore, (F.1) implies:

$$\sum_{l=l_{\text{right}}(i,j)}^{l_{\text{left}}(i,j)} \delta_{i,l,j}^\beta = \sum_{l=l_{\text{right}}(i,j)}^{l_{\text{left}}(i,j)} \lambda_{i,l,j}^\beta = \beta \lambda_{i,j}. \quad (\text{F.5})$$

Furthermore, it follows from the ordering (6.3) that each lane $l_{\text{right}}(i, j) + 1, \dots, l_{\text{left}}(i, j) - 1$ is equipped with only one lane-use arrow: lane-use arrow j . Since lanes $l_{\text{right}}(i, j) +$

$1, \dots, l_{\text{left}}(i, j) - 1$ are oversaturated for the real distribution of traffic δ^β , but, by definition, not for the optimized distribution of traffic λ^β , this implies that:

$$\delta_{i,l,j}^\beta > \lambda_{i,l,j}^\beta, \quad l = l_{\text{right}}(i, j) + 1, \dots, l_{\text{left}}(i, j) - 1. \quad (\text{F.6})$$

Combining (F.5) and (F.6) gives:

$$\delta_{i,l_{\text{right}}(i,j),j}^\beta + \delta_{i,l_{\text{right}}(i,j),j}^\beta < \lambda_{i,l_{\text{right}}(i,j),j}^\beta + \lambda_{i,l_{\text{right}}(i,j),j}^\beta.$$

Therefore, $\delta_{i,l_{\text{right}}(i,j),j}^\beta > \lambda_{i,l_{\text{right}}(i,j),j}^\beta$ implies $\delta_{i,l_{\text{left}}(i,j),j}^\beta < \lambda_{i,l_{\text{left}}(i,j),j}^\beta$, which proves the third statement and concludes this proof. \square

Lemma F.5. *Consider a growth factor $\beta < \beta^{\max}$. Assume that at least one lane at leg i is oversaturated for this real distribution of traffic; let lane l be the right most lane of leg i that is oversaturated. It then holds that:*

1. $\delta_{i,l,j_{\text{left}}(i,l)}^\beta > \lambda_{i,l,j_{\text{left}}(i,l)}^\beta$, and
2. lane l is the right most lane of leg i that is accommodated with lane-use arrow $j_{\text{left}}(i, l)$.

Proof. We first prove the first statement: $\delta_{i,l,j_{\text{left}}(i,l)}^\beta > \lambda_{i,l,j_{\text{left}}(i,l)}^\beta$. Lane l of leg i is equipped with the lane-use arrows $j_{\text{right}}(i, l), j_{\text{right}}(i, l) + 1, \dots, j_{\text{left}}(i, l)$. Since lane l of leg i is oversaturated for the real distribution δ^β but, by definition, not for the optimized distribution λ^β , it holds that:

$$\sum_{j=j_{\text{right}}(i,l)}^{j_{\text{left}}(i,l)} \frac{\delta_{i,l,j}^\beta}{\mu_{i,l,j}} > \sum_{j=j_{\text{right}}(i,l)}^{j_{\text{left}}(i,l)} \frac{\lambda_{i,l,j}^\beta}{\mu_{i,l,j}}. \quad (\text{F.7})$$

What remains is to prove $\delta_{i,l,j}^\beta \leq \lambda_{i,l,j}^\beta$, $j = j_{\text{right}}(i, l), \dots, j_{\text{left}}(i, l) - 1$; these inequalities would together with (F.7) prove the desired inequality $\delta_{i,l,j_{\text{left}}(i,l)}^\beta > \lambda_{i,l,j_{\text{left}}(i,l)}^\beta$ and prove the first statement. Assume the contrary: $\delta_{i,l,j}^\beta > \lambda_{i,l,j}^\beta$ for some lane-use arrow $j = j_{\text{right}}(i, l), \dots, j_{\text{left}}(i, l) - 1$. From the non-negativity of the arrival rate $\lambda_{i,l,j}^\beta$ it then follows that $\delta_{i,l,j}^\beta > 0$. Therefore, from Corollary F.2 it follows that lane-use arrow j is not positioned on any of the lanes l' on the right of lane l , i.e., the lanes $l' < l$ are not equipped with lane-use arrow j . Furthermore, from the ordering of the lane-use arrows (6.3) it follows that lane-use arrow j is also not positioned on any lane $l' > l$. Thus, lane-use arrow j is only positioned on lane l and therefore all traffic that follows lane-use arrow j at leg i must use lane l . Thus, $\delta_{i,l,j}^\beta = \lambda_{i,l,j}^\beta = \beta \lambda_{i,j}$, which contradicts the inequality $\delta_{i,l,j}^\beta > \lambda_{i,l,j}^\beta$. This concludes the proof of the first statement. The second statement follows directly from corollary F.2. \square

We are now ready to prove the main result:

Theorem F.6. *Consider a growth factor $\beta < \beta^{\max}$. Under Assumption F.1, each traffic light is undersaturated for the real distribution of traffic δ^β . Therefore, also in reality this growth factor β is sustainable.*

Proof. Assume the contrary: some leg i has a lane that is oversaturated for the real distribution of traffic δ^β . Define the sequence of lanes l_k , $k = 1, 2, \dots$ as follows. The lane l_0 is the right most lane of leg i that is oversaturated. Furthermore, let j_k , $k = 1, 2, \dots$ be the sharpest left turn that is allowed on lane l_k of leg i , i.e., $l_k := j_{\text{left}}(i, l_k)$. Furthermore, let lane l_{k+1} , $k = 1, 2, \dots$ be the left most lane of leg i that is equipped with the lane-use arrow j_k . In other words, $l_{k+1} := l_{\text{left}}(i, j_k)$. We can prove that l_k , $k = 1, 2, \dots$ is a strictly increasing sequence, which contradicts the finiteness of the number of lanes at leg i and proves this theorem.

We prove via induction that l_k is strictly increasing in k . Consider the following induction hypothesis (IH(k)):

1. lane l_k is oversaturated for the real distribution of traffic δ^β ,
2. $\delta_{i, l_k, j_k}^\beta > \lambda_{i, l_k, j_k}^\beta$, and
3. lane l_k is the right most lane equipped with lane use arrow j_k .

By definition lane l_0 is the right most lane that is oversaturated and, therefore, from Lemma F.5 (with $l := l_0$ and $j_{\text{left}}(i, l) := j_0$) it follows that the induction hypothesis IH(0) is satisfied. Using the induction hypothesis IH(k) we prove the induction step: we prove IH($k + 1$) and the strict inequality $l_k < l_{k+1}$.

From the induction hypothesis it follows that lane l_k is the right most lane that is equipped with lane-use arrow j_k , i.e., $l_{\text{right}}(i, j_k) := l_k$. Moreover, by definition, lane l_{k+1} is the left most lane that is equipped with lane-use arrow j_k , i.e., $l_{\text{left}}(i, j_k) := l_{k+1}$. Hence, from the ordering of the lane-use arrows (6.3), it follows that (only) the lanes $l_k, l_k + 1, \dots, l_{k+1}$ are equipped with lane-use arrow j_k . Using the induction hypothesis it follows from Lemma F.4 (with $j := j_k$, $l_{\text{right}}(i, j) := l_k$, and $l_{\text{left}}(i, j) := l_{k+1}$) that: $l_k < l_{k+1}$, lane l_{k+1} is oversaturated, and $\delta_{i, l_{k+1}, j_k}^\beta < \lambda_{i, l_{k+1}, j_k}^\beta$. So far we have proved the strict inequality $l_k < l_{k+1}$, and the first statement of IH($k + 1$).

Since, lane l_k and lane l_{k+1} are both equipped with lane-use arrow j_k it follows from the ordering of the lane-use arrows (6.3) that lane-use arrow j_k is the sharpest right turn allowed on lane l_{k+1} , i.e., $j_{\text{right}}(i, l_{k+1}) := j_k$. Moreover, by definition it holds that lane-use arrow j_{k+1} is the sharpest left turn allowed on lane l_{k+1} , i.e., $j_{\text{left}}(i, l_{k+1}) := j_{k+1}$. From the ordering of the lane-use arrows (6.3) it follows that lane l_{k+1} is (only) equipped with lane-use arrows $j_k, j_k + 1, \dots, j_{k+1}$. Therefore, we can apply Lemma F.3 (with $l := l_{k+1}$, $j_{\text{right}}(i, l) := j_k$, and $j_{\text{left}}(i, l) := j_{k+1}$) to find the inequality $\delta_{i, l_{k+1}, j_{k+1}}^\beta > \lambda_{i, l_{k+1}, j_{k+1}}^\beta$ and $j_{k+1} > j_k$; this proves the second statement of IH($k + 1$). Lane l_{k+1} is equipped with multiple lane-use arrows, which are the lane-use arrows $j_k, j_k + 1, \dots, j_{k+1}$. From the ordering of the lane-use arrows (6.3) it follows that lane l_{k+1} is the right most lane that

is equipped with lane use arrow j_{k+1} . This proves the third statement of IH($k + 1$). This proves that l_k , $k = 1, 2, \dots$ is a strictly increasing sequence, which contradicts the finiteness of the number of lanes at leg i and proves this theorem. \square

F.2 Optimizing intersection layout: exclusive bus lanes

Public transport busses often have to meet tight time schedules. Signalized intersections can cause large and uncertain waiting times that make it impossible to meet these tight schedules. To reduce this waiting time and uncertainty, a signalized intersection can be equipped with exclusive bus lanes. These exclusive bus lanes may also be desired as a means of promoting public transportation. In this appendix we indicate how to optimize the intersection layout with such exclusive bus lanes. To this end, we use the optimization problem of Chapter 6.

F.2.1 Exclusive arrival lane for busses

Let an *arrival bus lane* be an arrival lane intended for the exclusive use of busses. Similarly, we use *departure bus lane* to refer to a departure lane intended for the exclusive use of busses. It is possible that all busses that arrive at an arrival bus lane must depart the intersection via a certain departure bus lane. It is also possible that the busses that arrive at an arrival bus lane depart the intersection via a 'regular' departure lane. Let a_i^{bus} (e_i^{bus}) denote the number of arrival (departure) bus lanes that is required at leg $i \in \mathcal{L}$. For each leg $i \in \mathcal{L}$, we number the arrival bus lanes from 1 until a_i^{bus} . Besides the busses that arrive at the arrival bus lanes of leg i , several vehicular movements arrive at leg $i \in \mathcal{L}$. Each of these vehicular movements is associated with a lane-use arrow $j \in \mathcal{A}_i$. Recall that these lane-use arrows are numbered according to their angle α (see Figure 6.2 for this angle α). In other words, lane-use arrow 1 corresponds to the sharpest right turn that is allowed on leg i and lane-use arrow $|\mathcal{A}_i|$ corresponds to the sharpest left turn that is allowed on leg i . We assume that all arrival lanes (including the arrival bus lanes) of leg i are positioned on the right of all departure lanes (including the departure bus lanes). For each of the arrival bus lanes $l_b = 1, \dots, a_i^{\text{bus}}$, of leg $i \in \mathcal{L}$ we require its desired position $J(i, l_b)$, which is defined as follows (see also Figure F.1). Arrival bus lane $l_b = 1, \dots, a_i^{\text{bus}}$ must be positioned on the left (when facing towards the intersection) of each of the arrival lanes that is equipped with a lane-use arrow $j = 1, \dots, J(i, l_b)$ and on the right of each arrival lane that is equipped with a lane-use arrow $j = J(i, l_b) + 1, \dots, |\mathcal{A}_i|$. Note that the positions in Figure F.1a and Figure F.1d are the most common ones in practice.

Consider the optimization problem proposed in Chapter 6; for this optimization problem we disregard the arrival bus lanes and the departure bus lanes completely. Consider the following assumptions.

Assumption F.2. For each arrival bus lane $l_b = 1, \dots, a_i^{\text{bus}}$ of leg $i \in \mathcal{L}$ with $0 <$

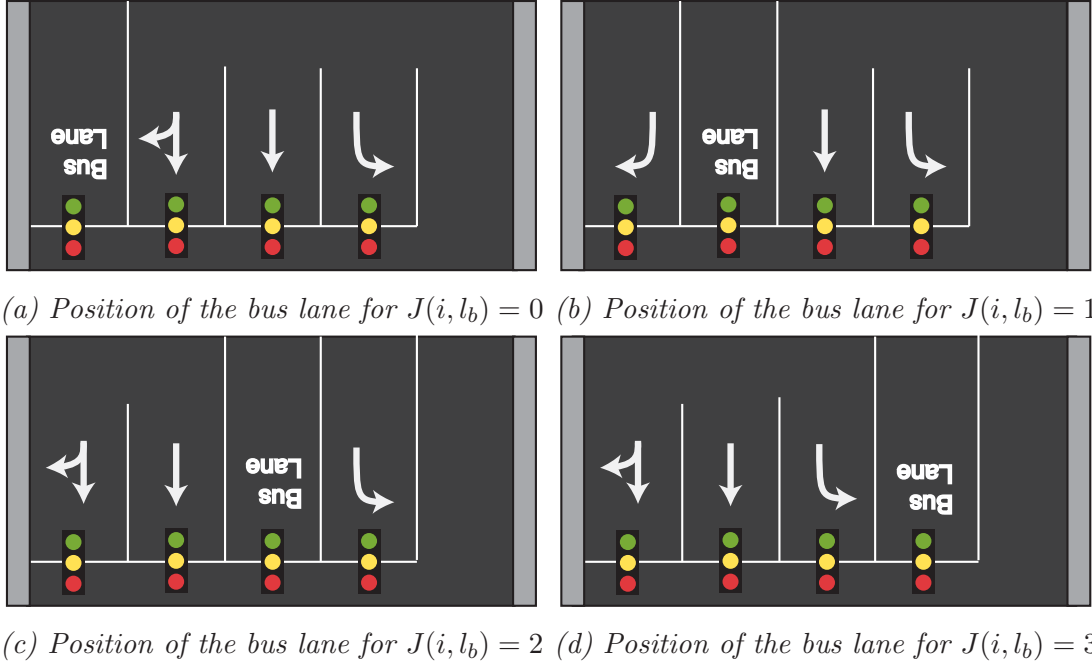


Figure F.1: Different positions $J(i, l_b)$ of exclusive arrival bus lane l_b of leg i . The vehicular movements of leg i are $\mathcal{A}_i = \{1, 2, 3\}$.

$J(i, l_b) < |\mathcal{A}_i|$, no lane exists that is equipped with both lane-use arrow $J(i, l_b)$ and lane-use arrow $J(i, l_b) + 1$. Thus, for the optimized layout it holds that $\Delta_{i,l,J(i,l_b)} + \Delta_{i,l,J(i,l_b)+1} \leq 1$ for each such arrival bus lane.

Consider an arrival bus lane $l_b = 1, \dots, a_i^{\text{bus}}$ of leg $i \in \mathcal{L}$ with $0 < J(i, l_b) < |\mathcal{A}_i|$. Let l' be the right most lane of leg i that is equipped with lane-use arrow $J(i, l_b)$. Under Assumption F.2, it follows from the ordering of these lane-use arrows (6.3) that the lane-use arrows $j \leq J(i, l_b)$ are positioned on the lanes $l \leq l'$ of leg i and that the lane-use arrows $j \geq J(i, l_b) + 1$ are positioned on the lanes $l \geq l' + 1$. Therefore, it is possible to include arrival bus lane l_b at its desired position, which is between lane l' and lane $l' + 1$. Note that it is also possible to include each arrival bus lane $l_b = 1, \dots, a_i^{\text{bus}}$ of leg $i \in \mathcal{L}$ with $J(i, l_b) \in \{0, |\mathcal{A}_i|\}$ at its desired position. One possible obstacle is the maximum number of lanes of each leg $i \in \mathcal{L}$.

Assumption F.3. For each leg $i \in \mathcal{L}$, the number of lanes $\mathbf{a}_i + \mathbf{e}_i$ satisfies the following inequality $\mathbf{a}_i + \mathbf{e}_i + a_i^{\text{bus}} + e_i^{\text{bus}} \leq \bar{L}_i$.

Under Assumption F.3, the a_i^{bus} arrival bus lanes and the e_i^{bus} departure bus lanes can be added to leg i without exceeding the maximum number of lanes \bar{L}_i .

When both Assumption F.2 and Assumption F.3 hold, we can add each of the arrival bus lanes and each of the departure bus lanes to the intersection at its desired positions. We can ensure that we can add the exclusive bus lanes to the intersection after we have solved the MIP problem of Chapter 6 by including the following constraint for each arrival

bus lane $l_b = 1, \dots, a_i^{\text{bus}}$ of leg $i \in \mathcal{L}$ with $0 < J(i, l_b) < |\mathcal{A}_i|$:

$$\Delta_{i,l,J(i,l_b)} + \Delta_{i,l,J(i,l_b)+1} \leq 1,$$

and adding the following constraint for each leg $i \in \mathcal{L}$:

$$\mathbf{a}_i + \mathbf{e}_i + a_i^{\text{bus}} + e_i^{\text{bus}} \leq \bar{L}_i.$$

These constraints guarantee that for each arrival bus lane $l_b = 1, \dots, a_i^{\text{bus}}$ of leg $i \in \mathcal{L}$ with $0 < J(i, l_b) < |\mathcal{A}_i|$ the lane-use arrows $J(i, l_b)$ and $J(i, l_b) + 1$ are not positioned on the same lane and that we can add a_i^{bus} arrival bus lanes and e_i^{bus} departure bus lanes to leg i without exceeding the maximum number of lanes \bar{L}_i .

F.2.2 Optimizing the green, yellow and red times of the bus lanes

To ensure that each arrival bus lane receives enough green light, we can keep track of one additional traffic light for each arrival bus lane during optimization; we then also optimize the green, yellow and red times for the arrival bus lanes. Note that the position $J(i, l_b)$ of an arrival bus lane l_b of leg i determines which of the traffic movements conflict with this arrival bus lane. As this position $J(i, l_b)$ is given, these conflicts are known before optimization. Therefore, we can optimize the green, yellow and red times of these additional traffic lights in exactly the same manner as we do for any of the pedestrian and cyclist movements of the intersection.

Remark F.1. *Consider an intersection for which some lanes are intended for the exclusive use of busses. In practice, such an intersection is often controlled with an actuated controller. When the arrival rate of busses at a exclusive bus lane is relatively small, this actuated controller may be based on a signal group diagram for which this arrival bus lane has no realization. Whenever a bus arrives at this bus lane, the 'normal' program is interrupted and the bus lane receives a green light as soon as possible. When the bus has crossed the intersection, the 'normal' program is again resumed. The exclusive bus lane does not receive any realizations during the 'normal program'. Therefore, when designing this 'normal program', it may be desired to optimize a signal group diagram that does not specify the green, yellow and red times of this bus lane. If the arrival rate of busses at some exclusive bus lane is large, it is sensible to also optimize the green, yellow and red times of this bus lane. The green intervals of this bus lane are then included in the 'normal' program of the actuated controller.*

F.3 Advanced layout optimization

In Chapter 6 we have proposed an optimization problem to optimize the layout of an intersection. This optimization problem assumes that the set of conflicting traffic movements $\Psi_{\mathcal{M}}$ is fixed and given. Hence, we cannot use this optimization problem to optimize

over layout variants as the one in Figure F.2. For this layout variant leg 3 has two departure lanes. The right-turn movement of leg 1 (movement 1) is guided towards the right most departure lane, while the through movement of leg 2 (movement 5) is guided towards the left most departure lane. As a consequence, movement 1 and movement 5 are not conflicting for this layout variant; it is safe for the right-turn movement to receive a green indication simultaneous with the through movement of leg 3. However, would leg 3 only have had one departure lane, then movements 1 and 5 would have to use the same departure lane and these two movements are then considered to be conflicting. In this appendix we show that we can modify the optimization problem proposed in Chapter 6 so that we can also optimize over layout variants like the one in Figure 6.7. We do so by using two examples: a T-junction and an intersection with four legs. First, in Section F.3.1 we consider the T-junction. Thereupon, in Section F.3.2 we consider an intersection with four legs.

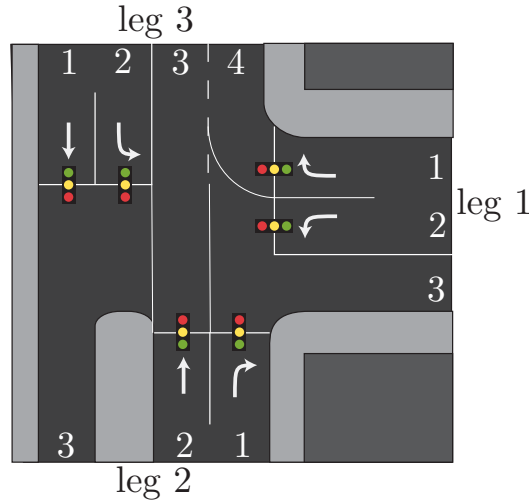


Figure F.2: A layout variant of a T-junction. For this layout variant the right-turn movement of leg 1 may be green simultaneously with the through movement of leg 2; leg 3 has sufficient lanes to allow these traffic movements simultaneously in a safe manner.

F.3.1 T-junction

In this section we consider the T-junction visualized in Figure F.2. We modify the optimization problem proposed in Chapter 6 so that movement 1 and movement 5 are considered to be conflicting only when the departure lanes of leg 3 do not have sufficient departure lanes to accommodate these movements at the same time. Note that the number of lanes that accommodate movement 1 equals $\sum_{l \in Q_{i(1)}^v} \Delta_{i(1),l,j(1)}$ and the number of lanes that accommodate movement 5 equals $\sum_{l \in Q_{i(5)}^v} \Delta_{i(5),l,j(5)}$. Therefore, movement 1 and movement 5 should only be conflicting whenever:

$$\sum_{l \in Q_{i(1)}^v} \Delta_{i(1),l,j(1)} + \sum_{l \in Q_{i(5)}^v} \Delta_{i(5),l,j(5)} > e_3. \quad (\text{F.8})$$

We modify the constraints of the optimization problem given in Chapter 6 accordingly. To this end, we introduce a binary-valued design variable $\mathbf{b}_{1,5}^{\text{sim}}$. This binary-valued design variable equals one whenever movement 1 and movement 5 may be green simultaneously (they are then not considered to be conflicting) and this binary-valued design variable equals zero otherwise. With the following linear constraint we force the binary-valued design variable $\mathbf{b}_{1,5}^{\text{sim}}$ to equal zero whenever (F.8) holds:

$$\mathbf{b}_{1,5}^{\text{sim}} \leq 1 - \frac{\sum_{l \in \mathcal{Q}_{i(1)}^v} \Delta_{i(1),l,j(1)} + \sum_{l \in \mathcal{Q}_{i(5)}^v} \Delta_{i(5),l,j(5)} - \mathbf{e}_3}{\bar{L}_3}. \quad (\text{F.9})$$

To verify this equation, recall that the constraint (6.5) ensures the following inequalities:

$$\begin{aligned} \sum_{l \in \mathcal{Q}_{i(1)}^v} \Delta_{i(1),l,j(1)} &\leq \mathbf{e}_3, \\ \sum_{l \in \mathcal{Q}_{i(5)}^v} \Delta_{i(5),l,j(5)} &\leq \mathbf{e}_3. \end{aligned}$$

Furthermore, constraint (6.6) ensures that the number of departure lanes of leg 3 (\mathbf{e}_3) does not exceed the maximum number of lanes that is allowed at leg 3 (\bar{L}_3). As a consequence, it holds that the value of the right-hand side of (F.9) is included in the interval $[0, 2]$. Moreover, this right-hand side is (strictly) smaller than one if and only if the inequality (F.8) is satisfied. As a result, the constraint (F.9) forces the binary-valued design variable $\mathbf{b}_{1,5}^{\text{sim}}$ to equal zero if and only if the inequality (F.8) is satisfied.

We use the binary-valued design variable $\mathbf{b}_{1,5}^{\text{sim}}$ to force movement 1 and movement 5 to be conflicting (only) when (F.9) is satisfied. We do so by making the constraints on the associated clearance times (6.20) redundant whenever $\mathbf{b}_{1,5}^{\text{sim}} = 1$; these associated clearance times are the clearance times between a realization $k_1 \in \bar{\mathcal{K}}_1$ of movement 1 and a realization $k_5 \in \bar{\mathcal{K}}_5$ of movement 5. We can do so by adding a term $\mathbf{b}_{1,5}^{\text{sim}} L$ to the right-hand side of these constraints, where L is some large and positive number.

F.3.2 An intersection with four legs

In this section we consider an intersection with four legs, see also Figure F.3. The movements that depart the intersection at leg 4 are: the right-turn movement of leg 1 (movement 1), the through movement of leg 2 (movement 5) and the left-turn movement of leg 3 (movement 9). In the same manner as explained in the previous section we can force each two movements $m_1 \in \{1, 5, 9\}$ and $m_2 \in \{1, 5, 9\}$ to be conflicting whenever leg 4 does not have sufficient departure lanes to allow these two movements at the same time. To this end, we use the following three binary-valued design variables: $\mathbf{b}_{1,5}^{\text{sim}}$, $\mathbf{b}_{1,9}^{\text{sim}}$ and $\mathbf{b}_{5,9}^{\text{sim}}$. See also Figure F.3 for the effect that these binary-valued design variables have on the layout of the intersection.

Theoretically it is also possible to allow the three movements 1, 5 and 9 to depart simultaneously at leg 4, see Figure F.3d. However, this is only allowed whenever the

number of departure lanes at leg 4 exceeds (or equals) the number of arrival lanes that accommodate movement 1, movement 5 or movement 9. With a similar reasoning as in the previous section we include the following constraint:

$$\mathbf{b}_{1,5}^{\text{sim}} + \mathbf{b}_{1,9}^{\text{sim}} + \mathbf{b}_{5,9}^{\text{sim}} \leq 3 - \frac{\sum_{m \in \{1,5,9\}} \sum_{l \in \mathcal{Q}_{i(m)}^v} \Delta_{i(m),l,j(m)} - e_4}{\bar{L}_4}.$$

The right-hand side of this equation is included in the interval $[0, 3]$. Moreover, this right-hand side is strictly smaller than two if and only if the number of departure lanes of leg 4 cannot accommodate the movements 1, 5 and 9 at the same time. This concludes this appendix.

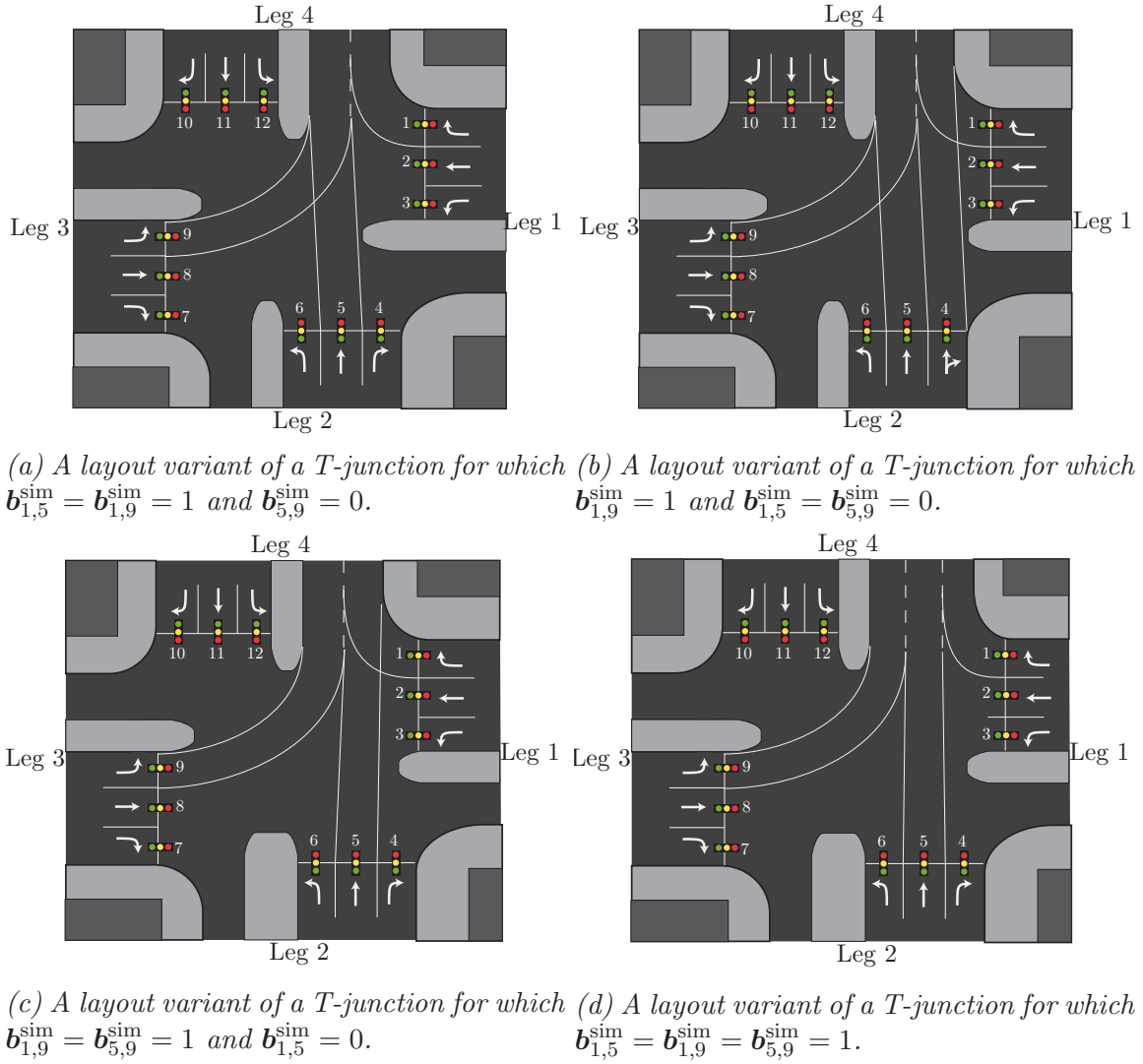
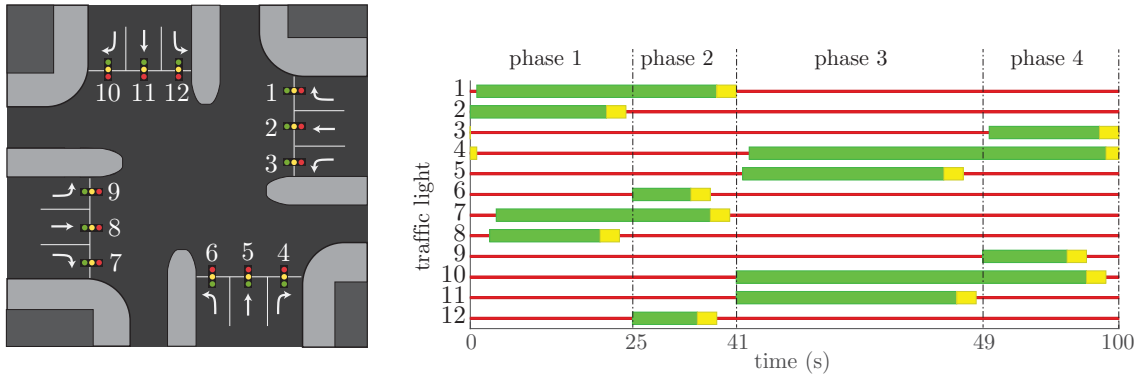


Figure F.3: Layout variants that are associated with different values for the binary-valued design variables $\mathbf{b}_{1,5}^{\text{sim}}, \mathbf{b}_{1,9}^{\text{sim}}, \mathbf{b}_{5,9}^{\text{sim}}$.

Summary

Optimizing pre-timed control at isolated intersections

Today's society has a large demand for road transport, which causes congestions in the road network. The consequences are serious: congestion increases the time we spend on the road, it affects our daily living, the quality of the air we breathe, the price tag on the products we buy, the costs of the trips we make, et cetera. In this thesis we develop several methods to take more efficient use of currently existing infrastructure and, as a result, these methods may also contribute to the mitigation of the congestion problem. To be more specific: this thesis concerns the optimization of pre-timed traffic light control at isolated intersections.



An intersection (left) and a signal group diagram (right).

For a pre-timed controller the green, yellow and red intervals are timed periodically. Such a controller can be visualized in a signal group diagram (see the figure above). In contrast to an actuated controller, the pre-timed controller does not use any detector information to control the traffic lights. Below we give a motivation for the optimization of pre-timed control

- Optimizing a pre-timed controller is usually the first step when designing an actuated controller; the traffic lights switch to green, yellow and red in the same order as

the pre-timed controller, while the duration of these intervals depend on the current traffic situation.

- An actuated controller may behave as a pre-timed controller in some situations, e.g., when the amount of traffic arriving at the intersection is large. Such situations call for the design of a pre-timed controller.
- By optimizing a pre-timed controller (for an isolated intersection) we can check whether the intersection can handle the amount of traffic that is forecasted to arrive at it, i.e., we can check whether the intersection has enough capacity. This may also be used when designing an intersection, e.g., seeking for the intersection layout with the largest capacity.
- The 'predictability' of pre-timed control makes it easier to synchronize different intersections. As a consequence, the delay that road users experience can be reduced by synchronising the pre-timed controllers of the different intersections, which may create so called green waves.
- This predictability can also be used by 'smart' cars. For example by visualizing the future state of the traffic light. The road user can use this information to adjust its speed, reduce its waiting time and save fuel. Furthermore, this predictability can be used to compute a smart route through the city. This computation also includes the waiting time caused by traffic lights.

This thesis is structured as follows. First, in Chapter 2 we consider the mathematical modeling of traffic lights and the additional travel time (delay) that these traffic lights induce on road users.

This mathematical model is used in the subsequent chapters to optimize pre-timed controllers. First, in Chapter 3 we assume that each traffic light has a single green interval during one repeating period (see for example the signal group diagram given at the start of this summary). We optimize simultaneously: the period duration of the signal group diagram, when the green intervals start, and when they end. Possible objective functions are the maximization of the capacity of the intersection (i.e., search for the signal group diagram that can handle the largest increase in the arrival rates) and the minimization of the average delay that road users experience at the intersection. The proposed optimization problem is a 'mixed-integer programming problem'. The integral design variables of this optimization problem aggregate the binary variables of many known formulations. Based on an extensive numerical study, we conclude that this new formulation is superior to currently existing formulations. In other words, the needed computation time is much smaller for the novel formulation.

The proposed formulation is extended in Chapter 4 to also optimize the number of green intervals that each traffic light has. To this end, we use additional binary variables. Each of these binary variables switches 'on' or 'off' a specific green interval. To our

knowledge, this is the first formulation that also optimizes the number of green intervals of each traffic light. This allows us to optimize over a larger set of signal group diagrams and, as a consequence, find better signal group diagrams. From a numerical study we conclude that by also optimizing the number of green intervals, we are often able to reduce the average delay that road users experience by more than 10 percent.

In this thesis we also consider two extensions of the proposed optimization problem. In Chapter 5, we use the proposed formulation to optimize integral signal group diagrams. Each traffic light switches to green, yellow and red at an integral second for such integral diagrams. These integral schedules are desired in practice as they are clear, presentable and easy to work with. The proposed method consists of two steps. In the first step we optimize over the signal group diagrams that satisfy some structural property (which is also satisfied by each integral signal group diagram). The goal of this first step is not to find an integral signal group diagram yet, but to find a signal group diagram that can easily be 'rounded' in the second step. During the second step we round the signal group diagram obtained in the first step, by solving an optimization problem (mixed-integer programming problem). The result is the desired integral signal group diagram.

In Chapter 6 we consider an extension that allows us to also optimize the layout of the intersection, e.g., how many lanes does the intersection need, which of these lanes are arrival lanes, which lane-use arrows are marked on each of the lanes, et cetera. We can, for example, use this optimization formulation to answer the following questions:

- How should we change the lane-use arrows such that the capacity of the intersection is maximized?
- What is the smallest intersection that has sufficient capacity?
- Which possible intersection has the largest capacity?

To answer these questions, the layout of the intersection has to be optimized simultaneously with a signal group diagram.

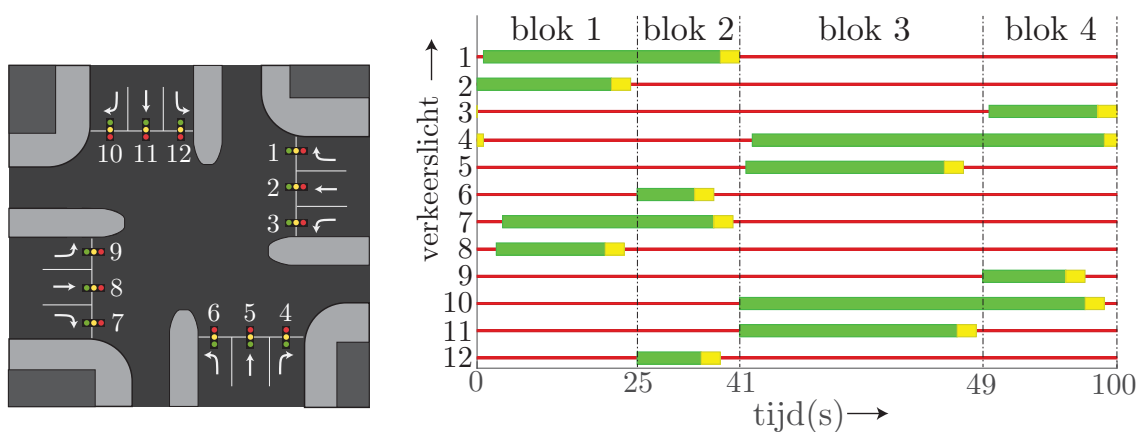
Subsequently, in Chapter 7 we consider some practical issues that were not yet addressed in the previous chapters and, finally, in Chapter 8 we give our conclusions and recommendations.

With the different chapters of this thesis, we give an insightful and comprehensive overview of the optimization of pre-timed control and its usefulness. The methods proposed in this thesis can be used to aid traffic engineers in their process of designing a (pre-timed) traffic light controller (as well as their process of designing intersection layout).

Nederlandse samenvatting

Vandaag de dag is er een grote vraag naar wegtransport. Dit resulteert in opstoppingen in het verkeersnetwerk en heeft grote negatieve gevolgen: het beïnvloedt ons dagelijks leven, zorgt voor grotere reistijden en hogere reiskosten, beïnvloedt de kwaliteit van de lucht die we inademen, resulteert in hogere consumentenprijzen, et cetera. Met dit proefschrift leveren wij een bijdrage aan een efficiënter gebruik van bestaande infrastructuur en, als gevolg, ook aan het verminderen van opstoppingen in het verkeersnetwerk. Om wat specifieker te zijn: we beschouwen de optimalisatie van starre verkeerslichten regelingen voor geïsoleerde kruispunten.

Zo'n starre regeling specificeert wanneer ieder verkeerslicht groen, oranje en rood is gedurende een herhalende periode en wordt vaak weergegeven in een fasediagram (zie het figuur hieronder). In tegenstelling tot een voertuigafhankelijke regeling maakt de starre regeling geen gebruik van detector informatie.



Een kruispunt (links) en een fasediagram (rechts).

Hieronder volgt een motivatie voor het optimaliseren van starre regelingen:

- Het optimaliseren van een starre regeling is normaliter de eerste stap in het ontwerp van een voertuigafhankelijke regeling. De verkeerslichten worden dan groen, oranje en rood in dezelfde volgorde als de starre regeling, terwijl de duur van deze perioden afhangt van de verkeerssituatie.

- De 'voorspelbaarheid' van starre regelingen maakt het gemakkelijker om verschillende kruispunten op elkaar af te stemmen. Dit maakt het bijvoorbeeld mogelijk om groene golven te creëren.
- Dezelfde voorspelbaarheid kan ook worden gebruikt door 'slimme' auto's. Bijvoorbeeld door de (toekomstige) status van het verkeerslicht te visualiseren, wat gebruikt kan worden om de snelheid aan te passen, de wachttijd voor verkeerslichten te reduceren en brandstof te besparen. Deze voorspelbaarheid kan ook gebruikt worden om een slimme route te berekenen door een netwerk van kruispunten. In deze berekening wordt ook de wachttijd bij de verkeerslichten meegenomen.
- Een voertuigafhankelijke regeling kan zich in sommige situaties gedragen als een starre regeling. Bijvoorbeeld wanneer de aankomstintensiteiten (hoeveelheid verkeer die aankomt bij het kruispunt) hoog zijn. Deze situaties roepen om de optimalisatie van starre regelingen.
- Door het optimaliseren van een starre regeling kan de capaciteit van een kruispunt worden bepaald. We kunnen bijvoorbeeld antwoord geven op de volgende vraag: hoelang kan het huidige kruispunt nog mee?

Het proefschrift is als volgt opgebouwd. In Hoofdstuk 2 wordt uitgelegd hoe verkeerslichten en de wachttijd die weggebruikers ondervinden bij deze verkeerslichten wiskundig gemodeleerd kunnen worden.

Deze wiskundige modelering wordt in de daaropvolgende hoofdstukken gebruikt om fasediagrammen te optimaliseren. In Hoofdstuk 3 wordt aangenomen dat ieder verkeerslicht maar een enkele keer groen is tijdens de herhalende periode (zie bijvoorbeeld het fasediagram aan het begin van deze samenvatting). We optimaliseren tegelijkertijd: de periodetijd van het fasediagram, wanneer de groen perioden starten en wanneer ze eindigen. Mogelijke doelfuncties zijn bijvoorbeeld het maximaliseren van de capaciteit van het kruispunt (vind het fasediagram die een zo groot mogelijke stijging in de aankomstintensiteiten aan kan) en het minimaliseren van de wachttijd die weggebruikers ondervinden bij het kruispunt. De voorgestelde formulering is een 'mixed-integer programming problem'. De integrale variabelen van dit optimalisatie probleem aggregeren de binaire variabelen die gebruikt worden in vele bestaande formuleringen. Gebaseerd op een uitgebreide numerieke studie concluderen we dat de nieuwe formulering superieur is ten opzichte van andere optimalisatie formuleringen. In andere woorden, met de nieuwe formulering heeft de computer veel minder tijd nodig om het optimale fasediagram te vinden.

In Hoofdstuk 4 breiden we de formulering uit zodat ook het aantal groen periode van ieder stoplicht kan worden geoptimaliseerd. Hiervoor worden binaire variabelen gebruikt. Iedere binaire variabele wordt gebruikt om een specifieke groen periode 'aan' of 'uit' te zetten. Voor zover wij weten is dit de eerste formulering die ook het aantal groen perioden van ieder verkeerslicht optimaliseert. Hierdoor is de nieuwe methode in staat om betere fasediagrammen te vinden dan andere bestaande methoden. Uit een numerieke studie

blijkt dat het optimaliseren van het aantal groen periode vaak leidt tot een vermindering van meer dan 10 procent in de wachttijd die weggebruikers bij het kruispunt ondervinden.

In dit proefschrift worden ook twee uitbreidingen van het voorgestelde optimalisatie probleem beschouwd. In Hoofdstuk 5 gebruiken we de voorgestelde formulering om integrale fasediagrammen te optimaliseren. Voor zo'n diagram start en eindigt iedere groen, oranje en rood periode op een gehele seconde. Deze integrale schema's zijn gewenst in de praktijk omdat ze overzichtelijk zijn, en daardoor zijn ze ook gemakkelijk om mee te werken. De voorgestelde methode bestaat uit twee stappen. In de eerste stap optimaliseren we alleen over de fasediagrammen die voldoen aan een bepaalde eigenschap (waar ook ieder integraal fasediagram aan voldoet). Het doel van deze eerste stap is nog niet om een integraal fasediagram te vinden, maar om een fasediagram te vinden die gemakkelijk kan worden 'afgerond' in de tweede stap. In de tweede stap wordt een optimalisatie probleem (mixed-integer programming problem) opgelost waarmee het fasediagram wordt afgerond; het resultaat is een integraal fasediagram.

In Hoofdstuk 6 beschouwen we een uitbreiding die het mogelijk maakt om ook de layout van een kruispunt te optimaliseren. Onder de layout van het kruispunt verstaan we de volgende gegevens: hoeveel rijstroken heeft het kruispunt, welke rijstroken leiden verkeer naar het kruispunt toe, welke rijstroken voeren verkeer van het kruispunt weg en welke wegmarkering (pijlen op de weg die aangeven in welke richting je het kruispunt mag verlaten) staan gemarkeerd op iedere rijstrook. Met deze optimalisatie kunnen we bijvoorbeeld een antwoord geven op de volgende vragen:

- Hoe moet de wegmarkering veranderen zodat het kruispunt zo veel mogelijk capaciteit heeft?
- Wat is het kleinste kruispunt met voldoende capaciteit?
- Welk mogelijke kruispunt heeft de grootste capaciteit?

Om deze vragen te beantwoorden moet de geometrie van het kruispunt tegelijkertijd worden geoptimaliseerd met een fasediagram.

In Hoofdstuk 7 beschouwen we een aantal kwesties die van belang zijn in de praktijk, maar die nog niet zijn behandeld in de voorgaande hoofdstukken en in hoofdstuk 8 geven we onze conclusies en aanbevelingen.

Met de verschillende hoofdstukken geeft dit proefschrift een inzichtelijk en uitgebreid overzicht van de optimalisatie van starre regelingen en het nut hiervan. De methodes die voorgesteld zijn in dit proefschrift sluiten direct aan bij de huidige werkwijze die wordt gebruikt in de praktijk en ze kunnen gebruikt worden om verkeersingenieurs te ondersteunen bij het ontwerpen van verkeerlichtenregelingen (en kruispunten).

Curriculum Vitae



Stijn Fleuren was born on July 20th, 1989 in Nijmegen, the Netherlands. After finishing his secondary education at Scholengemeenschap Stevensbeek, which was later renamed Metameer, he started his studies in Mechanical Engineering at the Technical University of Eindhoven (TU/e). Five years later, in 2012, he received his master degree with the distinction 'cum laude' after writing his thesis entitled: 'Optimization traffic light control - A control theory approach'.

Immediately after obtaining his masters degree, he continued to perform research on traffic light control in a Ph.D project within the Manufacturing Networks group of professor Ivo Adan at the TU/e. Results of this project have been written in this thesis, have been presented at various conferences and have been written in several articles. The Ph.D. project ends with the realization of this thesis, which Stijn defends on January, 19th, 2017 at the Technical University of Eindhoven. Stijn is currently employed at the Laboratory for Industrial Mathematics Eindhoven (LIME).