Optimizing fixed-time control at isolated intersections

Part I: A single green interval per traffic light

S.T.G. Fleuren and E. Lefeber [*]

Abstract

In this paper we propose a novel group-based approach to optimize fixed-time traffic light control at isolated intersections. This novel approach is based on the mathematical framework of (Serafini and Ukovich 1989a). The proposed optimization formulation aggregates the binary design variables of the currently existing group-based approaches. With an extensive computational study we compare the computation times of the novel approach with the computation times of the currently existing group-based approaches. We consider three different objective functions: minimization of the period duration of a fixed-time schedule, minimization of the delay that road users experience, and maximization of the capacity of the intersection. The results show superiority of the novel approach for the former two objective functions. For the latter objective function, the two approaches have a comparable performance.

Keywords. fixed-time control; isolated intersection; optimization framework; cycle periodicity formulation; mixed-integer programming problem

1 Introduction

One of the most important ingredients of modern society is mobility. Two important aspects of mobility are speed and safety; we not only want to move as quickly as possible, but we also want to get to our destination safely. To improve safety, the access of competing traffic streams and pedestrian streams to the intersection is controlled with traffic lights. To enhance comfort of road users, it is desirable to control these traffic lights such that the travel times of road users are minimized. Two types of traffic light controllers try to achieve this goal: vehicle-actuated controllers and fixed-time controllers. The former type uses vehicle detection to control the traffic lights and the latter type does not; for fixed-time control the green, amber and red indicator lights are timed periodically at fixed intervals. Even though nowadays some developed countries use vehicle-actuated control, also these countries are still interested in fixed-time control for several reasons. The most obvious reason is its necessity when sensors, detecting vehicles, are not present or defective. Fixed-time cotrol is also useful in a heavily congested situation; in case of heavy congestion a vehicle-actuated controller behaves as a fixed-time controller.

[*] Eindhoven University of Technology, Eindhoven, The Netherlands.

Moreover, a vehicle-actuated controller might be based on a fixed-time controller. For example, when a vehicle-actuated controller is allowed to make slight adjustments to a fixed-time controller, e.g., extent or shorten the duration of a green interval, based on measured traffic data. Another advantage of fixed-time control is its predictability. Today, traffic lights cause unpredictable delays when navigating through a network of traffic lights. However, more floating car data becomes available, i.e., more data about the current traffic situation becomes available. As (Krijger 2013) shows, this data can be used to make future predictions for the state of each of the traffic lights; these predictions are particularly accurate for fixed-time control. The future states of the traffic lights can be visualized to road users, who can then use this information to adjust their travel speed, reduce the waiting time at traffic lights, and save fuel. Furthermore, these state predictions can be used to obtain better estimates for travel times; these travel time estimations include the estimated waiting time at the traffic lights. Car navigation systems can then use these travel time estimations to calculate a smart route through a network of traffic lights. Further motivation for fixed-time control is its simplicity to implement and the simplicity to create harmonizations between intersections when using fixed-time controllers, e.g., it is easier to create so called green waves when using fixed-time controllers. Such harmonizations between intersections could be created by first considering these intersections in isolation and creating a fixed-time controller for each intersection separately. Thereupon, the timing of these different fixed-time controllers with respect to each other is optimized by solving a coordination problem with for example (Gartner et al. 1975) or (Wünsch and Köhler 1990); this creates the so called harmonizations. Another motivation to consider intersections in isolation is that not all intersections are highly affected by neighboring intersections. It may be convenient (and justified) to consider such intersections in isolation. This motivates the focus of this paper: *fixed-time control at isolated intersections*.

Two general approaches exists that optimize fixed-time control at isolated intersections: a phase-based approach and a group-based approach. A fixed-time schedule, defining when traffic lights are green, amber and red during a repeating period, can be split up into phases. During such a phase, the indicator lights of the traffic lights do not change. A phase-based approach uses these phases as elementary building blocks, e.g., (Allsop 1971a,b, 1972, 1981; Gallivan and Heydecker 1988; Han 1990, 1996; Rouphail 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 fixed-time schedule 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 fixed-time schedule and the duration of the green intervals simultaneously, these methods also optimize the lane-use arrows at the intersection; these lane-use arrows are marked on the road and indicate in which direction vehicular traffic is allowed to depart 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 paper we use 'group-based' to refer to both group-based and lane-based methods. In this paper we do not consider the optimization of the lane-use arrows; this extension is the topic of another paper.

To model the structure of a fixed-time schedule, the aforementioned group-based methods introduce one binary design variable for each pair of competing (conflicting) traffic lights, i.e., the aforementioned group-based methods have one binary design variable for each two traffic lights whose traffic streams cannot safely cross the intersection simultaneously. This binary variable indicates which of the two traffic lights switches to green first during the interval $[0, T)$, where T is the period duration of the fixed-time schedule. In this paper we show that the values of these binary variables are not uniquely defined for each fixed-time schedule, i.e., multiple values for the binary design variables may be associated with the same fixed-time schedule. Since the group-based approach 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 the same solution, which may result in a large search tree, and as a consequence, large computation times.

In this paper we propose a group-based approach that does not posses this undesirable property; this group-based method is based on the mathematical framework of (Serafini and Ukovich 1989a). In that paper Serafini and Ukovich give 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 fixed-time 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 fixed-time schedule with a fixed period duration. However, in this paper we consider the period duration to be a design variable and we do not just try to find any feasible fixed-time schedule, but we try to find 'the best' fixed-time schedule, i.e., we try to find the fixed-time schedule 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 fixed-time schedule, 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 that is used to approximate the (average) delay that road users experience at a traffic light under fixed-time control. Many different formulae exist that try to approximate the average delay that road users experience at a traffic light under fixed-time control, e.g., the formulae from (Miller 1963; van den Broek et al. 2006; Webster 1958). In (van den Broek et al. 2006) different approximations are compared with each other. Especially, the formula that he proposes seems to be a very good approximation. The average delay approximations of (van den Broek et al. 2006), (Miller 1963) and (Webster 1958) among other approximations are convex in the design variables. When using such a convex approximation, then the convex objective 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.

The contributions of this paper are the following. We formulate a novel group-based approach to find the optimal fixed-time schedule at an isolated intersection; this optimization problem adds an objective function to the linear constraints of (Serafini and Ukovich 1989b). In addition, we 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 paper we propose a different integral cycle basis than the one that is used in (Serafini and Ukovich 1989b). For each cycle in this integral cycle basis we have one integral design variable. For the integral cycle basis that we propose, many of these integral 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 that is needed to solve the optimization problem. In this paper, we also relate our approach to the currently existing group-based approaches and we 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 when minimizing the period duration of the fixed-time schedule and when minimizing the average delay that road users experience at the intersection. When maximizing the capacity of the intersection, then the two approaches seem to have a comparable performance.

The organisation of this paper is as follows. First, we formulate a mixed-integer programming (MIP) problem in Section 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; we relate the integral-valued design variables of both approaches and we compare both approaches by means of an extensive computational study. The conclusions and recommendations of this paper can be found in Section 4.

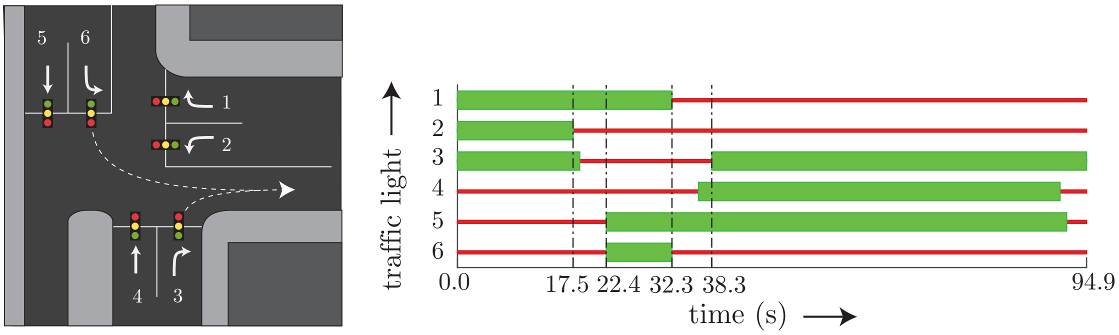
2 Formulating the MIP Problem

In practice each of the traffic lights can display three different colors: green, amber and red. However, mathematically it suffices to use only two different modes: effective green and effective red, see for example (Gartner et al. 1975); traffic departs during an effective green interval and it does not depart during an effective red interval. Note that this is different from the actual display colors; at the start of a display green interval the traffic light may still be effective red, which models the capacity loss at the start of a display green interval. This capacity loss is caused by the reaction time of road users and the acceleration of traffic. Furthermore, the amber interval may be modelled as part effective green and part effective red. In this way, we can model that traffic departs at the start of an amber interval, while traffic stops at the end of an amber interval. A fixed-time schedule that uses the effective green and effective red modes can easily be transformed into a fixed-time schedule that uses the actual display colors (and vise versa), see for example (Gartner et al. 1975). In this section we introduce an MIP formulation to optimize a fixed-time schedule that visualizes when each traffic light is effective green and effective red. 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

Traffic light (i) (-)	t_i^g (s)	t_i^r (s)
1	0	32.35
2	0	17.43
3	38.35	18.43
4	36.35	90.87
5	22.43	91.87
6	22.43	32.35

Table 1: The times (rounded to hundreds of a second) at which the traffic light $i = 1, \ldots, 6$ switch to effective green (t_i^g) and effective red (t_i^r) for the fixed-time schedule that is given in Figure 1b.

experience at the T-junction that is visualized in Figure 1a; the optimal fixed-time schedule is visualized in Figure 1b and the associated switching times are given in Table 1. Subsequently, we give the MIP problem, including the different objective functions, for the generic case.



(a) Visualisation of the intersection (T-junction) with six different traffic lights. We have numbered these traffic lights from one until six.

(b) The fixed-time schedule that minimizes the average delay that a road user experiences; we use the formula of (van den Broek et al. 2006) to approximate the delay. This schedule has a period duration of 94.87 seconds. On average a road user experiences a delay of 26.416 seconds for this fixed-time schedule.

Figure 1: An example of a T-junction and a fixed-time schedule.

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 that is visualized in Figure 1a. For this intersection the right-turn movement is equipped with a traffic light, which is common in Europe. Such a traffic light at a right-turn movement, e.g., traffic light 1, is often omitted in for example the United States of America; traffic can then make a right turn when the way is clear. The optimization problem (the general form of which we give in Section 2.2) that we formulate does not make any assumptions on which movement is equipped with a traffic light and which traffic streams are conflicting. Hence, this approach can be used to optimize fixed-time schedules in both the 'Europe' setting and the 'USA' setting. In the upcoming section we give the required inputs for the optimization of a fixed-time schedule. Subsequently, we introduce the real-valued design variables, we formulate the linear constraints and we introduce the objective function for this optimization problem.

2.1.1 Required Inputs.

In this section we give the inputs that are required to optimize a fixed-time schedule.

Signal Groups. Usually, the traffic lights at the intersection are divided amongst signal groups; the traffic lights in a signal group receive identical indication, i.e., the traffic lights in the same signal group always display the same color. We require a set of signal groups $\mathcal{S}$ for which we desire an optimized fixed-time schedule. For the example in Figure 1a we have:

$$\mathcal{S} = \{1, 2, 3, 4, 5, 6\},$$

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. We model the traffic that is waiting at the intersection by using first-in-first-out (FIFO) queues. Let $\mathcal{Q}$ denote the set of queues that is used to model the traffic that is 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 $q \in \mathcal{Q}_i$ (in a first come first serve 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 1a we model each of the lanes at which traffic arrives 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:

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

Thus, the traffic that is waiting at the intersection in Figure 1a is modelled with queues $\mathcal{Q} = \{1, 2, 3, 4, 5, 6\}$ 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. We express these arrival rates and saturation flow rate in passenger car equivalent units per seconds (PCE/s); a traffic flow that consists of multiple types of traffic, e.g., passenger cars, busses and motorcycles, can be converted into a traffic stream of only passenger cars (Wilson 2006; National Research Council (U.S.) 2010). The intersection in Figure 1a has the following arrival rates and saturation flow rates (in PCE/s):

$$\lambda_1 = \frac{320}{3600},\ \lambda_2 = \frac{280}{3600},\ \lambda_3 = \frac{180}{3600},\ \lambda_4 = \frac{980}{3600},\ \lambda_5 = \frac{820}{3600},\ \lambda_6 = \frac{150}{3600},$$
$$\mu_1 = \frac{1615}{3600},\ \mu_2 = \frac{1805}{3600},\ \mu_3 = \frac{1615}{3600},\ \mu_4 = \frac{1900}{3600},\ \mu_5 = \frac{1900}{3600},\ \mu_6 = \frac{1805}{3600}.$$

Note that the saturation flow rate that is 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 saturation flow rates we can obtain the loads $\rho_q := {}^{\lambda_q}/_{\mu_q}$.

Conflicts. When two traffic streams use a common part of the intersection that they cannot safely cross simultaneously, then we call their corresponding signal groups conflicting. We require a set $\Psi_{\mathcal{S}}$ of conflicting signal groups. For the intersection in Figure 1a we have:

$$\Psi_{\mathcal{S}} = \big\{\{1,4\},\{2,4\},\{2,5\},\{2,6\},\{3,6\},\{4,6\}\big\}.$$

Minimum Clearance Times. For each two conflicting signal groups $\{i,j\} \in \Psi_{\mathcal{S}}$, signal group j may only become effective green after signal group i has been effective red for at least $\underline{c}_{i,j}$ seconds. 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. We call this minimum amount of time $\underline{c}_{i,j}$ a minimum clearance time; we refer to the interval between signal group i switching to effective red and a conflicting signal group j switching to effective green as a clearance interval; we refer to its duration as a clearance time.

The minimum clearance time $\underline{c}_{i,j}$ depends on the geometry of the intersection amongst other factors. Consider for example the conflict between signal group 3 and signal group 6 in Figure 1a. The area where the traffic streams of signal groups 3 and 6 conflict is relatively close to the stop line of signal group 3 and relatively far away from the stop line of signal group 6. As a consequence the minimum clearance time $\underline{c}_{6,3}$ is larger than the minimum clearance time $\underline{c}_{3,6}$. We give the minimum clearance times for the intersection in Figure 1a below:

$$\begin{aligned}
&\underline{c}_{1,4} = 4, \quad \underline{c}_{2,4} = 4, \quad \underline{c}_{2,5} = 5, \quad \underline{c}_{2,6} = 5, \quad \underline{c}_{3,6} = 4, \quad \underline{c}_{4,6} = 4,\\
&\underline{c}_{4,1} = 4, \quad \underline{c}_{4,2} = 4, \quad \underline{c}_{5,2} = 3, \quad \underline{c}_{6,2} = 5, \quad \underline{c}_{6,3} = 6, \quad \underline{c}_{6,4} = 4.
\end{aligned}$$

For the T-junction of Figure 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, then signal group j is allowed to become effective green at most $\mathrm{abs}(\underline{c}_{i,j})$ seconds before signal group i becomes effective red, where $\mathrm{abs}(x)$ is the absolute value of x. In Appendix A we motivate the use of negative clearance times.

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 ($\overline{T}$) on the period duration. For the example of Figure 1a we have:

$$\underline{T} = 30, \qquad\qquad \overline{T} = 120.$$

Bounds on Effective Green Times and Effective Red Times. Another safety measure is to restrict the duration of an effective green (effective 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 may 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 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 $\overline{g}_i$. Similarly, we also have a (strictly positive) lower bound $\underline{r}_i > 0$ and an upper bound $\overline{r}_i$ on the duration of an effective red interval of signal group $i \in \mathcal{S}$. For the T-junction of Figure 1a we have:

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

2.1.2 Real-valued Design Variables.

The goal is to find the optimal fixed-time schedule for the isolated intersection of Figure 1a; we have depicted this optimal schedule in Figure 1b. Let T be the period duration of the fixed-time schedule. Furthermore, let ⓘ and 🅘 denote periodic events. The periodic event ⓘ (periodic event 🅘) 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:

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

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 2 we give these values for $f(\varepsilon)$ for the fixed-time schedule that is given in Figure 1b. A fixed-time schedule 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 fixed-time schedule it follows that the periodic event $\varepsilon \in \mathcal{E}$ occurs at times $(f(\varepsilon) + k)T$, $k \in \mathbb{Z}$. Let $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ 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, $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ is defined as follows:

$$\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2) \tag{1}$$

i	$f($ⓘ$)$	$f($🅘$)$
1	$^{0}/_{9487}$	$^{3235}/_{9487}$
2	$^{0}/_{9487}$	$^{1743}/_{9487}$
3	$^{3835}/_{9487}$	$^{1843}/_{9487}$
4	$^{3635}/_{9487}$	$^{9087}/_{9487}$
5	$^{2243}/_{9487}$	$^{9187}/_{9487}$
6	$^{2243}/_{9487}$	$^{3235}/_{9487}$

Table 2: The switching fractions $f($ⓘ$)$ $(f($🅘$))$ at which the effective green interval of signal group $i \in \mathcal{S}$ starts (ends) for the fixed-time schedule that is given in Figure 1b.

for some integer $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, 1\}$. The variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ 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 $\boldsymbol{T'} := 1/T$. All these real-valued design variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ and their integers $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously, i.e., for each of these variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$, both its own value and the value of its associated integer $z(\varepsilon_1, \varepsilon_2)$ are uniquely defined for each fixed-time schedule. We prove this unambiguity at the end of this section; we do so by proving for each of the real-valued design variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ that its value is included in some interval with a length that is strictly smaller than one, e.g., $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) \in [0, 1)$. 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.

2.1.3 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 1a we want to minimize the average weighted delay at the intersection:

$$D = \sum_{i \in \mathcal{S}} \sum_{q \in \mathcal{Q}_i} w_q d_q, \tag{2a}$$

where d_q is the (approximated) delay that road users experience at queue q and w_q is the weight factor that is associated with this queue q. To minimize the average delay that road user experience at the intersection we choose the weights proportional to the arrival rates, i.e,. $w_q = \frac{\lambda_q}{\Lambda}$, where $\Lambda = \sum_{q \in \mathcal{Q}} \lambda_q$.

We approximate d_q with the approximation of (van den Broek et al. 2006). Van den Broek divides a period into constant time intervals, which he calls slots; each slot has a length of $1/\mu_q$ seconds. Recall that μ_q is the saturation flow rate of queue q. Let $\rho_q > 0$ and $\sigma_q \geq 0$ be the average and the standard deviation of the amount of traffic that arrives during a slot, i.e., on average $\lambda_q := \rho_q \mu_q$ of traffic arrives per second at queue q. For ease of notation we define $r'_i := \boldsymbol{\gamma}($🅘$,$ⓘ$)$. For each queue $q \in \mathcal{Q}_i$ we can approximate the delay with the approximation of (van den Broek et al. 2006), which can be written as:

$$d_q = \frac{r'_i}{2(1-\rho_q)\rho_q} \left(\frac{\sigma_q^2}{\mu_q(1-\rho_q)} + \frac{\rho_q r'_i}{\boldsymbol{T'}} + \frac{r'_i \rho_q^2 \sigma_q^2}{\mu_q(1-r'_i)^2(1-r'_i-\rho_q)(1-\rho_q)} \right). \tag{2b}$$

If d_q is convex in the real-valued design variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ and $\boldsymbol{T}'$, then also the average weighted delay D is convex in these real-valued design variables; in Appendix B we prove that (2b) is convex.

2.1.4 Linear Constraints.

A fixed-time schedule 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 $\boldsymbol{T}'$:

$$1/\overline{T} \leq \boldsymbol{T}' \leq 1/\underline{T}. \tag{3a}$$

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

$$0 \leq \underline{g}_i \boldsymbol{T}' \leq \boldsymbol{\gamma}(\textcircled{i}, \textcircled{\colorbox{gray}{i}}) \leq \overline{g}_i \boldsymbol{T}', \tag{3b}$$

Note that $\boldsymbol{\gamma}(\textcircled{i}, \textcircled{\colorbox{gray}{i}})T$ is the effective green time of signal group i. Hence, (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:

$$0 < \underline{r}_i \boldsymbol{T}' \leq \boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}}, \textcircled{i}) \leq \overline{r}_i \boldsymbol{T}'. \tag{3c}$$

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:

$$0 < \max_{q \in \mathcal{Q}_i} \rho_q \leq \boldsymbol{\gamma}(\textcircled{i}, \textcircled{\colorbox{gray}{i}}). \tag{3d}$$

We call (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:

$$\underline{c}_{i,j} \boldsymbol{T}' \leq \boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}}, \textcircled{j}). \tag{3e}$$

2.1.5 Circuital Constraints.

The variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ are also related via circuital constraints, which are induced by the periodicity of the fixed-time schedule. These circuital constraints are best explained by using a directed graph

$G = (V, A)$ with vertices V and arcs A:

$$\begin{aligned} V &= \{\textcircled{i} \mid i \in \mathcal{S}\} \cup \{\colorbox{gray}{i} \mid i \in \mathcal{S}\}, \\ A &= A_g \cup A_r \cup A_c, \end{aligned}$$

where,

$$\begin{aligned} A_g &:= \{(\textcircled{i}, \colorbox{gray}{i}) \mid i \in \mathcal{S}\}, \\ A_r &:= \{(\colorbox{gray}{i}, \textcircled{i}) \mid i \in \mathcal{S}\}, \\ A_c &:= \{(\colorbox{gray}{i}, \textcircled{j}) \mid \{i, j\} \in \Psi_{\mathcal{S}}\}. \end{aligned}$$

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 $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$, i.e., the set A consists of the arcs $(\varepsilon_1, \varepsilon_2)$ for which $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ is subject to a (safety) constraint. Therefore, with each arc $(\varepsilon_1, \varepsilon_2) \in A$ we associate the variable $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$, see also Figure 2; recall that $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ 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 this 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 1 (Walk)**.** *A walk is a sequence of vertices $v_1, v_2, \ldots, v_N \in V$ 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 2 (Path)**.** *A path is a sequence of vertices $v_1, v_2, \ldots, v_N \in V$ 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.*

A path can be represented by the sets $\mathcal{P}^+$ and $\mathcal{P}^-$, which denote the sets 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. We define $\mathcal{P} := \mathcal{P}^+ \cup \mathcal{P}^-$. A cycle is defined in accordance with (Serafini and Ukovich 1989b; Kavitha and Krishna 2009):

Definition 3 (Cycle)**.** *A cycle is a closed path, i.e., a path for which $v_1 = v_N$.*

Thus, also a cycle is allowed to traverse arcs in the backward direction. We can represent a cycle by the sets $\mathcal{C}^+$ and $\mathcal{C}^-$, which denote the sets 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 $\mathcal{C} := \mathcal{C}^+ \cup \mathcal{C}^-$.

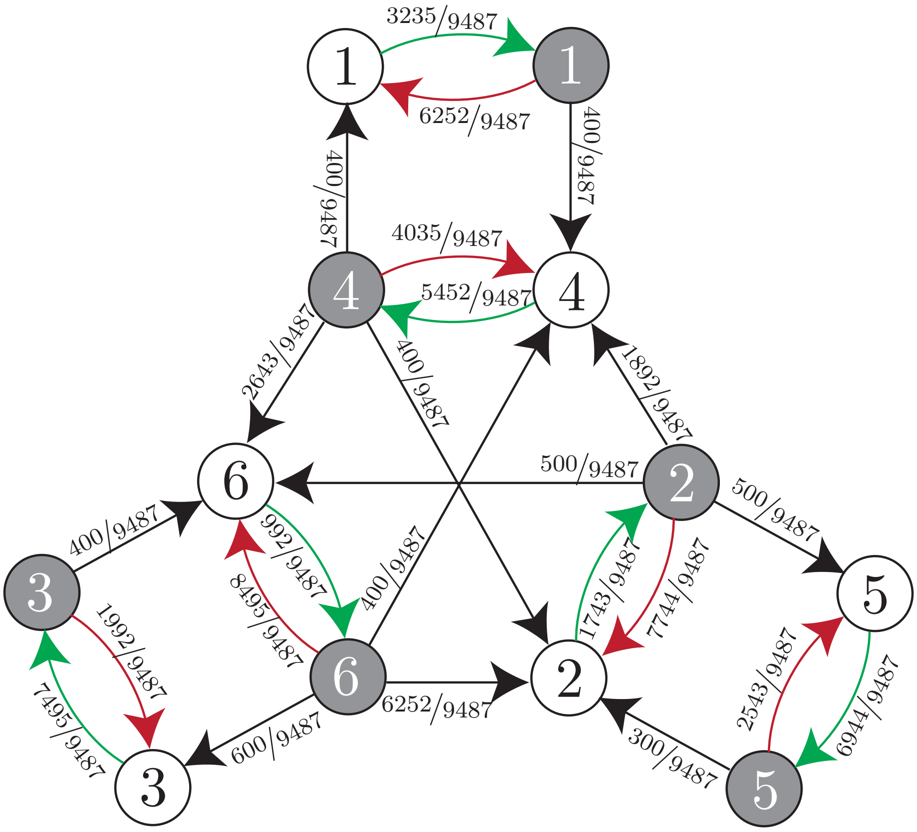
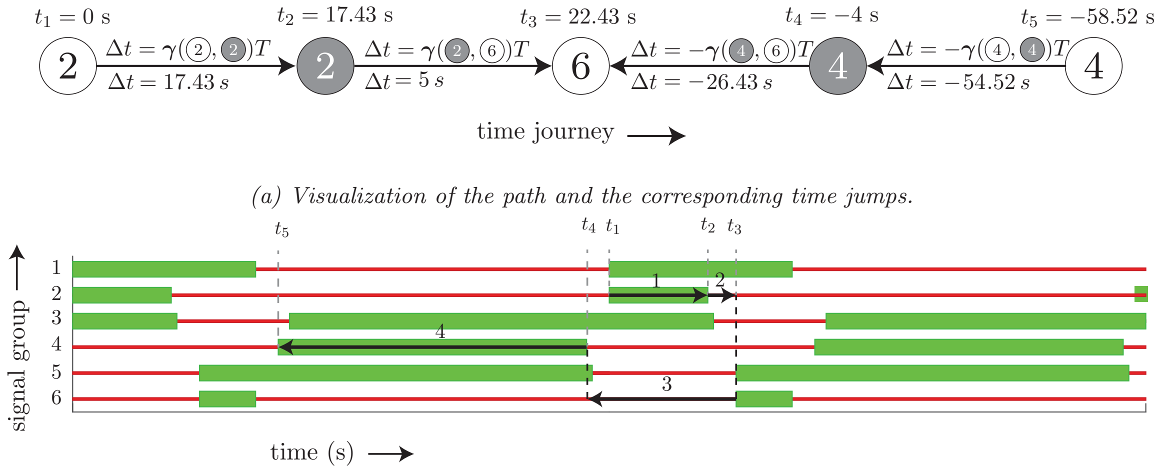


Figure 2: The constraint graph $G = (V, A)$ that corresponds to the fixed-time schedule in Figure 1b. The arcs in green (red) visualise effective green (effective red) intervals. The arcs in black represent clearance intervals. Attached to the tail of each arc is the corresponding value of $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$.

Definition 4 (Circuit)**.** *A circuit is a cycle for which the vertices $v_1, v_2, \ldots, v_{N-1}$ 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 2 contains a path with forward arcs $\mathcal{P}^+ = \{($②, ❷$), ($❷, ⑥$)\}$ and backward arcs $\mathcal{P}^- = \{($❸, ⑥$), ($③, ❸$\}$. Each such path can be associated with a journey through time. See for example Figure 3 for such a journey through time. Let v_s and v_e be the vertex at which the 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 $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)T$ seconds forward (backward) in time. The time journey then ends at a time at which event v_e occurs. To be more specific:

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



(a) Visualization of the path and the corresponding time jumps.

(b) The time journey visualized in the fixed-time schedule. The arrows represents the time journey. The time jumps are numbered 1 to 4. Jumps 3 and 4 are backward jumps.

Figure 3: The time journey associated with the path in the constraint graph G (see Figure 2) with forward arcs $\mathcal{P}^+ = \{(②,❷),(❷,⑥)\}$ and backward arcs $\mathcal{P}^- = \{(❸,⑥),(③,❸)\}$.

where z is some integer. To verify, plug in the definition $\boldsymbol{\gamma}(\varepsilon_1,\varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1,\varepsilon_2)$. 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 $\sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{P}^+} \boldsymbol{\gamma}(\varepsilon_1,\varepsilon_2)$ corresponds to the forward jumps in time of the associated time journey, and the term $\sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{P}^-} \boldsymbol{\gamma}(\varepsilon_1,\varepsilon_2)$ 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 fixed-time schedule 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 that is visualised in Figure 3. The path starts at the vertex ② and the corresponding journey through time starts at the time:

$$\begin{aligned} t_1 &:= f(②)T, \\ &= 0 \text{ seconds.} \end{aligned}$$

At this time the event ② takes place. The path then goes to the vertex ❷ via a forward arc $(②,❷) \in P^+$ and the corresponding time journey jumps $\Delta t = \boldsymbol{\gamma}(②,❷)T = 17.43$ seconds forward in time to the time:

$$\begin{aligned} t_2 &:= f(②)T + \boldsymbol{\gamma}(②,❷)T, \\ &= f(❷)T + z(②,❷)T, \\ &= 17.43 \text{ seconds,} \end{aligned}$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. To obtain the latter expression we have used the definition (1), which is the definition of $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$. Note that, by definition, event ❷ occurs at time t_2. Next, the path goes to the vertex ⑥ via the arc $(❷, ⑥) \in \mathcal{P}^+$ and the corresponding time journey jumps $\Delta t = \boldsymbol{\gamma}(❷, ⑥)T = 5$ seconds forward in time to the time:

$$
\begin{aligned}
t_3 &:= t_2 + \boldsymbol{\gamma}(❷, ⑥)T, \\
&= f(⑥)T + z(②, ❷)T + z(❷, ⑥)T, \\
&= 22.43 \text{ seconds},
\end{aligned}
$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. We have again used (1). Note that, by definition, event ⑥ occurs at time t_3. Subsequently, the path goes to the vertex ❹ via a backward arc $(❹, ⑥) \in P^-$ and the corresponding time journey jumps $\boldsymbol{\gamma}(❹, ⑥)T = 26.43$ seconds backward in time to the time:

$$
\begin{aligned}
t_4 &:= t_3 - \boldsymbol{\gamma}(❹, ⑥)T, \\
&= f(❹)T + z(②, ❷)T + z(❻, ⑥)T - z(❹, ⑥)T, \\
&= -4 \text{ seconds},
\end{aligned}
$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. At this time the event ❹ occurs. Finally, the path goes to the vertex ④ via a backward arc $(④, ❹) \in P^-$ and the corresponding time journey jumps $\boldsymbol{\gamma}(④, ❹)T = 54.52$ seconds backward in time to the time:

$$
\begin{aligned}
t_5 &:= t_4 - \boldsymbol{\gamma}(④, ❹)T, \\
&= f(④)T + z(②, ❷)T + z(❻, ⑥)T - z(❹, ⑥)T - z(④, ❹)T, \\
&= -58.52 \text{ seconds},
\end{aligned}
$$

where $z(\varepsilon_1, \varepsilon_2) \in \{-1, 0, +1\}$. Indeed at time t_5 event $v_e :=$ ④ occurs.

For any cycle $\mathcal{C}$ of 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 $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$:

$$
\sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^+} \boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1, \varepsilon_2) \in \mathcal{C}^-} \boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) = \boldsymbol{z}_{\mathcal{C}}, \tag{3f}
$$

where $\boldsymbol{z}_{\mathcal{C}} \in \mathbb{Z}$ 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 2 has a cycle $\mathcal{C}$ with forward arcs $\mathcal{C}^+ = \{(②, ❷), (❷, ⑥), (❹, ②)\}$ and backward arcs $\mathcal{C}^- = \{(❹, ⑥)\}$. Including constraint (3f) for every cycle $\mathcal{C}$ ($\boldsymbol{z}_{\mathcal{C}}$ is an integral-valued design variable) would ensure periodicity; these constraints then ensure that the time between two occurrences of the same periodic event 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 $\boldsymbol{z}_\mathcal{C}$. Consider the cycle $\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \colorbox{gray}{i}), (\colorbox{gray}{i}, \textcircled{j}), (\textcircled{j}, \colorbox{gray}{j}), (\colorbox{gray}{j}, \textcircled{i})\}$, where $\{i, j\} \in \Psi_\mathcal{S}$. Since, the effective green intervals of signal group i and signal group j must occur within one period, it should hold that:

$$\boldsymbol{\gamma}(\textcircled{i}, \colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i}, \textcircled{j}) + \boldsymbol{\gamma}(\textcircled{j}, \colorbox{gray}{j}) + \boldsymbol{\gamma}(\colorbox{gray}{j}, \textcircled{i}) = 1, \qquad \{i, j\} \in \Psi_\mathcal{S}, \tag{3g}$$

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 $\boldsymbol{z}_\mathcal{C}$ to one for each cycle $\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \colorbox{gray}{i}), (\colorbox{gray}{i}, \textcircled{i})\}$, $i \in \mathcal{S}$:

$$\boldsymbol{\gamma}(\textcircled{i}, \colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i}, \textcircled{i}) = 1, \qquad i \in \mathcal{S}, \tag{3h}$$

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 (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 that is 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 (3f) to $C\boldsymbol{\gamma} = \boldsymbol{z}_\mathcal{C}$, where $\boldsymbol{\gamma}$ is a column vector containing the variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$. 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, we call this space 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 (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 5 (Integral cycle basis)**.** *An integral cycle basis is a set of cycles $\{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ such that any cycle $\mathcal{C}$ in G can be written as:*

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

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

It is easy to prove that it suffices to include the cycle periodicity constraint (3f) only for the cycles in an integral cycle basis $\{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$. Assume that circuital constraint (3f) is satisfied for the cycles $\mathcal{C}_1, \ldots, \mathcal{C}_d$ and define $\boldsymbol{z}_{\mathcal{C}_i} := C_i \boldsymbol{\gamma} \in \mathbb{Z}$, $i = 1, \ldots, d$. For each cycle $\mathcal{C}$ we can then find $\alpha_1, \ldots, \alpha_d \in \mathbb{Z}$ such that:

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

which proves that circuital constraint (3f) is then 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, $i = 1, \ldots, d$ might not be integral and as a result $\alpha_1 \boldsymbol{z}_{\mathcal{C}_1} + \ldots + \alpha_d \boldsymbol{z}_{\mathcal{C}_d}$ might not be integral as well. Hence, when constraint (3f) is satisfied for all the cycles in some (non-integral) cycle basis of the constraint graph G, then this does not necessarily imply that constraint (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. To formulate the cycle periodicity constraints (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. To improve the computation time that is required to solve the optimization problem we would like the slack of these integral-valued design variables to be as small as possible. Therefore, some integral cycle bases $\mathcal{B} := \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ may be better than others to formulate the optimization problem: a smaller slack in the integral-valued design variables $\boldsymbol{z}_{\mathcal{C}}$, $\mathcal{C} \in \mathcal{B}$ relates to a smaller computation time that is needed to solve the MIP problem. This claim is supported by computational studies, e.g., the ones from (Liebchen 2003; Wünsch and Köhler 1990). 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 (3f). For each cycle $\mathcal{C}$ in the constraint graph G, define $\underline{z}_{\mathcal{C}}$ ($\overline{z}_{\mathcal{C}}$) to be the minimum (maximum) value that:

$$\sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}+} \boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}-} \boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2). \tag{4}$$

can attain for any solution to the linear constraints (3); these values could be obtained by solving two mixed-integer linear programming (MILP) problems: one minimizing (4) subject to the constraints (3), and one maximizing (4) subject to constraints (3). Note that the value for $\underline{z}_{\mathcal{C}}$ ($\overline{z}_{\mathcal{C}}$) is integral and independent of the integral cycle basis that is used to formulate (3f); each integral cycle basis poses the same restriction on the real-valued design variables $\boldsymbol{T}'$ 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:

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

Consider an integral cycle basis $\mathcal{B} := \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$. The vector of integral-valued design variables $\boldsymbol{z} = (\boldsymbol{z}_{\mathcal{C}_1}, \ldots, \boldsymbol{z}_{\mathcal{C}_d})$ can then, possibly, attain $w_{\mathcal{B}} := \prod_{i=1}^{d}(w_{\mathcal{C}_i} + 1)$ different values; when enumerating over all values of $\boldsymbol{z}$ 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 (3f), then the computation times of the associated MIP problem depend on the width of the integral cycle basis, see (Liebchen 2003; Wünsch and Köhler 1990); 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 that are associated with the circuital constraints (3g) and (3h); for such a cycle $\mathcal{C}$ we know the integral value of $\boldsymbol{z}_{\mathcal{C}}$, 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 built 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 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 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 $i = 1, \ldots, \nu(G)$. 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 8 (Strictly fundamental cycle basis (SFCB)). *The set of cycles $\mathcal{B} = \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ is a strictly fundamental cycle basis whenever $\mathcal{B}$ is the set of all the fundamental cycles that are associated with some spanning forest $\mathcal{F} \subseteq A$. In other words $\mathcal{B} = \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ 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 $\mathcal{F} \cup \{a\}$ that 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 $\{i, j\} \in \Psi_s$:

$$\begin{aligned} V' &:= \mathcal{S} \\ A' &:= \Psi_{\mathcal{S}}. \end{aligned} \tag{5}$$

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 4:

$$\mathcal{F} := \{({\color{gray}\textcircled{i}}, \textcircled{j}) \mid \{i,j\} \in \mathcal{F}', i < j\} \cup A_g \tag{6}$$

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 that are associated with the circuital constraints (3h). It does however not include all the cycles that are associated with the circuital constraints (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 build.

Lemma 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 that is calculated with* (6)*. Define $\mathcal{B} = \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ 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 that is obtained from $\mathcal{B}$ when, for each conflict $\{i,j\} \notin \mathcal{F}'$, $i < j$, we replace the cycle $\mathcal{C}_{\mathcal{F}}(({\color{gray}\textcircled{j}}, \textcircled{i}))$ by the cycle:*

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

The set $\mathcal{B}'$ is an integral cycle basis of the constraint graph G that includes all the cycles that are associated with the circuital constraints (3g)–(3h)*.*

Proof. See Appendix C. □

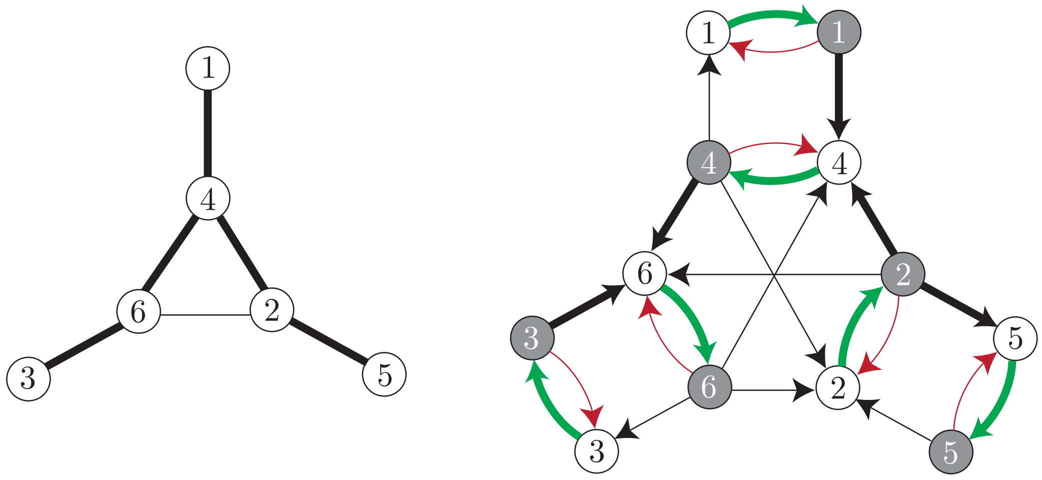


Figure 4: Visualization of the spanning forest $\mathcal{F}$ of the constraint graph G (right) that is obtained from a spanning forest $\mathcal{F}'$ of the conflict graph G' (left) for the intersection in Figure 1a; the arcs in the spanning forest are visualized in bold.

Thus, using Lemma 1 we can obtain an integral cycle basis of the constraint graph G that includes circuital constraints (3g)–(3h). To this end, we need a spanning forest $\mathcal{F}'$ of the conflict graph G'; we calculate this spanning forest with the algorithm that is 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 that is associated with the intersection in Figure 1a, the resulting integral cycle basis consists of thirteen cycles. Six of these cycles correspond to circuital constraints (3g) and six of them correspond to circuital constraints (3h). The last cycle is the following one:

$$\mathcal{C} = \mathcal{C}^{+} = \{(\text{④}, \text{❹}), (\text{❹}, \text{②}), (\text{②}, \text{❷}), (\text{❷}, \text{⑥}), (\text{⑥}, \text{❻}), (\text{❻}, \text{④})\}. \tag{7}$$

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

Remark. The integral cycle basis that is constructed with Lemma 1 includes each cycle that is associated with the circuital constraints (3g)–(3h). Therefore, the circuital constraints (3f) that are associated with these $|\Psi_{\mathcal{S}}| + |\mathcal{S}|$ cycles are made redundant by circuital constraints (3g)–(3h). Hence, only $d - (|\Psi_{\mathcal{S}}| + |\mathcal{S}|) = |\Psi_{\mathcal{S}}| - |\mathcal{S}| + \nu(G)$ integral-valued design variables $\boldsymbol{z}_{\mathcal{C}}$ remain.

2.1.6 Well-posedness.

For the T-junction of Figure 1 it holds that the minimum clearance times $\underline{c}_{i,j}$, $\{i, j\} \in \Psi_{\mathcal{S}}$ are non-negative; in Appendix A we motivate the use of negative clearance times. In case that all minimum clearance times are non-negative, then all real-valued design variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ are restricted to the interval $[0, 1)$. The inclusion $\boldsymbol{\gamma}(\text{ⓘ}, \text{🅘}) \in [0, 1)$, $i \in \mathcal{S}$, follows from circuital constraint (3h) combined with the positivity of each effective red time (3c), the inclusion $\boldsymbol{\gamma}(\text{🅘}, \text{ⓘ}) \in [0, 1)$, $i \in \mathcal{S}$, follows from the circuital constraint (3h) combined with the stability constraint (3d), and the inclusion $\boldsymbol{\gamma}(\text{🅘}, \text{ⓙ}) \in [0, 1)$, $\{i, j\} \in \Psi_{\mathcal{S}}$ follows from circuital constraint (3g) combined with the stability constraint (3d) and the non-negativity of the clearance times (3e). As a result, $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ 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 $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) \in [0, 1)$ it follows that the variable $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ and its associated integer $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously, i.e., for each fixed-time schedule the value of the real-valued design variable $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ 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 $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$.

2.1.7 Obtaining the Fixed-time Schedule.

The objective function (2a) together with linear constraints (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 $\boldsymbol{T'} := 1/T$ and $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$. For the example of Figure 1a we want to find the fixed-time schedule that minimizes the average delay that road users experience at this intersection. This fixed-time schedule can be found by solving the mixed-integer (convex) programming problem.

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

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

Similarly, we can also obtain the relation:

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

Since, the $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$'s are only known for the arcs $(\varepsilon_1, \varepsilon_2) \in A$ we cannot immediately find all fractions $f(\varepsilon)$ from $f($①$)$; see the constraint graph G in Figure 2. However, we can obtain $f($❶$)$ and $f($❹$)$ from $f($①$)$. Subsequently, we can obtain $f($②$)$, $f($④$)$ and $f($⑥$)$ from $f($❹$)$, et cetera. Since the graph of Figure 2 consists of one component we can obtain all fractions $f(\varepsilon)$, $\varepsilon \in \mathcal{E}$ in this way.

Remark. When creating harmonizations between intersections, then it is desirable that the fixed-time schedules of these intersections have a common period duration. Such a common period duration can be found by optimizing the fixed-time schedules of these intersections simultaneously. In that case the constraint graph G consists of multiple connected components. 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 $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ is not restricted by any constraints. Hence, we can set the fixed-time schedules of the different connected components independently.

2.2 The Generic MIP Problem

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

$$
\begin{aligned}
&\underset{\boldsymbol{T}',\boldsymbol{\gamma},\boldsymbol{z}}{\text{minimize}}\;\; J(\boldsymbol{T}',\boldsymbol{\gamma},\boldsymbol{z}), && && \text{(8a)}\\
&\text{subject to:} && && \\
&0 \le 1/\overline{T} \le \boldsymbol{T}' \le 1/\underline{T} && && \text{(8b)}\\
&0 \le \underline{g}_i \boldsymbol{T}' \le \boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) \le \overline{g}_i \boldsymbol{T}', && i \in \mathcal{S}, && \text{(8c)}\\
&0 < \underline{r}_i \boldsymbol{T}' \le \boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{i}) \le \overline{r}_i \boldsymbol{T}', && i \in \mathcal{S}, && \text{(8d)}\\
&0 < \max_{q\in\mathcal{Q}_i} \rho_q \le \boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}), && i \in \mathcal{S}, && \text{(8e)}\\
&\underline{c}_{i,j} \boldsymbol{T}' \le \boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j}), && \{i,j\} \in \Psi_{\mathcal{S}}, && \text{(8f)}\\
&\sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}+} \boldsymbol{\gamma}(\varepsilon_1,\varepsilon_2) - \sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}-} \boldsymbol{\gamma}(\varepsilon_1,\varepsilon_2) = \boldsymbol{z}_{\mathcal{C}}, && \forall \mathcal{C} \in \mathcal{B} && \text{(8g)}\\
&\boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j}) + \boldsymbol{\gamma}(\textcircled{j},\colorbox{gray}{j}) + \boldsymbol{\gamma}(\colorbox{gray}{j},\textcircled{i}) = 1, && \{i,j\} \in \Psi_{\mathcal{S}} && \text{(8h)}\\
&\boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{i}) = 1, && i \in \mathcal{S} && \text{(8i)}\\
&\boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j}) \ge \epsilon T', && \{i,j\} \in \Psi_{\mathcal{S}}. && \text{(8j)}
\end{aligned}
$$

where ϵ is some 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 (8b)–(8i) correspond to the constraints (3a)–(3h). Constraints (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 that was 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 (8b) restricts the period duration T to the correct interval $[\underline{T},\overline{T}]$. Constraints (8c) (constraints (8d)) gives a lower and an upper bound on each effective green (effective red) time. Constraints (8f) ensure that the minimum clearance times between conflicting signal groups are satisfied. Constraints (8e) guarantee stability of each signal group. Constraints (8g)–(8i) are the cycle periodicity constraints; these constraints model the periodicity of the fixed-time schedule. As previously mentioned, (8g) should be satisfied for each cycle $\mathcal{C}$ in the constraint graph G; however, as we have shown in Section 2.1.5, we only have to include this constraint for a subset of the cycles: an integral cycle basis of the constraint graph G. Constraints (8j) need some more explanation. These constraints make sure that a clearance time $\boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j})T$, $\{i,j\} \in \Psi_{\mathcal{S}}$ is restricted to the interval $(-T,T)$. To obtain this interval, first combine the constraint (8j) with the circuital constraint (8h) to obtain $0 < \boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j}) < 1$. Furthermore, combine the positivity of the effective green time (8e) with the positivity of the effective red time (8d) and the circuital constraint (8i) which gives $0 < \boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) < 1$. The inequalities $0 < \boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j}) < 1$ and $0 < \boldsymbol{\gamma}(\textcircled{i},\colorbox{gray}{i}) < 1$ together imply that $\boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j})$ is restricted to the interval $(-1,1)$ and therefore that $\boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j})T$ is restricted to the interval $(-T,T)$; a clearance time $\boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j})T$ then, as desired, refers to the time between an occurrence of the event $\colorbox{gray}{$i$}$ and the next or the previous occurrence of the event $\textcircled{$j$}$ depending on the sign of $\boldsymbol{\gamma}(\colorbox{gray}{i},\textcircled{j})$. Note that the MIP problem (8) permits negative clearance times. However, if a clearance time $\boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}},\textcircled{j})T$ is negative, then (8j) only permits the effective green interval of signal group j to start after the effective green interval of signal group i has started.

2.2.1 Well-posedness.

What remains to be proved is that each real-valued design variable $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ and its associated integer $z(\varepsilon_1, \varepsilon_2)$ are defined unambiguously; recall that $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) := f(\varepsilon_2) - f(\varepsilon_1) + z(\varepsilon_1, \varepsilon_2)$. As already stated, it holds that $\boldsymbol{\gamma}(\textcircled{i},\textcircled{\colorbox{gray}{i}}) \in (0, 1)$, which implies that $\boldsymbol{\gamma}(\textcircled{i},\textcircled{\colorbox{gray}{i}})$ and its corresponding integer $z(\textcircled{$i$},\textcircled{\colorbox{gray}{i}})$ are defined unambiguously. Similarly, we can find that $\boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}},\textcircled{i}) \in (0, 1)$, which implies that $\boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}},\textcircled{i})$ and its corresponding integer $z(\textcircled{\colorbox{gray}{i}},\textcircled{i})$ are also defined unambiguously; this inequality follows from the positivity of the effective green time (8e), the positivity of the effective red time (8d) and the circuital constraint (8i). Furthermore, from the already proved inequality $0 < \boldsymbol{\gamma}(\textcircled{i},\textcircled{\colorbox{gray}{i}}) + \boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}},\textcircled{j}) < 1$ it follows that $\boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}},\textcircled{j}) \in (-\boldsymbol{\gamma}(\textcircled{i},\textcircled{\colorbox{gray}{i}}), 1 - \boldsymbol{\gamma}(\textcircled{i},\textcircled{\colorbox{gray}{i}}))$; since $\boldsymbol{\gamma}(\textcircled{i},\textcircled{\colorbox{gray}{i}})$ is defined unambiguously, this inclusion implies that $\boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}},\textcircled{j})$ and its integer $z(\textcircled{\colorbox{gray}{i}},\textcircled{j})$ are both also defined unambiguously.

Remark. Note that constraint (8j) is not needed if $\underline{c}_{i,j} \geq 0$; it then holds that $\boldsymbol{\gamma}(\textcircled{i},\textcircled{\colorbox{gray}{i}}) + \boldsymbol{\gamma}(\textcircled{\colorbox{gray}{i}},\textcircled{j}) \geq \max_{q \in \mathcal{Q}_i} \rho_q > 0$. Hence, we did not need the constraints (8j) when formulating the MIP problem for the T-junction of the previous section.

2.2.2 Objective Functions.

Objective functions that are commonly used when optimizing fixed-time schedules are: minimizing the period duration of the fixed-time schedule, 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 $-\boldsymbol{T}'$. Therefore, the objective function is linear in the design variables when minimizing the period duration of the fixed-time schedule; the resulting problem is a mixed-integer linear programming (MILP) problem.

When maximizing the capacity of the intersection, then we multiply the left-hand sides of (8e) $(\max\limits_{q \in \mathcal{Q}_i} \rho_q)$ by a common capacity factor $\boldsymbol{\beta}$, which we call the *growth factor*. The objective is to maximize $\boldsymbol{\beta}$ (minimize $-\boldsymbol{\beta}$), i.e., the objective is to find the maximum growth factor $\beta^{\max}$ of the arrival rates λ_q, $q \in \mathcal{Q}$ for which we can find a solution to (8b)–(8j). A maximum value $\beta^{\max}$ that is less than one, indicates that the intersection is overloaded by $100(1 - \beta^{\max})$ percent. A maximum value $\beta^{\max}$ 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, then the constraints remain linear and the objective function is linear as well. Therefore, the resulting problem is an MILP problem.

The third 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 (approximated) delay that road users experience at queue q and w_q is the weight factor of this queue. The average weighted delay D is convex in the real-valued design variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2)$ and $\boldsymbol{T}'$ whenever d_q is convex in these real-valued design variables. This is satisfied by the approximations from (Miller 1963; van den Broek et al. 2006; Webster 1958) amongst other approximations. The resulting problem is then a mixed-integer convex programming problem.

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 with the new approach of Section 2, we first consider the small example of Figure 1. For this intersection, the optimal fixed-time schedule is the one in Figure 1b. In this section we relate the integral variables $\boldsymbol{z}_{\mathcal{C}}$ 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 that is proposed in this paper.

3.1 Currently Existing Group-based Approaches

To model the structure of a fixed-time schedule, the currently existing group-based approaches introduce one binary design variable $\boldsymbol{\Omega}_{i,j}$ for each pair of conflicting signal groups $\{i,j\} \in \Psi_{\mathcal{S}}$. Thus, the number of integral-valued design variables is equal to the number of conflicts at the intersection, which is $|\Psi_{\mathcal{S}}|$; for the example in Figure 1 we have 6 binary design variables: $\boldsymbol{\Omega}_{1,4}$, $\boldsymbol{\Omega}_{2,4}$, $\boldsymbol{\Omega}_{2,5}$, $\boldsymbol{\Omega}_{2,6}$, $\boldsymbol{\Omega}_{3,6}$ and $\boldsymbol{\Omega}_{4,6}$. The binary variable $\boldsymbol{\Omega}_{i,j}$ is based on the fractions $\boldsymbol{f}(\textcircled{i})$ and $\boldsymbol{f}(\textcircled{j})$: $\boldsymbol{\Omega}_{i,j} = 0$ if $\boldsymbol{f}(\textcircled{i}) \leq \boldsymbol{f}(\textcircled{j})$ and $\boldsymbol{\Omega}_{i,j} = 1$ otherwise; the fractions $\boldsymbol{f}(\varepsilon)$, $\varepsilon \in \mathcal{E}$ are visualized in bold as these are real-valued design variables for the group-based approaches. We can prove that $\boldsymbol{f}(\textcircled{i}) \neq \boldsymbol{f}(\textcircled{j})$ and, as a result, it holds that $\boldsymbol{\Omega}_{j,i} = 1 - \boldsymbol{\Omega}_{i,j}$. We can obtain the inequality $\boldsymbol{f}(\textcircled{i}) \neq \boldsymbol{f}(\textcircled{j})$ for each conflict $\{i,j\} \in \Psi_{\mathcal{S}}$ by combining the well-posedness constraint (8j) with the circuital constraint (8h). The binary design variables $\boldsymbol{\Omega}_{i,j}$ are not uniquely defined for each fixed-time schedule. For example, with the fixed-time schedule in Figure 1b we can associate four different values for the binary design variables $\boldsymbol{\Omega}_{i,j}$, see Table 3. If we may shift the fixed-time schedule in time, e.g., let the effective 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 $\boldsymbol{f}(\textcircled{1}) = 0$ and, as a result, only two combinations remain for the example. In general, this ambiguity then only occurs when $\boldsymbol{f}(\textcircled{i}) = 0$ (or $\boldsymbol{f}(\textcircled{i}) = 1$) 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 fixed-time schedule with $\boldsymbol{\Omega}_{i,j} = 0$ for all conflicting signal groups j is then the same as the fixed-time schedule with $\boldsymbol{\Omega}_{i,j} = 1$ for all conflicting signal groups j. However, if some fraction $\boldsymbol{f}(\textcircled{i})$ is close to zero or close to one, then these two fixed-time schedules 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.

Remark. Not all cited group-based approaches use the real-valued design variables $\boldsymbol{f}(\textcircled{i}) \in [0,1]$, $i \in \mathcal{S}$ and the binary design variables $\boldsymbol{\Omega}_{i,j}$, $\{i,j\} \in \Psi_{\mathcal{S}}$, $i < j$ (in combination with the durations of the effective green intervals) directly. However, by applying a coordinate transformation, each of these

	$\boldsymbol{f}(①)$	$\boldsymbol{f}(②)$	$\boldsymbol{f}(③)$	$\boldsymbol{f}(④)$	$\boldsymbol{f}(⑤)$	$\boldsymbol{f}(⑥)$	$\boldsymbol{\Omega}_{1,4}$	$\boldsymbol{\Omega}_{2,4}$	$\boldsymbol{\Omega}_{2,5}$	$\boldsymbol{\Omega}_{2,6}$	$\boldsymbol{\Omega}_{3,6}$	$\boldsymbol{\Omega}_{4,6}$
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: Four different combinations of the binary design variables $\boldsymbol{\Omega}_{i,j}$ that could be associated with the fixed-time schedule in Figure 1b

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 $\boldsymbol{f}(ⓘ)$, $i \in \mathcal{S}$ this symmetric variant uses the variables $\boldsymbol{f}(🅘) \in [0,1]$, $i \in \mathcal{S}$; for this symmetric variant the integral-valued design variable $\boldsymbol{\Omega}_{i,j}$ equals one whenever $\boldsymbol{f}(🅘) \leq \boldsymbol{f}(🅙)$ and it equals zero otherwise.

3.2 Novel Modeling Approach

Where the currently existing group-based approaches need 6 binary variables for the example of Figure 1, the MIP problem of Section 2 requires only one integral-valued design variable. The integral-valued design variables are needed to formulate the cycle periodicity constraints (8g). As mentioned before, in general, a cycle basis of a graph G with vertices V and arcs A consists of $|A| - |V| + \nu(G)$ cycles, where $\nu(G)$ is the number of connected components of the graph G. Hence, the integral cycle basis that is required to formulate the cycle periodicity constraints (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 that are associated with the circuital constraints (8h)–(8i); for these cycles we know the integral multiplicity $\boldsymbol{z}_{\mathcal{C}}$. Hence, $|\Psi_{\mathcal{S}}| - |\mathcal{S}| + \nu(G)$ integral-valued design variables remain. For the example in Figure 1a only one integral-valued design variable $\boldsymbol{z}_{\mathcal{C}}$ with unknown value remains, e.g., the integral variable $\boldsymbol{z}_{\mathcal{C}}$ that is associated with cycle (7), which is the cycle:

$$\mathcal{C} = \mathcal{C}^{+} = \{(④,❹),(❹,②),(②,❷),(❷,⑥),(⑥,❻),(❻,④)\}.$$

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

3.3 Relating the Approaches

The integral-valued design variable $\boldsymbol{z}_{\mathcal{C}}$ is related to the binary design variables $\boldsymbol{\Omega}_{i,j}$ as follows:

Lemma 2. *Consider a cycle basis $\mathcal{B} = \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ that is obtained by using Lemma 1. Consider a cycle $\mathcal{C} \in \mathcal{B}$ that does not equal the circuital constraint* (8i)*; for the circuital constraint* (8i) *we know* *that the multiplicity $\boldsymbol{z}_{\mathcal{C}}$ equals one. For each such cycle, we can relate its integral-valued design variable $\boldsymbol{z}_{\mathcal{C}}$ to the binary design variables $\boldsymbol{\Omega}_{i,j}$ as follows:*

$$\boldsymbol{z}_{\mathcal{C}} = \sum_{(\textcircled{i},\textcircled{j}) \in \mathcal{C}^+} \boldsymbol{\Omega}_{i,j} - \sum_{(\textcircled{i},\textcircled{j}) \in \mathcal{C}^-} \boldsymbol{\Omega}_{i,j}.$$

Proof. See Appendix D. □

For the example of Figure 1, we have only one cycle for which the value of $\boldsymbol{z}_{\mathcal{C}}$ is not known before optimization; this is the cycle (7). Applying Lemma 2 for this cycle gives: $\boldsymbol{z}_{\mathcal{C}} = \boldsymbol{\Omega}_{4,2} + \boldsymbol{\Omega}_{2,6} + \boldsymbol{\Omega}_{6,4} = 2 - \boldsymbol{\Omega}_{2,4} + \boldsymbol{\Omega}_{2,6} - \boldsymbol{\Omega}_{4,6}$. From Table 3 we can see that all four combinations of $\boldsymbol{\Omega}_{i,j}$ are captured by $\boldsymbol{z}_{\mathcal{C}} = 1$.

Remark. Fixing the binary variables $\boldsymbol{\Omega}_{i,j}$, $\{i,j\} \in \Psi_{\mathcal{S}}$ of the currently existing group-based approaches fixes the values for the integral-valued design variables $\boldsymbol{z}_{\mathcal{C}}$ of the novel approach; however, this does not hold the other way around. Therefore, when fixing the binary variables of a group-based approach (optimization problem 1) and fixing the integral variables of the novel approach to the associated values (optimization problem 2), then the fixed-time schedules that satisfy the constraints of optimization problem 1 are a subset of the fixed-time schedules that satisfy the constraints of optimization problem 2.

The currently existing group-based approaches and the novel approach that is given in this paper are closely related to the periodic event scheduling problem (PESP), see the following intermezzo.

Intermezzo: Periodic Event Scheduling Problem (PESP)

Introduced by (Serafini and Ukovich 1989a), the periodic event scheduling problem (PESP) is a framework for optimizing periodic schedules. This optimization framework has been widely used, especially in the railway setting (Lindner 2000; Peeters 2003; Odijk 1996; Schrijver and Steenbeek 1993). For the PESP we require a period duration T and a constraint graph $G = (V, A)$. Each vertex $\varepsilon \in V$ of this constraint graph represents a periodic event. With each such a vertex $\varepsilon \in V$ we can associate a time $t(\varepsilon) \in [0, T]$ that indicates when this event occurs. Furthermore, for each arc $(\varepsilon_1, \varepsilon_2) \in A$ we are given a lower bound $l(\varepsilon_1, \varepsilon_2)$ and an upper bound $u(\varepsilon_1, \varepsilon_2)$ on the time between periodic event ε_1 and periodic event ε_2, i.e., for each arc $(\varepsilon_1, \varepsilon_2) \in A$ the times $t(\varepsilon_1)$ and $t(\varepsilon_2)$ have to satisfy the following periodic interval constraint:

$$(t(\varepsilon_2) - t(\varepsilon_1) - l(\varepsilon_1, \varepsilon_2)) \bmod T \leq u(\varepsilon_1, \varepsilon_2) - l(\varepsilon_1, \varepsilon_2). \tag{9}$$

The objective is to find the times $t(\varepsilon) \in [0, T]$, $\varepsilon \in V$ that minimize some weighted sum:

$$\sum_{(\varepsilon_1, \varepsilon_2) \in A} w(\varepsilon_1, \varepsilon_2) \, (t(\varepsilon_2) - t(\varepsilon_1) - l(\varepsilon_1, \varepsilon_2)) \bmod T. \tag{10}$$

In the railway setting an event $\varepsilon \in V$ corresponds for example to an arrival of a train at a station or a departure of a train from a station. The arcs $a \in A$ then represent trips (of a train from one station to the next), dwell intervals (trains waiting at the station), or transfer intervals (intervals wherein passengers can transfer from one train to another). The objective function can then for example be the minimization of the (average) time that passengers have to wait for their transfers.

Two different formulations of the PESP exist. The original formulation directly optimizes the event times $\boldsymbol{t}(\varepsilon)$ and uses integral variables $\boldsymbol{\Omega}(\varepsilon_1, \varepsilon_2)$ to calculate the modulus in (9) and (10):

Original formulation

$$\begin{aligned}
&\text{minimize} \sum_{(\varepsilon_1,\varepsilon_2)\in A} w(\varepsilon_1,\varepsilon_2)\big(\boldsymbol{t}(\varepsilon_2)-\boldsymbol{t}(\varepsilon_1)+T\boldsymbol{\Omega}(\varepsilon_1,\varepsilon_2)-l(\varepsilon_1,\varepsilon_2)\big)\\
&\text{subject to}\\
&l(\varepsilon_1,\varepsilon_2) \le \boldsymbol{t}(\varepsilon_2)-\boldsymbol{t}(\varepsilon_1)+\boldsymbol{\Omega}(\varepsilon_1,\varepsilon_2)T \le u(\varepsilon_1,\varepsilon_2), && \forall(\varepsilon_1,\varepsilon_2)\in A,\\
&\boldsymbol{t}(\varepsilon)\in[0,1] && \forall\varepsilon\in V,\\
&\boldsymbol{\Omega}(\varepsilon_1,\varepsilon_2)\in\mathbb{Z} && \forall(\varepsilon_1,\varepsilon_2)\in A,
\end{aligned} \tag{11}$$

The formulation of the currently existing group-based approaches that we have mentioned in this paper, are very similar to this original formulation; these group-based approaches formulate the constraints on the minimum clearance times as follows:

$$\boldsymbol{T}'\underline{c}_{i,j} \le \boldsymbol{f}(\textcircled{j}) - \big(\boldsymbol{f}(\textcircled{i}) + g'_i\big) + \boldsymbol{\Omega}_{i,j}, \qquad \{i,j\}\in\Psi_{\mathcal{S}}$$

where g'_i is the effective green time of signal group i expressed as a fraction of the period duration. Notice that $\big(f(\textcircled{i}) + g'_i\big)$ is a time (expressed as a fraction of the period duration) at which signal group i becomes effective red; this fraction is not necessarily included in the interval $[0,1]$. The resemblance with constraint (11) becomes apparent by performing the coordinate transformation $t(\textcircled{j}) := \boldsymbol{f}(\textcircled{j})/\boldsymbol{T}'$ and $T := 1/\boldsymbol{T}'$, and defining $t(\textcircled{\colorbox{gray}{i}}) := \big(\boldsymbol{f}(\textcircled{i}) + g'_i\big)/\boldsymbol{T}'$. We can then rewrite the above constraint to:

$$\underline{c}_{i,j} \le t(\textcircled{j}) - t(\textcircled{\colorbox{gray}{j}}) + \boldsymbol{\Omega}_{i,j}T, \qquad \{i,j\}\in\Psi_{\mathcal{S}}.$$

Besides the original formulation of the PESP also another formulation exists; this alternative formulation is called the *cycle periodicity formulation*. Instead of the times $t(\varepsilon)$, $\varepsilon \in V$, this formulation optimizes the tensions $\boldsymbol{\chi}(\varepsilon_1, \varepsilon_2) := t(\varepsilon_2) - t(\varepsilon_1) + \boldsymbol{\Omega}(\varepsilon_1, \varepsilon_2)T$:

Cycle periodicity formulation

$$
\begin{aligned}
&\text{minimize} \sum_{(\varepsilon_1,\varepsilon_2)\in A} w(\varepsilon_1, \varepsilon_2)\boldsymbol{\chi}(\varepsilon_1, \varepsilon_2) \\
&\text{subject to} \\
&l(\varepsilon_1, \varepsilon_2) \le \boldsymbol{\chi}(\varepsilon_1, \varepsilon_2) \le u(\varepsilon_1, \varepsilon_2), && \forall(\varepsilon_1, \varepsilon_2) \in A, \\
&\sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}} \boldsymbol{\chi}(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}} \boldsymbol{\chi}(\varepsilon_1, \varepsilon_2) = \boldsymbol{z}_{\mathcal{C}}T, && \forall\mathcal{C} \in \mathcal{B}, \\
&\boldsymbol{z}_{\mathcal{C}} \in \mathbb{Z},
\end{aligned}
$$

where $\mathcal{B}$ is an integral cycle basis of some graph $G = (V, A)$. Just like the optimization problem that is proposed in this paper, the cycle periodicity formulation uses cycle periodicity constraints to force periodicity. Also some differences exist between the optimization problem that is proposed in this paper and the cycle periodicity formulation of the PESP, which we have given above. We elaborate on some of these differences. First, the constraint graph $G = (V, A)$ associated with optimization problem (8) contains some cycles $\mathcal{C}$ for which the integral-valued design variable $\boldsymbol{z}_{\mathcal{C}}$ must be fixed, i.e., we have $\boldsymbol{z}_{\mathcal{C}} = 1$ for the cycles that are associated with the circuital constraints (8h)–(8i); we would like to include these zero-width cycles in our cycle basis. Second, we do not assume that the period duration T is fixed. Therefore, for the constraints to remain linear, we use the design variables $\boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) := \chi(\varepsilon_1, \varepsilon_2)/T$, $(\varepsilon_1, \varepsilon_2) \in A$ and $\boldsymbol{T'} := 1/T$ instead of the design variables $\chi(\varepsilon_1, \varepsilon_2)$, $(\varepsilon_1, \varepsilon_2) \in A$ and T. Third, the formulation that is proposed in this paper has additional constraints, e.g., the stability constraints (8e) and the well-posedness constraints (8j). Fourth, the objective functions that we consider cannot be written as the weighted sum that is assumed in the cycle periodicity formulation of the PESP.

According to the literature (Peeters 2003; Liebchen 2007; Wünsch and Köhler 1990) the cycle periodicity formulation is a more efficient formulation than the original formulation. This is in accordance with our finding in the upcoming section that the formulation that is proposed in this paper is more efficient than the formulation of the currently existing group-based approaches.

3.4 Numerical Comparison of the Approaches

In this section we perform an extensive numerical comparison of the currently existing group-based approaches with the novel approach that is proposed in this paper. 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 Lefeber 2016). We have grouped these test cases according to their sizes, see also Table 4. The first five test cases (S1, S2, S3, S4 and S5) concern T-junctions with only six signal groups and are considered to be small. Test cases M1, M2, M3 and M4 concern intersections with 9-15 signal groups and are considered to have a medium size. For the last 4 test cases (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 considerer twelve different optimization problems, which are depicted in Table 4. 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 (8) by multiplying the left-hand-side of the stability constraint (8e) with $\boldsymbol{\beta}$ and maximizing $\boldsymbol{\beta}$. 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, which we also obtain 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 (2b). For each signal group we introduce an auxiliary variable $\boldsymbol{d}_i$, which denotes the contribution of signal group i to the average delay:

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

Since we minimize the average delay at the intersection, we take w_q proportional to the arrival rate λ_q, i.e., $w_q = \lambda_q/\Lambda$. Let the period duration be fixed to T seconds. The variable $\boldsymbol{d}_i$ is then only a function of the variable $\boldsymbol{\gamma}(\textcolor{gray}{\textcircled{i}}, \textcircled{i})$. We approximate the contribution to the average delay $\boldsymbol{d}_i$ by a piecewise linear function. To this end, we include linear inequalities of the form: $\boldsymbol{d}_i \geq a\boldsymbol{\gamma}(\textcolor{gray}{\textcircled{i}}, \textcircled{i}) + b$. The resulting optimization problem is then an MILP problem. We define this piecewise linear function as follows. Note that each feasible effective red time $\boldsymbol{\gamma}(\textcolor{gray}{\textcircled{i}}, \textcircled{i})T$ must be included in the following interval:

$$[\lfloor\max\{\underline{r}_i, T-\overline{g}_i\}\rfloor, \lceil\min\{\overline{r}_i, T-\underline{g}_i\}\rceil] \cap [0, T-\max_{q\in\mathcal{Q}_i}\rho_q T).$$

We obtain the approximated delay for each integral-valued effective red time $\boldsymbol{\gamma}(\textcolor{gray}{\textcircled{i}}, \textcircled{i})T$ that is included in this interval; we approximate the delay linearly between each two such subsequent points. The resulting piecewise linear function is exact for each of these integral-valued effective red times and it is linear between these integral-valued effective red times, see also Figure 5. Note that we exclude a duration of $T - \max_{q\in\mathcal{Q}_i}\rho_q T$ seconds as the delay $\boldsymbol{d}_i$ is infinite for this effective red time. Furthermore, note that the above interval might contain only one integral value. For our numerical study however, this interval always includes at least two integral values. The goal is to minimize $\sum_{i\in\mathcal{S}}\boldsymbol{d}_i$.

In Table 4 we give the results (objective values) of all 13×12 optimization problems and in Table 5 we give the sizes of these optimization problems. In Table 6 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). To obtain these results we have solved each optimization problem 100 times and obtained the average

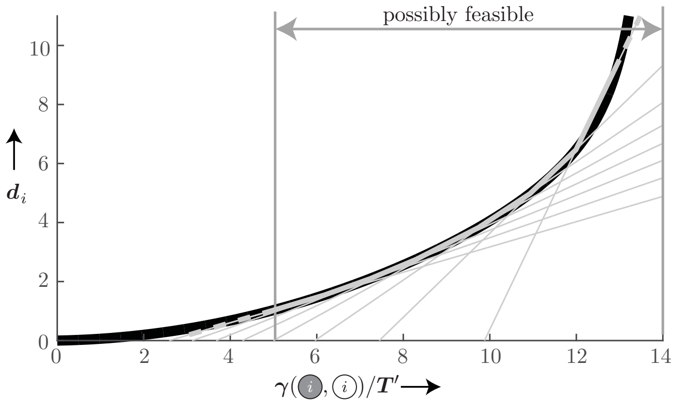


Figure 5: Piecewise linear approximation (grey) of the delay equation $\boldsymbol{d}_i$ of Van den Broek (2b) *(black) when $\mathcal{Q}_i = \{1\}$, $\rho_i = 0.3$, $w_i = 1$, $\mu_i = 0.5$, $T = 20$, $\underline{g}_i = 5$, $\underline{r}_i = 5$, and $\overline{g}_i = \overline{r}_i = \infty$ seconds. The interval of possibly feasible values for the effective red time is then* $[5, 14)$*. The piecewise linear function is exact for the following durations:* $5, 6, 7, \ldots, 13$*. The light gray lines visualize the tangent of the piecewise linear function.*

computation times over these 100 runs. The results are obtained on a computer with specifications: Intel i5-4300U CPU @1.90GHZ with 16.0GB of RAM. From Table 6 we conclude that the novel approach performs better when minimizing the period duration of the fixed-time schedule and when minimizing the average delay that road users experience at the intersection. When maximizing the capacity of the intersection, then the performance of both methods is comparable. The difference in computation times is the smallest when maximizing the capacity and 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 0.056, which implies a reduction in computation times of $(1 - 0.056)100 = 94.4$ percent and a speed up of a factor $1/0.056 \approx 18$. 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.

4 Conclusions and Recommendations

This paper presents a novel group-based approach to optimize fixed-time schedules for isolated intersections. With this approach, the structure of the fixed-time schedule 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

Objective	$\min T$						$\max \beta$	$\min D$				
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	47.93	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	∞	∞	∞	∞	∞	1.02	86.37	64.44	58.27	57.16	58.10
M4	80.01	85.81	92.53	100.38	109.69	∞	1.25	47.71	36.82	33.24	31.95	31.71
L1	74.71	78.10	81.81	86.37	∞	∞	1.20	40.48	38.12	38.56	39.46	40.58
L2	83.59	89.83	97.07	105.58	115.73	∞	1.21	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 4: 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* ($\max \beta$) *or minimizing the average delay* ($\min D$)*. In case that the average delay is minimized, then 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, then 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.	Eq.	Int.	Cont.	Ineq.	Eq.	Int.	Cont.	Ineq.	Eq.
S1	1 (6)	25 (13)	36 (38)	13 (0)	1 (6)	26 (14)	36 (38)	13 (0)	1 (6)	30 (18)	171-234 (173-236)	13 (0)
S2	1 (6)	25 (13)	36 (38)	13 (0)	1 (6)	26 (14)	36 (38)	13 (0)	1 (6)	30 (18)	151-210 (153-212)	13 (0)
S3	1 (6)	25 (13)	36 (38)	13 (0)	1 (6)	26 (14)	36 (38)	13 (0)	1 (6)	30 (18)	223-304 (225-306)	13 (0)
S4	4 (9)	31 (13)	42 (44)	19 (0)	4 (9)	32 (14)	42 (44)	19 (0)	4 (9)	36 (18)	381-518 (383-520)	19 (0)
S5	4 (9)	31 (13)	42 (44)	19 (0)	4 (9)	32 (14)	42 (44)	19 (0)	4 (9)	36 (18)	368-502 (370-504)	19 (0)
M1	10 (18)	55 (19)	72 (74)	37 (0)	10 (18)	56 (20)	72 (74)	37 (0)	10 (18)	63 (27)	374-508 (376-510)	37 (0)
M2	10 (18)	55 (19)	72 (74)	37 (0)	10 (18)	56 (20)	72 (74)	37 (0)	10 (18)	63 (27)	385-526 (387-528)	37 (0)
M3	22 (35)	99 (29)	126 (128)	71 (0)	22 (35)	100 (30)	126 (128)	71 (0)	22 (35)	114 (44)	1966-2669 (1968-2671)	71 (0)
M4	22 (35)	99 (29)	126 (128)	71 (0)	22 (35)	100 (30)	126 (128)	71 (0)	22 (35)	114 (44)	1157-1569 (1159-1571)	71 (0)
L1	44 (70)	195 (55)	248 (250)	141 (0)	44 (70)	196 (56)	248 (250)	141 (0)	44 (70)	221 (81)	2127-2882 (2129-2884)	141 (0)
L2	44 (70)	195 (55)	248 (250)	141 (0)	44 (70)	196 (56)	248 (250)	141 (0)	44 (70)	221 (81)	2352-3201 (2354-3203)	141 (0)
L3	45 (71)	201 (59)	258 (260)	145 (0)	45 (71)	202 (60)	258 (260)	145 (0)	45 (71)	229 (87)	2250-3027 (2252-3029)	145 (0)
L4	45 (71)	201 (59)	258 (260)	145 (0)	45 (71)	202 (60)	258 (260)	145 (0)	45 (71)	229 (87)	2137-2897 (2139-2899)	145 (0)

Table 5: 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 that is given in this paper. When minimizing the delay ($\min D$*), then 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.*

		$\min T$			$\max \beta$			$\min D$		
Solver	Intersections	GB (s)	novel (s)	novel/GB (s)	GB (s)	novel (s)	novel/GB (s)	GB (s)	novel (s)	novel/GB (s)
CPLEX	Small	0.085	0.068	**0.792**	0.083	0.080	**0.959**	0.149	0.087	**0.582**
	Medium	0.108	0.079	**0.737**	0.094	0.100	1.063	0.390	0.182	**0.467**
	Large	0.458	0.209	**0.456**	0.296	0.296	**0.999**	8.559	0.877	**0.102**
GUROBI	Small	0.003	0.002	**0.771**	0.004	0.003	**0.717**	0.015	0.006	**0.391**
	Medium	0.020	0.015	**0.772**	0.013	0.021	1.602	0.296	0.070	**0.235**
	Large	0.721	0.237	**0.329**	0.441	0.236	**0.536**	7.859	0.928	**0.118**
SCIP	Small	0.027	0.020	**0.768**	0.039	0.021	**0.528**	0.074	0.033	**0.438**
	Medium	0.104	0.057	**0.544**	0.097	0.102	1.049	1.397	0.185	**0.133**
	Large	0.955	0.612	**0.641**	0.828	1.083	1.307	53.168	2.955	**0.056**
Average				**0.622**			**0.916**			**0.215**

Table 6: 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 have distinguished 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.*

design variables are used to model the periodicity of the fixed-time schedule.

Possible objective functions of the optimization problem are: minimizing the period duration of the fixed-time schedule, 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 traffic light under fixed-time control. Examples of approximations that are convex are the ones from (Miller 1963; van den Broek et al. 2006; Webster 1958). Whenever a convex objective function is used, then the optimization problem can be solved to optimality.

For the currently existing group-based approaches, multiple solutions may be associated with the same fixed-time schedule; all these solutions have different values for the binary-valued design variables. The formulation that is proposed in this paper, which aggregates the binary variables of the currently existing group-based approaches, does not posses this undesirable property. With an extensive computational study we have compared the currently existing group-based approaches with the novel proposed method. Both approaches seem to have comparable performance when we maximize the capacity of the intersection. When minimizing the period duration of the fixed-time schedule or minimizing the delay that road users experience at the intersection, then the novel approach is superior.

As this optimization problem seems very promising, the next step is to experiment with different cycle bases to further reduce the computation times. Furthermore, this approach can be extended in several ways. First of all, this approach can be extended so that the lane markings can also be optimized. Another extension is to allow a signal group to have multiple effective green intervals and optimize the number of green intervals that each traffic light receives during a period, which in this paper was assumed to be one; both these extensions are topics of papers to come.

A A Motivation for Negative Clearance Times

In this appendix we motivate the use of negative minimum clearance times; when the minimum clearance time $\underline{c}_{i,j}$, $\{i,j\} \in \Psi_{\mathcal{S}}$ is negative, then 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. Consider queue 1 and queue 2 of the intersection in Figure 6. 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 (queue 2 then becomes effective green during an effective green interval of queue 1). In other words, 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.

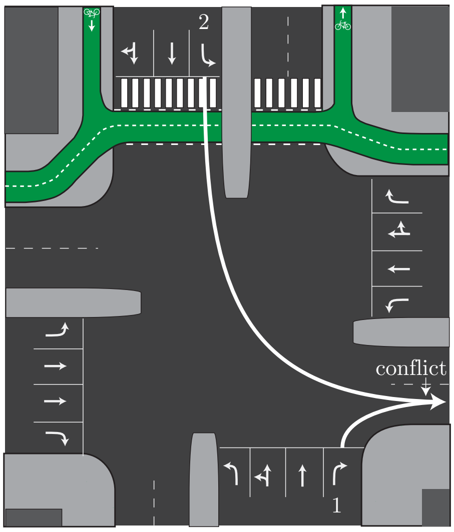


Figure 6: An intersection for which a negative clearance time between queue 1 *and queue* 2 *is motivated.*

B 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, then the arrival rate and saturation flow rate 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 $r'_i := \boldsymbol{\gamma}(\textcircled{i}, \textcircled{i})$. The approximation of van den Broek for a queue $q \in \mathcal{Q}_i$ can be written as:

$$d_q = \frac{r'_i}{2(1-\rho_q)\rho_q}\left(\frac{\sigma_q^2}{\mu_q(1-\rho_q)} + \frac{\rho_q r'_i}{\boldsymbol{T}'} + \frac{r'_i \rho_q^2 \sigma_q^2}{\mu_q(1-r'_i)^2(1-r'_i-\rho_q)(1-\rho_q)}\right).$$

Note that the stability constraint (8e), the strictly positive red times (8d) and the circuital constraint (8i) imply $0 < \rho_q \leq 1 - r'_i < 1$ and note that $\sigma_q, \boldsymbol{T}' \geq 0$. 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 $\frac{\partial^2 d_q}{\partial \boldsymbol{T}'^2} \geq 0$, where H is the Hessian of this equation. We show these inequalities below:

$$\begin{aligned}
\frac{\partial^2 d_q}{\partial \boldsymbol{T}'^2} &= \frac{r_i'^2}{\boldsymbol{T}'^3(1-\alpha_q+r'_i)} \geq 0, \\
\det(H) &= \frac{\alpha_q \sigma_q^2 r_i'^2\left(\left((1-\alpha_q)+r'_i\right)^2 + 2(1-\alpha_q)^2 r'_i\right)}{\mu_q(1-\alpha_q)^3(1-\alpha_q+r'_i)^3(1-r'_i)^4\boldsymbol{T}'^3} \geq 0,
\end{aligned}$$

C Proof of Lemma 1

Lemma 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 that is calculated with* (6)*. Define $\mathcal{B} = \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ 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 that is obtained from $\mathcal{B}$ when, for each conflict $\{i, j\} \notin \mathcal{F}'$, $i < j$ we replace the cycle $\mathcal{C}_{\mathcal{F}}((\colorbox{gray}{j}, \textcircled{i}))$ by the cycle:*

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

The set $\mathcal{B}'$ is an integral cycle basis of the constraint graph G that includes all the cycles that are associated with the circuital constraints (3g)–(3h)*.*

Proof. The SFCB $\mathcal{B}$ that is obtained from the spanning forest $\mathcal{F}$ includes the cycles that are associated with the circuital constraints (3g); each arc $(\colorbox{gray}{$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 the circuital constraint (3h). Consider a conflict $\{i, j\} \in \Psi_{\mathcal{S}}$ and assume w.l.o.g. that $i < j$. The cycle that is associated with the circuital constraint (3h) of the conflict $\{i, j\} \in \Psi_{\mathcal{S}}$ is the cycle $\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \colorbox{gray}{i}), (\colorbox{gray}{i}, \textcircled{j}), (\textcircled{j}, \colorbox{gray}{j}), (\colorbox{gray}{j}, \textcircled{i})\}$. This cycle is included in the SFCB if and only if $\{i, j\} \in \mathcal{F}'$; this is then the cycle $\mathcal{C}_{\mathcal{F}}((\colorbox{gray}{j}, \textcircled{i}))$. Consider a conflict $\{i, j\} \in \Psi_{\mathcal{S}}$ for which $\{i, j\} \notin \mathcal{F}'$ and assume w.l.o.g. that $i < j$. For each such conflict we replace the cycle $\mathcal{C}_{\mathcal{F}}((\colorbox{gray}{j}, \textcircled{i}))$ by the cycle $\mathcal{C} = \mathcal{C}^+ = \{(\textcircled{i}, \colorbox{gray}{i}), (\colorbox{gray}{i}, \textcircled{j}), (\textcircled{j}, \colorbox{gray}{j}), (\colorbox{gray}{j}, \textcircled{i})\}$. The resulting set $\mathcal{B}'$ includes all the cycles that are associated with the circuital constraints (3g)–(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 $\mathcal{B}_k := \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ by performing one additional replacement. Let $\{i, j\} \in \Psi_{\mathcal{S}}$ 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 $\mathcal{B}$ if follows that $\mathcal{C}_{\mathcal{F}}((\colorbox{gray}{i}, \textcircled{j})) \in \mathcal{B}$ and that $\mathcal{C}_{\mathcal{F}}((\colorbox{gray}{j}, \textcircled{i})) \in \mathcal{B}$. The previous $k-1$ replacements did not affect these cycles and, as a consequence, $\mathcal{C}_{\mathcal{F}}((\colorbox{gray}{i}, \textcircled{j})) \in \mathcal{B}_k$ and $\mathcal{C}_{\mathcal{F}}((\colorbox{gray}{j}, \textcircled{i})) \in \mathcal{B}_k$. Assume w.l.o.g. that $\mathcal{C}_{d-1} := \mathcal{C}_{\mathcal{F}}((\colorbox{gray}{i}, \textcircled{j}))$ and that $\mathcal{C}_d := \mathcal{C}_{\mathcal{F}}((\colorbox{gray}{j}, \textcircled{i}))$. Furthermore, assume w.l.o.g. that $i < j$; we then replace the cycle $\mathcal{C}_d$. From the induction hypothesis it follows that $\mathcal{B}_k := \{\mathcal{C}_1, \ldots, \mathcal{C}_d\}$ 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:

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

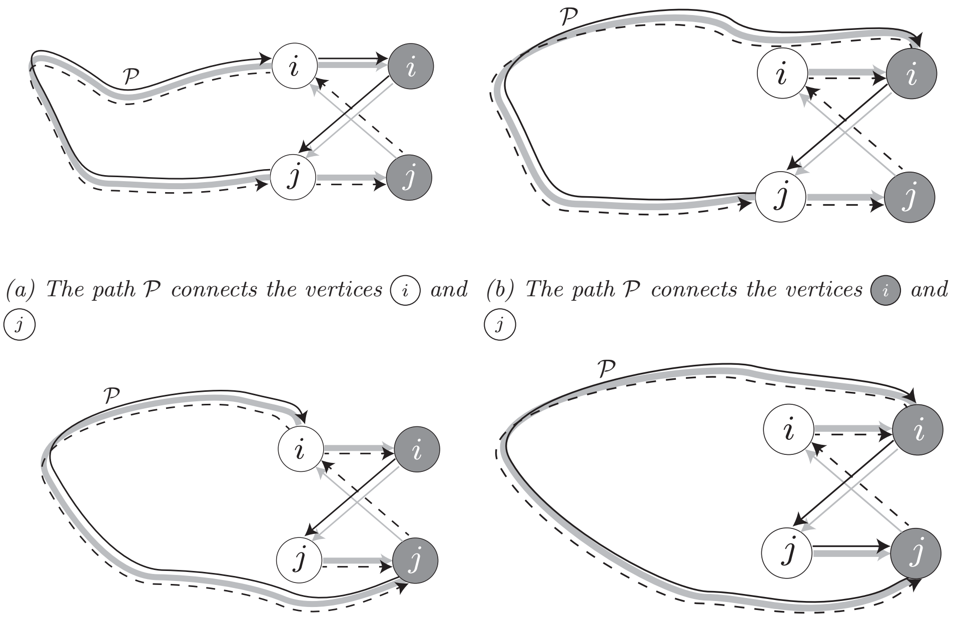
The cycles $\mathcal{C}_{d-1}$ and $\mathcal{C}_d$ are visualized in Figure 7. Let $\mathcal{C}'_d$ be the cycle that is associated with the circuital constraint (3h) of the conflict $\{i, j\} \in \Psi_{\mathcal{S}}$, i.e.,

$$\mathcal{C}'_d := \mathcal{C}'^+_d = \{(\textcircled{i}, \colorbox{gray}{i}), (\colorbox{gray}{i}, \textcircled{j}), (\textcircled{j}, \colorbox{gray}{j}), (\colorbox{gray}{j}, \textcircled{i})\}.$$

Note that the cycle-arc incidence vector of the cycle $\mathcal{C}'_d$ satisfies $C'_d := C_{d-1} + C_d$, see Figure 7; 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 $a \notin \mathcal{C}'_d$ that is 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:

$$\begin{aligned} C &= \alpha_1 C_1 + \ldots + \alpha_d C_d, \\ &= \alpha_1 C_1 + \ldots + \alpha_{d-2} C_{d-2} + (\alpha_{d-1} - \alpha_d) C_{d-1} + \alpha_d C'_d, \\ &= \alpha'_1 C_1 + \ldots + \alpha'_{d-1} C_{d-1} + \alpha'_d C'_d, \end{aligned}$$

which implies that we can write each cycle 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.



(a) The path $\mathcal{P}$ connects the vertices i and j

(b) The path $\mathcal{P}$ connects the vertices i and j

(c) The path $\mathcal{P}$ connects the vertices i and j

(d) The path $\mathcal{P}$ connects the vertices i and j

Figure 7: These figures visualize the cycles $\mathcal{C}_{d-1} := \mathcal{C}_{\mathcal{F}}((i, j))$ and $\mathcal{C}_d := \mathcal{C}_{\mathcal{F}}((j, i))$. The solid black lines visualize 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 the constraint graph G; the arcs that are in the spanning forest $\mathcal{F}$ are visualized in bold. The 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 $\mathcal{P}$.

□

D Proof of Lemma 2

Lemma 2. *Consider a cycle basis $\mathcal{B} = \{\mathcal{C}_1, \dots, \mathcal{C}_d\}$ that is obtained by using Lemma 1. Consider a cycle $\mathcal{C} \in \mathcal{B}$ that does not equal the circuital constraint* (8i)*; for the circuital constraint* (8i) *we know that the multiplicity $\boldsymbol{z}_{\mathcal{C}}$ equals one. For each such cycle, we can relate its integral-valued design variable $\boldsymbol{z}_{\mathcal{C}}$ to the binary design variables $\boldsymbol{\Omega}_{i,j}$ as follows:*

$$\boldsymbol{z}_{\mathcal{C}} = \sum_{(\colorbox{gray}{i},\textcircled{j})\in\mathcal{C}^+} \boldsymbol{\Omega}_{i,j} - \sum_{(\colorbox{gray}{i},\textcircled{j})\in\mathcal{C}^-} \boldsymbol{\Omega}_{i,j}. \tag{12}$$

Proof. For a cycle $\mathcal{C} \in \mathcal{B}$ that is associated with the circuital constraint (8h), the equation to prove, reduces to:

$$\boldsymbol{z}_{\mathcal{C}} = \boldsymbol{\Omega}_{i,j} + \boldsymbol{\Omega}_{j,i} = 1,$$

which is indeed true; for these cycles we fix the value of $\boldsymbol{z}_{\mathcal{C}}$ to one.

Now consider some other cycle $\mathcal{C} \in \mathcal{B}$ that is not associated with the circuital constraint (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:

$$\colorbox{gray}{i_1}, \textcircled{i_2}, \colorbox{gray}{i_2}, \textcircled{i_3}, \colorbox{gray}{i_3} \dots, \textcircled{i_N}, \colorbox{gray}{i_N}, \tag{13}$$

where $v_1 = v_N$ and $N > 3$. The equation to prove (12), then reduces to:

$$\boldsymbol{z}_{\mathcal{C}} = \sum_{k=1}^{N-1} \boldsymbol{\Omega}_{i_k,i_{k+1}}.$$

The cycle periodicity constraint (8g) of the circuit (13) gives the following expression for $\boldsymbol{z}_{\mathcal{C}}$:

$$\begin{aligned} \boldsymbol{z}_{\mathcal{C}} &= \sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}^+} \boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) - \sum_{(\varepsilon_1,\varepsilon_2)\in\mathcal{C}^-} \boldsymbol{\gamma}(\varepsilon_1, \varepsilon_2) \\ &= \sum_{k=1}^{N-1} \gamma(i_k, i_{k+1})', \end{aligned} \tag{14}$$

where

$$\gamma(i,j)' := \boldsymbol{\gamma}(\textcircled{i}, \colorbox{gray}{i}) + \boldsymbol{\gamma}(\colorbox{gray}{i}, \textcircled{j}).$$

By combining circuital constraint (8h) with the well-posedness constraint(8j) we find that:

$$0 < \gamma(i_k, i_{k+1})' < 1, \qquad k = 1, \dots, N-1.$$

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 $f(i) := f(\textcircled{$i$})$, which is the time (expressed as a fraction of the period duration) at which signal group i becomes effective green. Recall that $\boldsymbol{\Omega}_{i,j}$ is defined such that $\boldsymbol{\Omega}_{i,j} = 0$ if $\boldsymbol{f}(\textcircled{i}) \leq \boldsymbol{f}(\textcircled{j})$ and $\boldsymbol{\Omega}_{i,j} = 1$ otherwise. Since, $\gamma(i_k, i_{k+1})' \in (0, 1)$ it holds that:

$$\gamma(i_k, i_{k+1})' = f(i_{k+1}) - f(i_k) + \boldsymbol{\Omega}_{i_k, i_{k+1}},$$

Substituting this equation in (14) and using $v_N = v_1$ gives:

$$\boldsymbol{z}_{\mathcal{C}} = \sum_{k=1}^{N-1} \boldsymbol{\Omega}_{i_k, i_{k+1}},$$

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 (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:

$$v_1, v_2, v_3, v_4, \ldots, v_N, \tag{15}$$

where $(v_1, v_2) = a$ 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 (15) does not equal the desired sequence (13). Let v_k be the first vertex in the sequence (15) that breaks this desired sequence; note that $k > 2$ as $(v_1, v_2) := (\colorbox{gray}{i}, \textcircled{j})$ for some conflict $\{i, j\} \in \Psi_{\mathcal{S}}$. Consider the case that $v_{k-1} = \colorbox{gray}{i}$ 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 the vertex v_{k-1}; however, all these adjacent vertices satisfy the desired sequence (see Figure 8), which contradicts the definition of v_k. Therefore, it must hold that:

$$v_{k-1} = \textcircled{i} \qquad \text{for some } i \in \mathcal{S}.$$

Since the vertex v_k is the first vertex that breaks the desired sequence (13), it holds that:

$$v_{k-2} = \colorbox{gray}{j} \qquad \text{for some } j \neq i.$$

The vertex v_{k-1} is only connected (via an arc) with vertices that represent a switch to effective red. Therefore, we have:

$$v_k = \colorbox{gray}{j'} \qquad \text{for some } j'.$$

By definition the vertex v_k does not satisfy the sequence (13), which implies $j' \neq i$. Furthermore, the sequence (15) 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 $(\textcircled{i}, \colorbox{gray}{i})$, $i \in \mathcal{S}$ (see its definition (6)). Therefore, the spanning forest $\mathcal{F}$ includes the arcs $(\textcircled{i}, \colorbox{gray}{i})$, $(\colorbox{gray}{$i$}, \textcircled{j'})$ and $(\colorbox{gray}{$j'$}, \textcircled{j'})$. Since $\mathcal{F} \cup \{a\}$ contains a single circuit this implies that:

$$(\colorbox{gray}{i}, \textcircled{j'}) \notin \mathcal{F} \cup \{a\},$$

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:

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

Note that the circuit $\mathcal{C}''$ is the cycle that is associated with the circuital constraint (8h) of the conflict $\{i, j'\} \in \Psi_{\mathcal{S}}$. Define the spanning forest $\tilde{\mathcal{F}}$ as follows:

$$\tilde{\mathcal{F}} = F \setminus \{(\colorbox{gray}{j'}, \textcircled{i})\} \cup \{(\colorbox{gray}{i}, \textcircled{j'})\}.$$

Thus, we obtain the spanning forest $\tilde{\mathcal{F}}$ by replacing the arc $(\colorbox{gray}{$j'$}, \textcircled{i})$ of spanning forest $\mathcal{F}$ by the arc $(\colorbox{gray}{$i$}, \textcircled{j'})$. Let C be the cycle-arc incidence vector that is 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 that is associated with the circuit $\mathcal{C}' := \mathcal{C}_{\tilde{\mathcal{F}}}(a)$. These cycle-arc incidence vectors satisfy (see also Figure 9):

$$C' = C + C'',$$

As a consequence,

$$\boldsymbol{z}_{\mathcal{C}'} = C'\boldsymbol{\gamma} = (C + C'')\boldsymbol{\gamma} = \boldsymbol{z}_{\mathcal{C}} + \boldsymbol{z}_{\mathcal{C}''} = \boldsymbol{z}_{\mathcal{C}} + 1.$$

Furthermore, it holds that:

$$\sum_{(\colorbox{gray}{i}, \textcircled{j}) \in \mathcal{C}'^{+}} \boldsymbol{\Omega}_{i,j} - \sum_{(\colorbox{gray}{i}, \textcircled{j}) \in \mathcal{C}'^{-}} \boldsymbol{\Omega}_{i,j} = \sum_{(\colorbox{gray}{i}, \textcircled{j}) \in \mathcal{C}^{+}} \boldsymbol{\Omega}_{i,j} - \sum_{(\colorbox{gray}{i}, \textcircled{j}) \in \mathcal{C}^{-}} \boldsymbol{\Omega}_{i,j} + 1,$$

which follows from the fact that the cycle $\mathcal{C}'$ uses the arc $(\colorbox{gray}{$i$}, \textcircled{j'})$ in the forward direction instead of the arc $(\colorbox{gray}{$j'$}, \textcircled{i})$ in the backward direction; as a consequence, the term $-\boldsymbol{\Omega}_{j',i} = \boldsymbol{\Omega}_{i,j'} - 1$ is replaced by the term $\boldsymbol{\Omega}_{i,j'}$. Therefore, circuit $\mathcal{C}_{\mathcal{F}}(a)$ satisfies (12) if and only if the circuit $\mathcal{C}_{\tilde{\mathcal{F}}}(a)$ satisfies (12). Furthermore, by its definition, the circuit $\mathcal{C}_{\tilde{\mathcal{F}}}(a)$ can be associated with a sequence of vertices:

$$v_1, v_2, \ldots, v_{k-1}, v'_k, v'_{k+1}, v'_{k+2} \ldots, v'_N. \tag{16}$$

where $(v_1, v_2) = a$, $v'_k = \colorbox{gray}{$i$}$ and $v'_{k+1} = \textcircled{j'}$. Thus, this sequence satisfies the desired sequence (13) 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 (12) if and only if the cycle $\mathcal{C} := \mathcal{C}_{\mathcal{F}}(a)$ satisfies (12). 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. □

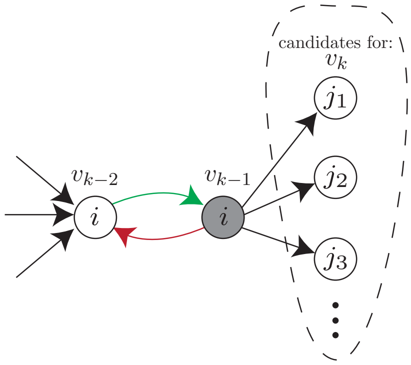
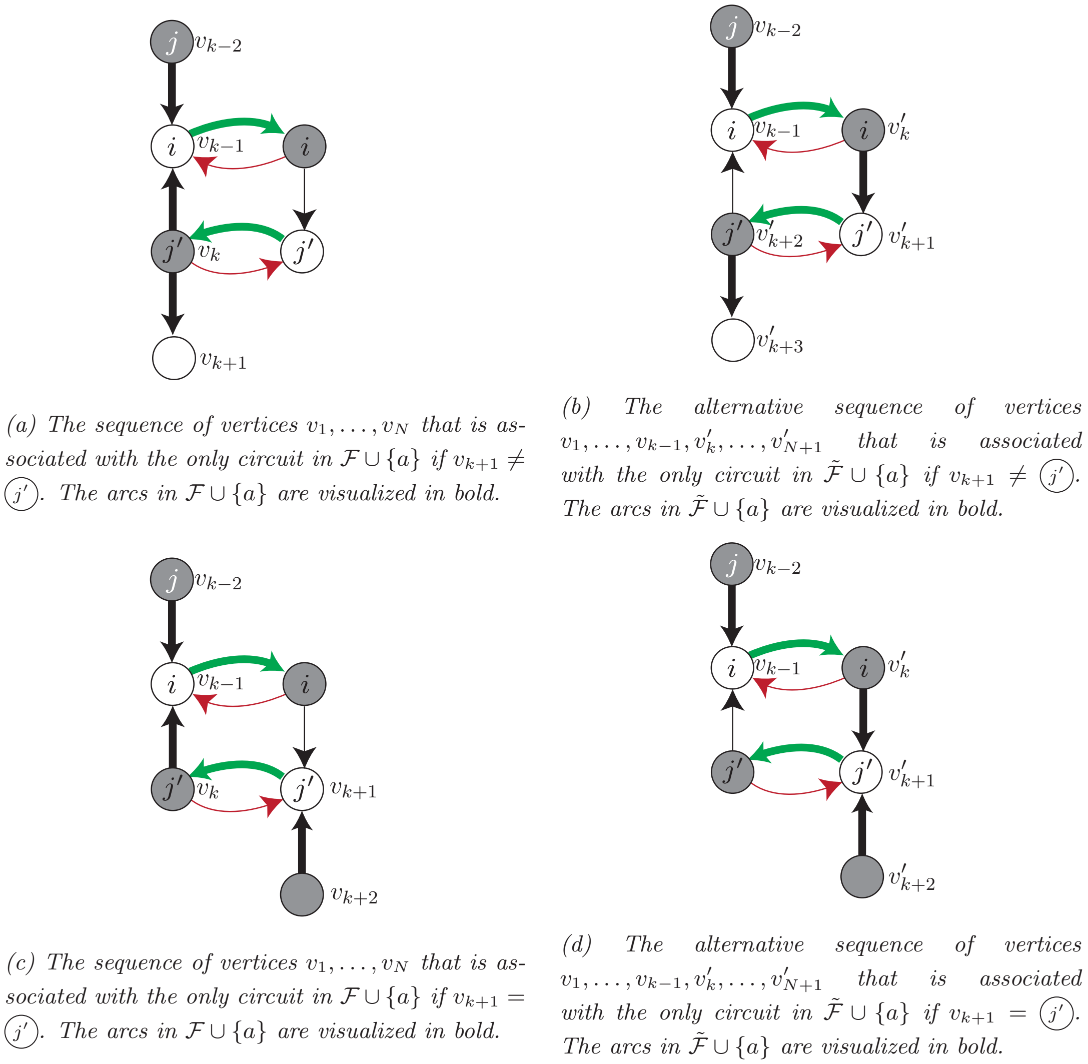


Figure 8: Visualization of the following statement: for a circuit in the constraint graph G that is associated with the periodic order $v_1, v_2, \ldots, v_N$, $N > 3$ the vertex v_k represents a switch to effective green when the vertices v_{k-2} and v_{k-1} represent the switch to effective green respectively 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 $v_1, \ldots, v_N$ that is associated with the only circuit in $\mathcal{F} \cup \{a\}$ if $v_{k+1} \neq$ j'. The arcs in $\mathcal{F} \cup \{a\}$ are visualized in bold.

(b) The alternative sequence of vertices $v_1, \ldots, v_{k-1}, v'_k, \ldots, v'_{N+1}$ that is associated with the only circuit in $\tilde{\mathcal{F}} \cup \{a\}$ if $v_{k+1} \neq$ j'. The arcs in $\tilde{\mathcal{F}} \cup \{a\}$ are visualized in bold.

(c) The sequence of vertices $v_1, \ldots, v_N$ that is associated with the only circuit in $\mathcal{F} \cup \{a\}$ if $v_{k+1} =$ j'. The arcs in $\mathcal{F} \cup \{a\}$ are visualized in bold.

(d) The alternative sequence of vertices $v_1, \ldots, v_{k-1}, v'_k, \ldots, v'_{N+1}$ that is associated with the only circuit in $\tilde{\mathcal{F}} \cup \{a\}$ if $v_{k+1} =$ j'. The arcs in $\tilde{\mathcal{F}} \cup \{a\}$ are visualized in bold.

Figure 9: 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 $v_1, \ldots, v_N$ where v_k is the first vertex that breaks the alternating sequence. In these figures, we visualize a part of the sequence $v_1, \ldots, v_N$ (Figure 9a and Figure 9c) and a part of an alternative sequence $v_1, \ldots, v_{k-1}, v'_k, \ldots, v'_{N+1}$ (Figure 9b and Figure 9d); the alternative sequence is associated with the only circuit in $\tilde{\mathcal{F}} \cup \{a\}$, where $\tilde{\mathcal{F}} := F \setminus \{(j', i)\} \cup \{(i, j')\}$ is a spanning forest.

References

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.

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.

Fleuren, S. and Lefeber, E. (2016). Data of real-life intersections for fixed-time traffic light control. `https://pure.tue.nl/ws/files/26651263/DataOfRealLifeIntersections.pdf`.

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.

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.

Gurobi Optimization, Inc. (2015). Gurobi optimizer reference manual.

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.

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.

International Business Machines Corp (2015). *IBM ILOG CPLEX Optimization studio cplex users manual*. `http://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.3`.

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.

Krijger, P. (2013). Traffic light prediction for tom tom devices. Master's thesis, Eindhoven University of Technology, the Netherlands.

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. (2007). Periodic timetable optimization in public transport. In *Operations research proceedings 2006*, Operations research proceedings, pages 29–36. Springer.

Liebchen, C. and Peeters, L. (2002). On cyclic timetabling and cycles in graphs. Technical report, Technical Report 761/2002, TU Berlin.

Lindner, T. (2000). Train schedule optimization in public rail transport. *Mathematics - Key Technology for the Future: Joint Projects Between Universities and Industry*, pages 703–716.

Miller, A. J. (1963). Settings for fixed-cycle traffic signals. *Operational Research Quarterly*, 14(4):373–386.

National Research Council (U.S.) (2010). *Highway Capacity Manual*. Transporation Research Board, Washington D.C.

Odijk, M. A. (1996). A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B: Methodological*, 30(6):455–464.

Peeters, L. (2003). *Cyclic railway timetable optimization*. PhD thesis, Erasmus Research Institute of Management (ERIM).

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.

Schrijver, A. and Steenbeek, A. G. (1993). Dienstregelingontwikkeling voor Nederlandse Spoorwegen NS rapport fase 1. *Centrum voor Wiskunde en Informatica (Oktober 1993)*.

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.

Silcock, J. P. (1997). Designing signal-controlled junctions for group-based operation. *Transportation Research Part A: Policy and Practice*, 31(2):157–173.

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 Leeuwaarden, 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 Zwieten, D. A. J. (2014). *Fluid flow switching servers: Control and observer design*. PhD thesis, Eindhoven University of Technology.

Webster, F. V. (1958). Traffic signal settings. Road Research Technical Paper 39, Great Britain Road Research Laboratory, Her Majesty's Stationery Office, London.

Wilson, A. (2006). *Handboek verkeerslichtenregelingen*, volume 2006. 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.

Wünsch, G. and Köhler (1990). *Coordination of traffic signals in networks*. 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.