Networks of fixed-cycle intersections

Document Version:
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Networks of fixed-cycle intersections

M.A.A. Boon   J.S.H. van Leeuwaarden

September 18, 2018

Abstract

We present an algorithmic method for analyzing networks of intersections with static signaling, with as primary example a line network that allows traffic flow over several intersections in one main direction. The method decomposes the network into separate intersections and treats each intersection in isolation using an extension of the fixed-cycle traffic-light (FCTL) queue. The network effects are modeled by matching the output process of one intersection with the input process of the next (downstream) intersection. This network analysis provides insight into wave phenomena due to vehicles experiencing progressive cascades of green lights and sheds light on platoon forming in case of imperfections. Our algorithm is shown to match results from extensive discrete-event simulations and can also be applied to more complex network structures.

Keywords: fixed-cycle traffic-light queue; performance evaluation; queueing theory; steady-state distribution; traffic engineering; transform solution; stochastic networks
AMS 2000 Subject Classification. 60E10, 60J10, 60K25, 68M20, 90B20

1 Introduction

Intersections are natural bottlenecks and crucially influence the dynamics of urban traffic. Traffic lights trigger a switching process meant to manage conflicting traffic flows. The coordination is sometimes done dynamically, according to sensor data of currently existing traffic flows; otherwise it is done statically, by the use of timers. While intersections can be studied in isolation [8, 10, 18, 22, 23], the larger picture of networks of multiple intersections is increasingly important, also in view of the rapid growth of urbanization [3, 7]. This paper contributes to the theoretical underpinning of traffic networks by extending classical models for isolated intersections to models for networks of intersections with static signaling.

Think of a series of traffic lights designed to let traffic flow over several intersections in one main direction. Any vehicle traveling along (at an approximate prescribed speed) wants to meet a progressive cascade of green lights, and not have to stop at intersections. In practical use, only a group of vehicles – referred to as *platoon* – can pass the intersection before the time band is interrupted to give way to other traffic flows. The platoon sizes are governed by the signal times. Our method to model such situations consists of two ingredients: An extension of a classical queueing model for one isolated intersection that can deal with correlated input and that allows for

*Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands. Email: m.a.a.boon@tue.nl and j.s.h.v.leeuwaarden@tue.nl

a detailed characterization of the output process of an intersection, and an algorithm for network analysis that decomposes the series of queues into multiple isolated queues. While interesting in their own right, and likely to find more applications in transportation science, network analysis of intersections requires the delicate combination of both ingredients. We now discuss each of them separately.

Queueing model for one intersection. The classical model for an isolated intersection that we adopt and extend in this paper is the fixed-cycle traffic-light (FCTL) queue; one of the most well-studied stochastic models in traffic engineering [8, 17, 19, 25]. Vehicles arrive to an intersection controlled by a traffic light and form a queue. The time scale is divided into time intervals of unit length, and the traffic light alternates between red and green periods of fixed durations r and g time units. Delayed vehicles depart during the green period, where it takes one time unit for each delayed vehicle to depart; departures thus occur at equally spaced times until either the queue dissipates or the green phase terminates. Darroch [8] obtained the probability generating function (pgf) of the steady-state overflow queue (the number of vehicles waiting in front of the traffic light at the end of a green period) and the pgf of the steady-state delay was obtained in van Leeuwaarden [23]. Hence, all information about the distribution of the steady-state overflow queue and steady-state delay in the FCTL queue can be obtained from the results in [8, 23], including all moments of the steady-state queue length and delay, and the distribution of the output process (the way vehicles leave the intersection). The output process of the first intersection is of crucial importance for the present paper, because it will serve as input process for some other signalized intersection. Moreover, the output process of a second intersection serves as input for a third intersection, and so forth. This network effect acts as a filter that modifies, and perhaps streamlines, the arrival process at consecutive intersections. Therefore, we shall address in this paper the technical challenge of extending the classical FCTL queue to allow for nonuniform and hence time-dependent correlated arrival processes. We call this extended model the generalized FCTL queue.

Network algorithm. A network of intersections with correlated input and output processes poses completely new challenges. Several approaches have been developed so far, but they are not based on the on the FCTL queue. Existing techniques to analyze isolated intersections are known to be inadequate in network settings, due to the highly correlated arrival process [9, 10]. Even techniques that are not based on a steady-state analysis, such as the probabilistic approach by Viti and Van Zuylen [24] that does not require arrivals in different time slots to be identically distributed, still assume independent arrivals. We therefore develop an approximation scheme to evaluate the system performance based on decomposition. While this approach has been successfully applied to classic queueing networks [4, 14, 26], a network of generalized FCTL queues poses additional challenges due to the non-synchronized cyclic structures and inherently correlated arrival processes. We decompose the network into isolated generalized FCTL queues, which are then analyzed separately by assuming specific arrival processes, and in particular the output process of one intersection serves as the input process of an upstream intersection, hence creating the correlation structure that comes with network topologies.

Osorio, Flötteröd and Bierlaire [20, 21] take a different approach for modeling correlated network structure, by extending the deterministic kinematic wave model with additional sources of randomness. Numerical evaluation of the models in [20, 21] allows to investigate the impact of the correlations structure on dynamic build-up, spill back and dissipation of queues. In contrast, our network model does not depart from the deterministic kinematic wave model (that comes

with differential equations), but rather builds on the FCTL queue and starts from a model formulation in terms of a multi-dimensional Markov process. Although it is possible to use our Markov network model for numerically calculating transient effects like dynamic build-up and spill back, the present paper develops the mathematical methods for an exact steady-state analysis.

Outline of the paper. In Section 2 we provide a detailed model description of the generalized FCTL queue. In Section 3 we present the full analytic solution of the generalized FCTL queue, both in terms of a formal characterization of the probability generating functions of the queue length distribution, and in terms of practically implementable algorithms for calculating the queue length distribution, for any given correlated arrival pattern. In Section 4 we design the network algorithm based on decomposition and the results in Section 3. We also compare our analytical results with extensive discrete-event simulation of the same network model. In Section 5 we present conclusions.

2 FCTL queue with correlated arrivals

We now present a generalization of the classical FCTL queue that can deal with correlated arrivals. In Subsection 2.1 we detail the model and its underlying assumptions. We then discuss in Subsection 2.2 an example of an arrival pattern that contains some crucial features than are anticipated in network settings. The numerical calculations for that example were performed with the algorithmic method developed in Section 3.

2.1 Model description

The first two model assumptions are adopted from the classical FCTL queue [23]:

Assumption 2.1 (Discrete-time assumption) The time axis is divided into constant time intervals of unit length, so-called slots, where each slot corresponds to the time needed for a delayed vehicle to depart from the queue. The green and red periods, and thus the cycle time c, are assumed to be fixed multiples of one slot. Hence, g, r, c are integers expressed in slots. Those vehicles that arrive to the queue and are delayed, join the queue at the end of the slot in which they arrive.

Assumption 2.2 (FCTL assumption) For those cycles in which the queue clears before the green period terminates, all vehicles that arrive during the residual green period pass through the system and experience no delay whatsoever.

The FCTL assumption lets vehicles that arrive during the residual green period pass the intersection without slowing down, and therefore the discharge rate of these vehicles is larger than the discharge rate of the delayed vehicles (one per time unit). Because of the huge difference in discharge rates of delayed vehicles (these vehicles have to accelerate) and non-delayed vehicles, the FCTL assumption is a sensible assumption. The next assumption is new:

Assumption 2.3 (Correlated arrivals assumption) Let $Y_{i,n}$ denote the number of vehicles that arrive to the intersection during slot i in cycle n. The random variables $Y_{i,n}$ are allowed to be dependent within cycle n, but we assume that $Y_{i,n}$ and $Y_{i,m}$ are independent when $n \neq m$.

Notice that Assumptions 2.1 and 2.3 together make that the queue lengths at the end of time slots can be modeled as a discrete-time Markov chain. This feature is exploited in Section 3 to find a fully analytic characterization for the steady-state queue length distribution.

3

It is crucial that Assumption 2.3 is less restrictive than its counterpart in [23] that assumes the $Y_{i,n}$ to be independent and identically distributed (i.i.d.). We need to move beyond this i.i.d. assumption in order to consider the correlated $Y_{i,n}$ sequences as they can occur in real network settings, for instance when the output process of one intersection (or FCTL queue) forms the input process for another intersection. Because the first intersection alters the original arrival pattern, the second intersection is likely to be confronted with platoons of vehicles that have been delayed by the upstream red signal.

The generalized FCTL queue defined by Assumptions 2.1-2.3 is in essence a queueing system with multiple customer types and batch arrivals, where $Y_{i,n}$ can be interpreted as the number of type i customers in batch n. Denote the pgf of the joint distribution of $(Y_{1,n}, Y_{2,n}, \ldots, Y_{c,n})$ by

$$Y_n(y_1, y_2, \ldots, y_c) = \mathbb{E}\Big[\prod_{i=1}^{c} y_i^{Y_{i,n}}\Big].$$

Although arrivals within a batch can be correlated, successive batches are i.i.d. Multi-type queueing models with batch arrivals have been well-studied [11, 15, 16]. A marked difference however, is that arrivals in the FCTL queue do not join the queue instantaneously, but are dictated when to arrive according to their type: type i arrivals join the queue in time slot i, for $i = 1, 2, \ldots, c$. Another crucial difference is Assumption 2.2, of course, which is very specific for traffic-light settings.

2.2 Motivating example

We now give an exemplary arrival pattern with features that are anticipated in network settings. We consider a generalized fixed-cycle traffic light queue with traffic arriving from one synchronized upstream traffic intersection. The cyclic arrival pattern at this queue, illustrated in Figure 1, starts with the arrival of a platoon of delayed vehicles from the major upstream flow. The settings are synchronized such that the signal turns green at the exact moment that the first car in this platoon arrives. This platoon is followed by a phase of free flow, with arriving vehicles that were not delayed at the upstream intersection. After a short period of no arrivals, we have a similar pattern of a platoon followed by free flow, arriving from a minor upstream flow. A more detailed description of the arrival process can be found in Appendix B.

In Section 4.2 we show that this specific arrival pattern arises naturally in network settings. Arrivals in the same cycle are correlated, but we assume independence between arrivals in successive cycles.
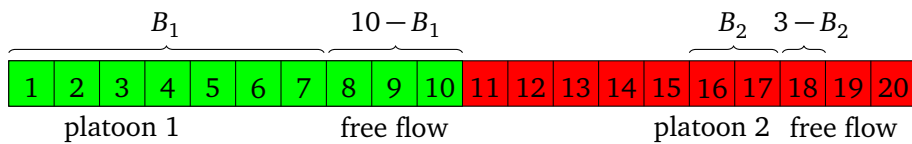

Figure 1: The arrival pattern in Section 2.2. The lengths of platoons 1 and 2 are respectively B_1 and B_2. The cycle length is $c = g + r = 10 + 10 = 20$ time slots.

Figure 2 shows the mean queue length during a cycle. The results from a microscopic discrete-event simulation of this generalized FCTL queue have been used to validate the analytic method.
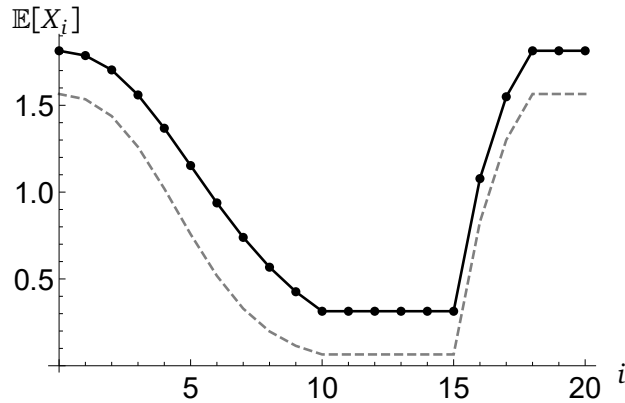
Figure 2: The queue length of the isolated intersection in Section 2.2. The black dots represent queue lengths obtained via a microscopic simulation of our model and the solid black line is obtained by the algorithm developed in Section 3. The gray dashed line is the traditional FCTL queue without correlated arrivals.

The longest queue occurs just before the traffic light turns green, and the queue is typically minimized at the end of green periods. The graph shows an interesting pattern, where the queue length remains constant for the first part of the red period, due to the absence of arrivals during that period. The arrival of platoon 2 results in a steep increase in queue length, starting after time slot 15. The shape of the graph in Figure 2 is different from the standard FCTL queue, which is typically more V-shaped due to the constant arrival rate throughout the cycle. To illustrate just how big the impact of correlation in the arrival pattern can be, we have also depicted the mean queue length computed using a classic FCTL analysis (cf. [23]). It is interesting to see that this traditional method severely underestimates the queue length, throughout the whole cycle. In Section 4 we show that this effect can be even bigger in a network of intersections.

The analytic results obtained in the next section allow us to determine the queue length distribution. Table 1 shows probabilities of the queue length exceeding certain levels during the cycle. We see that the probability that the queue exceeds five vehicles at the cycle start is 0.015. At the end of the green period this probability is only 0.002. At any arbitrary moment, the probability that there are more than five delayed vehicles is 0.008. Note that we use the term *queue length* to denote the number of vehicles in the queue. A related performance measure is the actual position of the back of the queue. Viti and Van Zuylen [24, Section 5] discuss how to determine the position of the last vehicle for the standard FCTL model using shockwave theory. This method can be extended to our model, but this is beyond the scope of the present paper.

X	$\mathbb{P}(X \geq 1)$	$\mathbb{P}(X \geq 2)$	$\mathbb{P}(X \geq 3)$	$\mathbb{P}(X \geq 4)$	$\mathbb{P}(X \geq 5)$	$\mathbb{P}(X \geq 6)$
X_0	0.829	0.547	0.302	0.075	0.036	0.015
X_{10}	0.159	0.089	0.042	0.014	0.006	0.002
$\bar{X}$	0.496	0.294	0.146	0.042	0.019	0.008

Table 1: The queue length distribution at the beginning of a cycle (X_0), at the end of a green period (X_{10}), and at arbitrary moments ($\bar{X}$). The notation in brackets will be introduced in Section 3.

3 Analytic solution and algorithms

We shall now show that the generalized FCTL queue with Assumptions 2.1-2.3 is analytically solvable. Using generating function techniques and complex analysis, we are able to obtain an explicit, analytic characterization for the steady-state queue length distribution. In order to do so, we exploit the Markovian nature of the queueing model at the level of cycles, and account for the correlations that occur within the cycle. The recursion relation that connects consecutive queue lengths during a cycle is given in Subsection 3.1, along with some further preliminaries. Then in Subsection 3.2 we derive the generating function for the steady-state queue lengths at the beginning of time slots throughout a cycle. In particular, we leverage this explicit generating function to create an algorithm for calculating the complete queue length distribution, at all points in time. Special attention is paid to the mean queue lengths in Subsection 3.3.

3.1 Preliminaries

Let $X_{i,n}$ denote the queue length at the end of the i-th time slot in the n-th cycle with $n = 1, 2, \ldots$. Let $Y_{i,n}$ denote the number of arrivals in time slot i during the n-th cycle, as defined in Assumption 2.1.

For convenience, define $X_{0,n+1}(z) := X_{c,n}(z)$ as the pgf of the queue length distribution at the start of cycle $n+1$. We will determine the the steady-state queue length, in the limit as $n \to \infty$, at the end of each time slot. Without loss of generality, we assume that the first g time slots in the cycle have a green signal, and the last r time slots are red. The queue length in this slot-based system then evolves according to

$$X_{k+1,n} = \begin{cases} X_{k,n} + Y_{k+1,n} - 1 & \text{if } X_{k,n} > 0 \text{ and } k = 0, \ldots, g-1, \\ 0 & \text{if } X_{k,n} = 0 \text{ and } k = 0, \ldots, g-1, \\ X_{k,n} + Y_{k+1,n} & \text{for } k = g, \ldots, c-1. \end{cases} \tag{3.1}$$

In the classical FCTL queue [8, 23], with i.i.d. $Y_{1,n}, \ldots, Y_{c,n}$, the function $X_{k+1,n}(z)$ can be expressed in terms of $X_{k,n}(z)$ by conditioning on the event that $X_{k,n}$ is equal to zero, or not. For the generalized FCTL queue more detailed information is needed, which is why we will use a lattice path counting approach to describe the queue length evolution throughout the cycles.

Let $\mathbf{1}_{\{A\}}$ be the indicator function for event A. Define the function

$$T(x_0, n_1, n_2, \ldots, n_k) = \mathbf{1}_{\{\min(x_0, x_0+n_1-1, x_0+n_1+n_2-2, \ldots, x_0+\sum_{i=1}^{k} n_i - k) \le 0\}},$$

for $k = 0, 1, 2, \ldots, g$. The parameters $x_0, n_1, \ldots, n_k$ are allowed to be random variables. We define $\mathscr{S}^{(k)} \subset \mathbb{N}^k$ as the set of k-dimensional integer-valued vectors $(l, n_1, n_2, \ldots, n_{k-1})$ for which $T(l, n_1, n_2, \ldots, n_{k-1}) = 1$. For $j = 0, 1, \ldots, k-1$, let $\mathscr{S}^{(k)}_j \subseteq \mathscr{S}^{(k)}$ denote the subset of $\mathscr{S}^{(k)}$ with elements $(l, n_1, n_2, \ldots, n_{k-1})$ that satisfy for $m = 0, 1, \ldots, j-1$,

$$l + \sum_{i=1}^{m} n_i - m > 0, \qquad \text{and}$$

$$l + \sum_{i=1}^{j} n_i - j = 0.$$

Notice that $\mathscr{S}^{(k)}$ contains all possible combinations $(X_0, Y_1, Y_2, \ldots, Y_{k-1})$ that will cause X_{k-1} to be zero. Out of all these combinations, $\mathscr{S}_j^{(k)}$ contains the combinations for which $X_j = 0$, while $X_0, \ldots, X_{j-1}$ all take strictly positive values. Adopting the terminology from [5], we say that $\mathscr{S}_j^{(k)}$ contains all elements for which the *effective green time* equals j, which means to say that the queue becomes empty after precisely j time slots. We will subdivide these subsets further by conditioning on their first element, which represents the queue length at the beginning of the cycle. Define $\mathscr{S}_{j,l}^{(k)} \subseteq \mathscr{S}_j^{(k)}$ as the elements of $\mathscr{S}_j^{(k)}$ that have first entry l. Note that $\mathscr{S}_{j,l}^{(k)}$ only needs to be defined for $l = 0, 1, \ldots, k-1$, for otherwise the queue cannot be empty at the end of time slot $k-1$. Finally, define

$$\mathscr{G}_{j,l} := \{(n_1, \ldots, n_j) \,|\, (l, n_1, n_2, \ldots, n_j, n_{j+1}, \ldots, n_k) \in \mathscr{S}_{j,l}^{(j+1)}\}$$

and notice that

$$(n_1, n_2, \ldots, n_j) \in \mathscr{G}_{j,l}$$

implies

$$(l, n_1, n_2, \ldots, n_j, n_{j+1}, \ldots, n_k) \in \mathscr{S}_{j,l}^{(k)},$$

for any $n_{j+1}, \ldots, n_k \in \mathbb{N}$ and $k = j+1, \ldots, g-1$.

Remark 3.1 A technical issue is that we allow j to be zero in the definition of $\mathscr{G}_{j,l}$. The only way to obtain an effective green time of length zero is when the queue length was zero at the cycle start. As a consequence, we have

$$\mathscr{G}_{0,0} = \{(\,)\}, \qquad \mathscr{G}_{0,l} = \emptyset \text{ for } l = 1, 2, \ldots,$$

where we have used $(\,)$ to denote a vector of dimension 0.

We can now express the queue lengths $X_{k,n}$ in terms of $X_{0,n}$. For $k = 1, \ldots, g$,

$$X_{k,n} = \begin{cases} 0 & \text{if } T(X_{0,n}, Y_{1,n}, Y_{2,n}, \ldots, Y_{k-1,n}) = 1, \\ X_{0,n} + \sum_{i=1}^{k} Y_{k,n} - k & \text{otherwise.} \end{cases} \tag{3.2}$$

and for $k = g+1, \ldots, c$,

$$X_{k,n} = \begin{cases} \sum_{i=g+1}^{k} Y_{k,n} & \text{if } T(X_{0,n}, Y_{1,n}, Y_{2,n}, \ldots, Y_{g-1,n}) = 1, \\ X_{0,n} + \sum_{i=1}^{k} Y_{k,n} - g & \text{otherwise.} \end{cases} \tag{3.3}$$

3.2 Generating function solution

From the recursion relation (3.2), and using the fact that X_0 is independent of the future arrivals, we find

$$X_k(z) = \sum_{l=0}^{\infty} \sum_{n_1=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} \left(1 - T(l, n_1, \ldots, n_{k-1})\right) z^{l+n_1+n_2+\cdots+n_k-k} \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_k = n_k)$$

$$+ \sum_{l=0}^{\infty} \sum_{n_1=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} T(l, n_1, \ldots, n_{k-1}) \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_k = n_k)$$

$$= \sum_{l=0}^{\infty} \sum_{n_1=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} z^{l+n_1+n_2+\cdots+n_k-k} \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_k = n_k)$$

$$+ \sum_{l=0}^{\infty} \sum_{n_1=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} T(l, n_1, \ldots, n_{k-1}) \left(1 - z^{l+n_1+n_2+\cdots+n_k-k}\right) \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_k = n_k)$$

Using the definitions of $\mathscr{S}_j^{(k)}$ and the independence between X_0 and $Y_1, \ldots, Y_k$,

$$X_k(z) = X_0(z) \frac{Y(\overbrace{z, \ldots, z}^{k}, 1, \ldots, 1)}{z^k}$$

$$+ \sum_{l=0}^{k-1} \sum_{j=0}^{k-1} \sum_{(l, n_1, \ldots, n_{k-1}) \in \mathscr{S}_j^{(k)}} \sum_{n_k=0}^{\infty} \left(1 - z^{l+n_1+n_2+\cdots+n_k-k}\right) \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_k = n_k).$$

Since $\mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_k = n_k) = q_l \mathbb{P}(Y_1 = n_1, \ldots, Y_k = n_k)$ we get

$$X_k(z) = X_0(z) \frac{Y(\overbrace{z, \ldots, z}^{k}, 1, \ldots, 1)}{z^k}$$

$$+ \sum_{l=0}^{k-1} q_l \sum_{j=0}^{k-1} \sum_{(n_1, \ldots, n_j) \in \mathscr{G}_{j,l}} \sum_{n_{j+1}=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} \left(1 - z^{l+n_1+n_2+\cdots+n_k-k}\right) \mathbb{P}(Y_1 = n_1, \ldots, Y_k = n_k).$$

Now, for an arbitrary element $(n_1, \ldots, n_j)$ in $\mathscr{G}_{j,l}$, it should hold that $l + n_1 + \cdots + n_j = j$. We can use this to write

$$\sum_{n_{j+1}=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} \left(1 - z^{l+n_1+n_2+\cdots+n_k-k}\right) \mathbb{P}(Y_1 = n_1, \ldots, Y_k = n_k)$$

$$= \mathbb{P}(Y_1 = n_1, \ldots, Y_j = n_j) - \sum_{n_{j+1}=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} z^{j+n_{j+1}+\cdots+n_k-k} \mathbb{P}(Y_1 = n_1, \ldots, Y_k = n_k)$$

$$=: f_k(1, n_1, \ldots, n_j) - f_k(z, n_1, \ldots, n_j),$$

where

$$f_k(z, n_1, \ldots, n_j) = \frac{\partial}{\partial y_1^{n_1} \partial y_2^{n_2} \ldots \partial y_j^{n_j}} \left. \frac{Y(y_1, y_2, \ldots, y_j, \overbrace{z, \ldots, z}^{k-j}, 1, \ldots, 1)}{n_1! n_2! \ldots n_j! z^{k-j}} \right|_{y_1 = \cdots = y_j = 0}. \tag{3.4}$$

8

To see the latter, note that by definition

$$f_k(z, n_1, \ldots, n_j) = \sum_{n_{j+1}=0}^{\infty} \sum_{n_{j+2}=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} z^{n_{j+1}+n_{j+2}+\cdots+n_k-(k-j)} \mathbb{P}(Y_1 = n_1, Y_2 = n_2, \ldots, Y_k = n_k),$$

and hence

$$f_k(1, n_1, \ldots, n_j) = \mathbb{P}(Y_1 = n_1, Y_2 = n_2, \ldots, Y_j = n_j).$$

We thus obtain the following lemma, where we have omitted the subscript n that refers to a particular cycle.

Lemma 3.2 *Let $\underline{n} = (n_1, \ldots, n_j)$ and $q_l = \mathbb{P}(X_0 = l)$ for $l = 0, 1, \ldots, g-1$. Then, for $k = 1, 2, \ldots, g$,*

$$X_k(z) = X_0(z)\frac{Y(\overbrace{z, \ldots, z}^{k}, 1, \ldots, 1)}{z^k} + \sum_{l=0}^{k-1} q_l \sum_{j=0}^{k-1} \sum_{\underline{n} \in \mathcal{G}_{j,l}} \left(f_k(1, n_1, \ldots, n_j) - f_k(z, n_1, \ldots, n_j)\right). \quad (3.5)$$

Having expressed $X_k(z)$ in terms of $X_0(z)$, for $k = 1, 2, \ldots, g$, we can use (3.3) to find $X_c(z)$. Following the same steps as for deriving Lemma 3.2, we get the lemma below.

Lemma 3.3 *For $k = g + 1, \ldots, c$,*

$$X_k(z) = X_0(z)\frac{Y(\overbrace{z, \ldots, z}^{k}, 1, \ldots, 1)}{z^g}$$

$$+ \sum_{l=0}^{g-1} q_l \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathcal{G}_{j,l}^{(g)}} \left(h_g(1, \overbrace{z, \ldots, z}^{k-g}, \overbrace{1, \ldots, 1}^{c-k}, n_1, \ldots, n_j) - h_g(z, \overbrace{z, \ldots, z}^{k-g}, \overbrace{1, \ldots, 1}^{c-k}, n_1, \ldots, n_j)\right)$$

with $h_m(z, y_{m+1}, \ldots, y_c, n_1, \ldots, n_j)$ defined as

$$\frac{\partial}{\partial y_1^{n_1} \partial y_2^{n_2} \ldots \partial y_j^{n_j}} \frac{Y(y_1, y_2, \ldots, y_j, \overbrace{z, \ldots, z}^{m-j}, y_{m+1}, \ldots, y_c)}{n_1! n_2! \ldots n_j! z^{m-j}}\Bigg|_{y_1 = \cdots = y_j = 0}. \quad (3.6)$$

If the system is in steady state, it should hold that $X_{c,n} \stackrel{d}{=} X_{0,n}$, so we can equate $X_c(z)$ and $X_0(z)$ and solve for $X_0(z)$ to arrive at the following result.

Theorem 3.4

$$X_0(z) = \frac{\sum_{l=0}^{g-1} q_l \zeta_l(z)}{z^g - Y(z, \ldots, z)}, \quad (3.7)$$

with

$$\zeta_l(z) = \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathcal{G}_{j,l}} z^g \left(h_g(1, z, \ldots, z, n_1, \ldots, n_j) - h_g(z, z, \ldots, z, n_1, \ldots, n_j)\right). \quad (3.8)$$

9

Remark 3.5 To help interpret (3.7) write

$$z^g h_g(1, z, \ldots, z, n_1, \ldots, n_j) = \mathbb{E}\left[z^{g+Y_{g+1}+\cdots+Y_c} \mathbf{1}_{\{Y_1=n_1,\ldots,Y_j=n_j\}} \right], \tag{3.9}$$

$$z^g h_g(z, z, \ldots, z, n_1, \ldots, n_j) = \mathbb{E}\left[z^{j+Y_{j+1}+\cdots+Y_c} \mathbf{1}_{\{Y_1=n_1,\ldots,Y_j=n_j\}} \right], \tag{3.10}$$

and

$$
\begin{aligned}
X_0(z) &= \frac{Y(z,\ldots,z)}{z^g} X_0(z) + \frac{\sum_{l=0}^{g-1} q_l \zeta_l(z)}{z^g} \\
&= \frac{Y(z,\ldots,z)}{z^g} X_0(z) + \sum_{l=0}^{g-1} q_l \mathbb{E}\left[\left(z^{g+Y_{g+1}+\cdots+Y_c} - z^{j+Y_{j+1}+\cdots+Y_c} \right) \mathbf{1}_{\{Y_1=n_1,\ldots,Y_j=n_j\}} \right].
\end{aligned} \tag{3.11}
$$

The first expression on the right hand side of (3.11) represents the relation $X_{0,n+1} = X_{0,n} + \sum_{i=1}^{c} Y_{i,n} - g$ in terms of generating functions. The last term in (3.11) contains all "correction terms" for those summands where the queue becomes empty at time slot j, because then $X_{0,n+1} = X_{0,n} + \sum_{i=1}^{j} Y_{i,n} + \sum_{i=g+1}^{c} Y_{i,n} - j$, which can be written as

$$X_{0,n+1} = X_{0,n} + \sum_{i=1}^{c} Y_{i,n} - g + \left(g + \sum_{i=g+1}^{c} Y_{i,n} - j - \sum_{i=j+1}^{c} Y_{i,n} \right),$$

the last part being the correction term. The function $\zeta_l(z)$ is composed of all correction terms that correspond to $X_{0,n} = l$. Each correction term consists of the term (3.10), representing the part that should be removed, and its replacement term (3.9).

Remark 3.6 If the Y_i are independent (but not necessarily identically distributed) random variables, Equation (3.7) can be simplified. In this case, $Y(y_1, \ldots, y_c) = \prod_{i=1}^{c} Y_i(z)$, which can be substituted in (3.6) to find

$$h_g(z, z, \ldots, z, n_1, \ldots, n_j) = z^{j-g} \, \mathbb{P}(Y_1 = n_1, \ldots, Y_j = n_j) \prod_{i=j+1}^{c} Y_i(z), \tag{3.12}$$

$$h_g(1, z, \ldots, z, n_1, \ldots, n_j) = \mathbb{P}(Y_1 = n_1, \ldots, Y_j = n_j) \prod_{i=g+1}^{c} Y_i(z). \tag{3.13}$$

Let the random variable G denote the effective green time, as defined earlier, and notice that

$$\mathbb{P}(G = j) = \sum_{l=0}^{g-1} q_l \sum_{\underline{n} \in \mathcal{G}_{j,l}} \mathbb{P}(Y_1 = n_1, \ldots, Y_j = n_j). \tag{3.14}$$

From (3.7) we then obtain

$$
\begin{aligned}
X_0(z) &= \frac{z^g \sum_{j=0}^{g-1} \mathbb{P}(G=j) \left(\prod_{i=g+1}^{c} Y_i(z) - z^{j-g} \prod_{i=j+1}^{c} Y_i(z) \right)}{z^g - \prod_{i=1}^{c} Y_i(z)} \\
&= \frac{(1 - \mathbb{P}(G=g)) z^g \prod_{i=g+1}^{c} Y_i(z) - \sum_{j=0}^{g-1} \mathbb{P}(G=j) z^j \prod_{i=j+1}^{c} Y_i(z)}{z^g - \prod_{i=1}^{c} Y_i(z)}.
\end{aligned} \tag{3.15}
$$

Using that $\mathbb{P}(G = 0) = \mathbb{P}(X_0 = 0)$, $\mathbb{P}(G = g) = 1 - \mathbb{P}(X_{g-1} = 0)$, and $\mathbb{P}(G = j) = \mathbb{P}(X_j = 0) - \mathbb{P}(X_{j-1} = 0)$, for $j = 1, \ldots, g-1$, (3.15) can be written as

$$X_0(z) = \frac{\sum_{j=1}^{g} \left(1 - Y_j(z)/z\right) \mathbb{P}(X_{j-1} = 0) z^j \prod_{i=j+1}^{c} Y_i(z)}{z^g - \prod_{i=1}^{c} Y_i(z)},$$

where the unknown probabilities $\mathbb{P}(X_i = 0)$ are still to be determined.

Remark 3.7 When Y_i are i.i.d. random variables, with pgf $Y(z)$, this expression simplifies further to the classical result [8, 23]

$$X_0(z) = \frac{(1 - Y(z)/z) \sum_{j=1}^{g} \mathbb{P}(X_{j-1} = 0) z^j Y(z)^{c-j}}{z^g - Y(z)^c}.$$

Let us now proceed with the expression in (3.7) and show how this generating function can be converted into algorithms for the performance analysis of the generalized FCTL queue. First note that there are still g unknowns $q_0, \ldots, q_{g-1}$ in (3.7), which can be found using a classical reasoning. With Rouché's theorem [2], it can be shown that the denominator of (3.7) has g zeros on or within the unit circle $|z| \leq 1$. Since a pgf is analytic and well-defined in $|z| \leq 1$, the numerator of $X_0(z)$ should vanish at each of the zeros. This gives g equations. One of the zeros equals 1, and leads to a trivial equation. However, the normalization condition $X_0(1) = 1$ provides an additional equation. Using l'Hôpital's rule, this condition is found to be

$$\sum_{l=0}^{g-1} q_l \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} \left(g - j - \sum_{i=j+1}^{g} \mathbb{E}[Y_i \,|\, A_j] \right) \mathbb{P}(A_j) = g - \sum_{i=1}^{c} \mathbb{E}[Y_i], \qquad (3.16)$$

where we have defined A_j as the event $\{Y_1 = n_1, \ldots, Y_j = n_j\}$ for compactness. From this definition it follows that $\mathbb{E}[Y_i \,|\, A_0] = \mathbb{E}[Y_i]$ and $\mathbb{P}(A_0) = 1$. We can write (3.16) as

$$\sum_{l=0}^{g-1} q_l b_l = \eta.$$

Note that $b_l = \zeta_l'(1)$, with $\zeta_l(z)$ as defined in (3.7). Denote the g roots of $z^g = Y(z, \ldots, z)$ on and within the unit circle by $z_0 = 1, z_1, \ldots, z_{g-1}$. The g unknowns $q_0, \ldots, q_{g-1}$ then follow from solving the set of linear equations

$$\begin{pmatrix} b_0 & b_1 & b_2 & \ldots & b_{g-1} \\ \zeta_0(z_1) & \zeta_1(z_1) & \zeta_2(z_1) & \ldots & \zeta_{g-1}(z_1) \\ \zeta_0(z_2) & \zeta_1(z_2) & \zeta_2(z_2) & \ldots & \zeta_{g-1}(z_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \zeta_0(z_{g-1}) & \zeta_1(z_{g-1}) & \zeta_2(z_{g-1}) & \ldots & \zeta_{g-1}(z_{g-1}) \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ \vdots \\ q_{g-1} \end{pmatrix} = \begin{pmatrix} \eta \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \qquad (3.17)$$

To make this work, we need an efficient numerical algorithm for finding the roots $z_1, \ldots, z_{g-1}$. Many root-finding algorithms exist, for instance based on successive substitution [12] or the Lagrange inversion theorem [13]. In the appendix we present a new scheme (Algorithm 5) based on the truncation of infinite series in combination with a root-finding procedure for polynomial functions.

We also need to deal with the sets $\mathscr{S}_{j,l}^{(k)}$ that may have an infinite number of elements. For this, notice that $\mathscr{G}_{j,l}$ has a finite cardinality. Finding all elements in $\mathscr{G}_{j,l}$ is rather straightforward, because they can be found by a simple enumeration. When $l = 0$, the effective green period is zero, meaning that j must be zero as well. As a consequence, all $\mathscr{G}_{j,0}$ are empty sets, except for $\mathscr{G}_{0,0}$, which is a set containing one element: the empty vector ( ). For $0 < l < k$, the minimum effective green time is at least l. Fixing l between 1 and $k-1$, and j between l and $k-1$, we note that all elements in $\mathscr{G}_{j,l}$ must satisfy the following conditions:

$$l + n_1 - 1 > 0,$$
$$l + n_1 + n_2 - 2 > 0,$$
$$\vdots$$
$$l + n_1 + \cdots + n_{j-1} - (j-1) > 0,$$
$$l + n_1 + \cdots + n_j - j = 0.$$

Since $n_1, \ldots, n_j$ are nonnegative integers, these conditions lead to an easy enumeration of all elements $\underline{n}$ in $\mathscr{G}_{j,l}$. See Algorithm 6 in the appendix for efficiently calculating all nonempty sets $\mathscr{G}_{j,l}$.

Taken everything together, we can evaluate the generating function $X_0(z)$ as described in Algorithm 1.

Algorithm 1 Computing $X_0(z)$

1: Input g, c, and $Y(z_1, z_2, \ldots, z_c)$
2: Set $\mathbb{E}[Y] = \frac{\mathrm{d}}{\mathrm{d}z} Y(z, z, \ldots, z)\big|_{z=1}$
3: Generate all nonempty sets $\mathscr{G}_{j,l}$ using Algorithm 6
4: Determine all roots $z_1, z_2, \ldots, z_{g-1}$ inside the unit circle using Algorithm 5
5: **for** $l = 0$ to $g - 1$ **do**
6:     Set $\zeta_l(z) = 0$
7:     **for** $j = l$ to $g - 1$ **do**
8:         **for all** $\underline{n} \in \mathscr{G}_{j,l}$ **do**
9:             Set $(n_1, n_2, \ldots, n_j) = \underline{n}$
10:             Compute $h_g(z, y_{g+1}, \ldots, y_c, n_1, \ldots, n_j)$ using Equation (3.6)
11:             Set $\zeta_l(z) = \zeta_l(z) + z^g \big(h_g(1, z, \ldots, z, n_1, \ldots, n_j) - h_g(z, z, \ldots, z, n_1, \ldots, n_j) \big)$
12:         **end for**
13:     **end for**
14:     Set $b_l = \zeta_l'(1)$
15: **end for**
16: Set $\eta = g - \mathbb{E}[Y]$
17: Find $q_1, q_2, \ldots, q_{g-1}$ by solving system (3.17)
18: Compute $X_0(z)$ using Equation (3.7)
19: **return** $X_0(z)$

3.3 Queue length distribution

Our ultimate goal is to obtain algorithmic solutions to compute the expectation and tail probabilities of the queue length during a cycle. In the previous sub-section we determined $X_1(z), X_2(z), \ldots, X_c(z)$,

the pgfs of the queue length at the end of each time slot. These probability generating functions are useful tools to make the analysis tractable, but for practical purposes it is essential that we determine the actual queue length probabilities, in particular in case of spillback. In this sub-section we present an efficient algorithm to numerically invert the pgfs and obtain the queue length distribution at various epochs. For the *mean* queue length it is possible to obtain explicit expressions, by differentiating the pgfs and substituting $z = 1$, which we show at the end of this section.

Now that we can calculate $X_0(z)$, we proceed to use Algorithm 1 to obtain more information about the queue length. Notice that with Lemmas 3.2 and 3.3 and Algorithm 1 we can evaluate all $X_k(z)$ for $k = 1, \ldots, c$. While these are all pgfs at the end of specific time slots, the pgf of the queue length at an arbitrary time slot $\bar{X}(z)$ follows from

$$\bar{X}(z) = \frac{1}{c} \sum_{i=1}^{c} X_i(z). \tag{3.18}$$

The queue length distribution can then be calculated using a standard inverse theorem such as Algorithm 2. We have used Algorithm 2 with (3.18) to generate the results in Table 1.

Algorithm 2 Determine $P(X = k)$ by inverting $X(z)$ [1, 6]

1: Input $X(z)$ and k
2: $l = 1$
3: $\gamma = 10$
4: **if** $k = 0$ **then**
5:     $p_k = X(10^{-\gamma})$
6: **else**
7:     $r = 10^{-\gamma/(2lk)}$
8:     $p_k = \frac{1}{2lkr^k} \left(X(r) + (-1)^k X(-r) + 2 \sum_{j=1}^{lk-1} \text{Re}\left[X\left(r \exp\left(\frac{\pi i j}{lk}\right) \exp\left(-\frac{\pi i j}{l}\right) \right) \right] \right)$
9: **end if**
10: **return** p_k

Finally, we present some compact expressions for the mean queue length at the end of each time slot. We used these expressions to generate all results in Section 4.2. Denote by $N(z)$ and $D(z)$ the numerator and denominator of $X_0(z)$ as given in (3.7). By differentiation and L'Hôpital's rule, it can be shown that

$$\mathbb{E}[X_0] = \frac{N''(1) - D''(1)}{2D'(1)},$$

where we have used that $X_0(1) = 1$ and hence $N'(1) = D'(1)$. Define

$$Y^{(j)} := \sum_{i=j+1}^{g} Y_i, \qquad Y^{(r)} = \sum_{i=g+1}^{c} Y_i, \qquad Y = \sum_{i=1}^{c} Y_i.$$

It is readily checked that

$$D'(1) = g - \mathbb{E}[Y], \qquad D''(1) = g(g-1) - \mathbb{E}[Y(Y-1)].$$

13

For the numerator, we find

$$N'(1) = \sum_{l=0}^{g-1} q_l \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} \left(\mathbb{E}\left[(g + Y^{(r)}) \mathbf{1}_{\{A_j\}} \right] - \mathbb{E}\left[(j + Y^{(j)} + Y^{(r)}) \mathbf{1}_{\{A_j\}} \right] \right)$$

$$= \sum_{l=0}^{g-1} q_l \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} \left(g - j - \mathbb{E}[Y^{(j)} | A_j] \right) \mathbb{P}(A_j),$$

where as before $A_j = \{Y_1 = n_1, \ldots, Y_j = n_j\}$. Taking the second derivative yields

$$N''(1) = \sum_{l=0}^{g-1} q_l \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} \left(\mathbb{E}\left[(g + Y^{(r)})^2 \mathbf{1}_{\{A_j\}} \right] - \mathbb{E}\left[(j + Y^{(j)} + Y^{(r)})^2 \mathbf{1}_{\{A_j\}} \right] \right) - N'(1).$$

Using $D''(1) = g^2 - \mathbb{E}[Y^2] - D'(1)$ and $N'(1) = D'(1)$ gives the following result:

Lemma 3.8

$$\mathbb{E}[X_0] = \frac{\mathbb{E}[Y^2] - g^2 + \sum q_l \mathbb{P}(A_j)(g^2 - j^2)}{2(g - \mathbb{E}[Y])}$$

$$- \sum_{l=0}^{g-1} \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} \frac{q_l}{2(g - \mathbb{E}[Y])} \mathbb{P}(A_j) \mathbb{E}\left[2(g-j)Y^{(r)} - 2(j - Y^{(r)})Y^{(j)} - \left(Y^{(j)}\right)^2 \Big| A_j \right]. \quad (3.19)$$

Observe that (3.19) depends on the q_l's, $\mathbb{E}[Y_i]$ for $i = 1, \ldots, c$, and $\mathbb{E}[Y_i Y_k | A_j]$ for $i = 1, \ldots, g$ and $k = 1, \ldots, c$. The mean queue lengths $\mathbb{E}[X_k]$ for $k = 1, 2, \ldots, c$ can be found by expressing $\mathbb{E}[X_k]$ in terms of $\mathbb{E}[X_{k-1}]$ or $\mathbb{E}[X_{k+1}]$. For queue lengths during the red period, we see that

$$\mathbb{E}[X_k] = \mathbb{E}[X_{k+1}] - \mathbb{E}[Y_{k+1}], \qquad k = c-1, c-2, \ldots, g,$$

which means that these mean queue lengths can be expressed in $\mathbb{E}[X_c]$ (and hence $\mathbb{E}[X_0]$) by successive substitution.

During the green period, the expressions become slightly more involved, but it is possible to express $\mathbb{E}[X_k]$ in terms of $\mathbb{E}[X_{k-1}]$ as

$$\mathbb{E}[X_k] = \mathbb{E}[X_{k-1}] + \mathbb{E}[Y_k] - 1 - \sum_{l=0}^{k-1} q_l \sum_{j=0}^{k-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} \mathbb{P}(A_j) \left(\mathbb{E}[Y_{k+1} | A_j] - 1 \right), \quad \text{for } k = 1, \ldots, g-1.$$

4 Network settings

With the algorithms developed in Section 3 to analyze the output of an intersection with correlated input, we now extend the scope in order to deal with a network of intersections. In Subsection 4.1 we describe the decomposition approach, a heuristic method to combine multiple isolated intersections into a network model, and in Subsection 4.2 we demonstrate the algorithm for two network scenarios.

4.1 Decomposition approach

Let us first quantify the output process of the generalized FCTL queue in Section 3. Define $O_{i,n}$ as the output in time slot i ($i = 1, \ldots, g$) in cycle n, so that

$$O_{i,n} = \begin{cases} 1 & \text{if } X_{i-1,n} > 0, \\ Y_{i,n} & \text{if } X_{i-1,n} = 0. \end{cases} \tag{4.1}$$

Let $O_n(z_1, \ldots, z_g)$ denote the pgf of the joint output $(O_{1,n}, \ldots, O_{g,n})$ and G_n denote the effective green time in cycle n. Note that the joint output vector $(O_{1,n}, \ldots, O_{g,n})$, given that $G_n = j$, equals

$$(\overbrace{1, 1, \ldots, 1}^{j}, Y_{j+1,n}, \ldots, Y_{g,n}), \qquad j = 1, 2, \ldots, g.$$

We will again omit the subscript n denoting the cycle number. If $(n_1, \ldots, n_j)$ is an arbitrary vector in $\mathscr{G}_{j,l}$, it holds that $l + n_1 + \cdots + n_j = j$. The pgf of the joint output during this cycle follows from summing over all possible l, j and $(n_1, \ldots, n_j) \in \mathscr{G}_{j,l}$:

$$O(z_1, \ldots, z_g) = \sum_{l=0}^{\infty} \sum_{n_1=0}^{\infty} \cdots \sum_{n_g=0}^{\infty} \left(1 - T(l, n_1, \ldots, n_{g-1})\right) \left(\prod_{k=1}^{g} z_k\right) \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_g = n_g)$$

$$+ \sum_{l=0}^{g-1} \sum_{j=0}^{g-1} \sum_{(n_1, \ldots, n_j) \in \mathscr{G}_{j,l}} \sum_{n_{j+1}=0}^{\infty} \cdots \sum_{n_g=0}^{\infty} \left(\prod_{k=1}^{j} z_k\right) \left(\prod_{k=j+1}^{g} z_k^{n_k}\right) \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_g = n_g). \tag{4.2}$$

This then gives

$$O(z_1, \ldots, z_g) =$$

$$= \left(1 - \sum_{l=0}^{g-1} \sum_{j=0}^{g-1} \sum_{(n_1, \ldots, n_j) \in \mathscr{G}_{j,l}} \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_j = n_j)\right) \prod_{k=1}^{g} z_k$$

$$+ \sum_{l=0}^{g-1} \sum_{j=0}^{g-1} \sum_{(n_1, \ldots, n_j) \in \mathscr{G}_{j,l}} \sum_{n_{j+1}=0}^{\infty} \cdots \sum_{n_g=0}^{\infty} \left(\prod_{k=1}^{j} z_k\right) \left(\prod_{k=j+1}^{g} z_k^{n_k}\right) \mathbb{P}(X_0 = l, Y_1 = n_1, \ldots, Y_g = n_g)$$

$$= \left(1 - \sum_{l=0}^{g-1} \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} q_l \mathbb{P}(A_j)\right) \prod_{k=1}^{g} z_k$$

$$+ \sum_{l=0}^{g-1} q_l \sum_{j=0}^{g-1} \sum_{\underline{n} \in \mathscr{G}_{j,l}} \left(\prod_{k=1}^{j} z_k\right) h_j(1, z_{j+1}, \ldots, z_g, \overbrace{1, \ldots, 1}^{c-g}, n_1, \ldots, n_j) \tag{4.3}$$

with h_j as defined in (3.6). Note that $h_0(1, z_1, \ldots, z_g, 1, \ldots, 1) = Y(z_1, \ldots, z_g, 1, \ldots, 1)$ and $\mathbb{P}(A_j) = h_j(1, 1, \ldots, 1, n_1, \ldots, n_j)$. The complete program to compute $O(z_1, z_2, \ldots, z_g)$ is then given in Algorithm 3.

Remark 4.1 It is evident that correlation in input carries over to correlation in output. Less obvious is that even with independent input, the output processes in consecutive time slots are

Algorithm 3 Computing $O(z_1, z_2, \ldots, z_g)$

1: Input g, c, and $Y(z_1, z_2, \ldots, z_c)$
2: Find $q_1, q_2, \ldots, q_{g-1}$ using Algorithm 1
3: Set $S = 0, T = 0$
4: **for** $l = 0$ to $g - 1$ **do**
5:     **for** $j = l$ to $g - 1$ **do**
6:         **for all** $\underline{n} \in \mathscr{G}_{j,l}$ **do**
7:             Set $(n_1, n_2, \ldots, n_j) = \underline{n}$
8:             Compute $h_j(z, y_{j+1}, \ldots, y_c, n_1, \ldots, n_j)$ using Equation (3.6)
9:             Set $S = S + q_l h_j(1, 1, \ldots, 1, n_1, \ldots, n_j)$
10:            Set $T = T + q_l \left(\prod_{k=1}^{j} z_k \right) h_j(1, z_{j+1}, \ldots, z_g, 1, \ldots, 1, n_1, \ldots, n_j)$,
11:         **end for**
12:     **end for**
13: **end for**
14: Set $O(z_1, z_2, \ldots, z_g) = (1 - S) \prod_{k=1}^{g} z_k + T$
15: **return** $O(z_1, z_2, \ldots, z_g)$

---

correlated. In the case of independent $Y_1, \ldots, Y_c$, the joint output pgf reduces to

$$O(z_1, z_2, \ldots, z_g) = \mathbb{P}(G = 0) \prod_{i=1}^{g} Y_i(z_i) + \sum_{j=1}^{g-1} \mathbb{P}(G = j) \prod_{i=1}^{j} z_i \prod_{i=j+1}^{g} Y_i(z_i)$$

$$+ \mathbb{P}(G = g) \prod_{i=1}^{g} z_i. \tag{4.4}$$

The correlation follows from the fact that this is not the product of the marginal pgfs $O_i(z_i)$ for $i = 1, \ldots, g$.

To analyze a network of fixed-cycle intersections, we decompose the network into isolated generalized FCTL queues. We exclude situations where the output of a queue could become the input of that same queue through some cyclic path in the network, meaning that the network can be represented as a directed acyclic graph where nodes represent queues. Nodes without parents can be modeled as standard FCTL queues with independent external arrivals. Figure 3 shows an example of three intersections with one main traffic flow, with queues labeled $Q^{(1,0)}, Q^{(2,0)}$, and $Q^{(3,0)}$, and three minor flows $Q^{(1,1)}, Q^{(2,1)}$, and $Q^{(3,1)}$ representing side traffic.
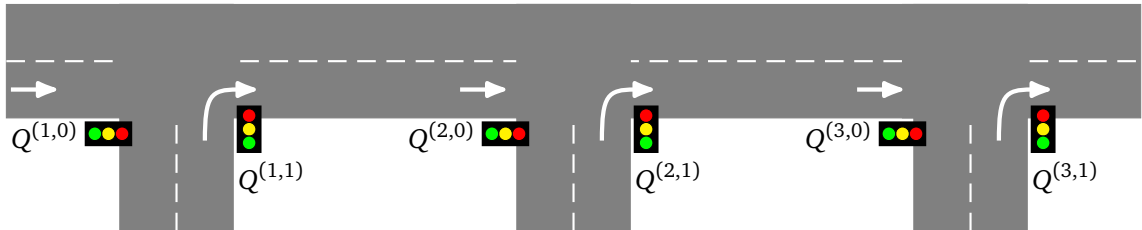

Figure 3: The network considered in this section.

This network can be decomposed into six queues. Queues $Q^{(1,0)}, Q^{(1,1)}, Q^{(2,1)}$, and $Q^{(3,1)}$ are standard FCTL queues with Poisson input. The analysis in [23] can be used to obtain queue length distributions and Equation (4.4) gives the joint output of each of these queues. Following the main flow downstream, we first determine the input of $Q^{(2,0)}$, which is the superposition of the delayed output processes of $Q^{(1,0)}$ and $Q^{(1,1)}$. The analysis in Section 3 gives the queue length pgf, and the results from this section give the joint output. Algorithm 4 provides a detailed description of this method, which can be used to find approximations for all queue length pgfs in the network.

Algorithm 4 Network analysis

1: Input the number of intersections n
2: Input $g^{(1,0)}$, c, and $Y^{(1,0)}(z_1, z_2, \ldots, z_c)$ for $Q^{(1,0)}$
3: Compute $X_0^{(1,0)}(z)$ using Algorithm 1
4: Compute $O^{(1,0)}(z_1, z_2, \ldots, z_{g^{(1,0)}})$ using Algorithm 3
5: **for** $i = 2$ to n **do**
6:     Input $g^{(i-1,1)}$ and $Y^{(i-1,1)}(z_1, z_2, \ldots, z_c)$ for $Q^{(i-1,1)}$
7:     Input δ, the number of red time slots preceding the first green time slot at $Q^{(i-1,1)}$
8:     Compute $O^{(i-1,1)}(z_1, z_2, \ldots, z_{g^{(i-1,1)}})$ using Algorithm 3
9:     Set $Y^{(i,0)}(z_1, z_2, \ldots, z_c) = O^{(i-1,0)}(z_{\mathrm{mod}(d,c)+1}, z_{\mathrm{mod}(1+d,c)+1}, \ldots, z_{\mathrm{mod}(g^{(i-1,0)}-1+d,c)+1})$
10:        $\times O^{(i-1,1)}(z_{\mathrm{mod}(\delta+d,c)+1}, z_{\mathrm{mod}(1+\delta+d,c)+1}, \ldots, z_{\mathrm{mod}(g^{(i-1,1)}-1+\delta+d,c)+1})$
11:     Compute $X_0^{(i,0)}(z)$ using Algorithm 1
12:     Compute $O^{(i,0)}(z_1, z_2, \ldots, z_{g^{(i,0)}})$ using Algorithm 3
13: **end for**

4.2 Two example networks

Consider an idealized network of ten connected generalized FCTL queues, $Q^{(1)}, \ldots, Q^{(10)}$, each with a fixed cycle of $g = 10$ green and $r = 10$ red time slots and a fixed travel time between two consecutive intersections of d time slots. The setting is similar to that in Figure 3 without the side traffic. Since all intersections have the same fixed-cycle plan of 10 green and 10 red time slots, the case $d = 0$ corresponds to a perfect green wave where vehicles arrive exactly at the moment that the signal turns green, and no queues will build up. In the case $d = 5$, the vehicles departing during the first five time slots of $Q^{(i)}$, will arrive at $Q^{(i+1)}$ during time slots 6–10. These vehicles will pass without delay, except when there is a queue of more than five vehicles waiting at the beginning of the cycle. The vehicles departing in time slots 6–10 from $Q^{(i)}$, will arrive at $Q^{(i+1)}$ during time slots 11–15, which means that they arrive during the red period and will be delayed until (at least) the next green period.

Only $Q^{(1)}$ has external arrivals, assumed to arrive according to a Poisson process with rate $\lambda = 0.45$ per time slot. The *occupation rate* $\rho := \lambda c / g = 0.9$ is close to one, indicating that the intersections are operating close to their maximum capacity. Appendix B provides more details about the input settings of this example, and how to determine the arrival processes at the other intersections.

To assess the accuracy of our decomposition approach, we have written a microscopic discrete-event simulation of the complete network of generalized FCTL queues for comparison. Observe that in Figures 4(a) and 4(b) the simulated values (the black dots) are indistinguishable from those computed using our generalized FCTL analysis, based on the decomposition approach (the

black solid lines). Clearly, $d = 5$ results in a better traffic flow, with smaller mean queue lengths, than $d = 10$. Noting that even the classical FCTL analysis (the gray dashed lines) approximates the mean queue lengths very well, we conclude that the arrival pattern does not have an influential impact on the queue lengths in this simplified example. In the remainder of the section we discuss a more realistic example where this impact is much bigger.
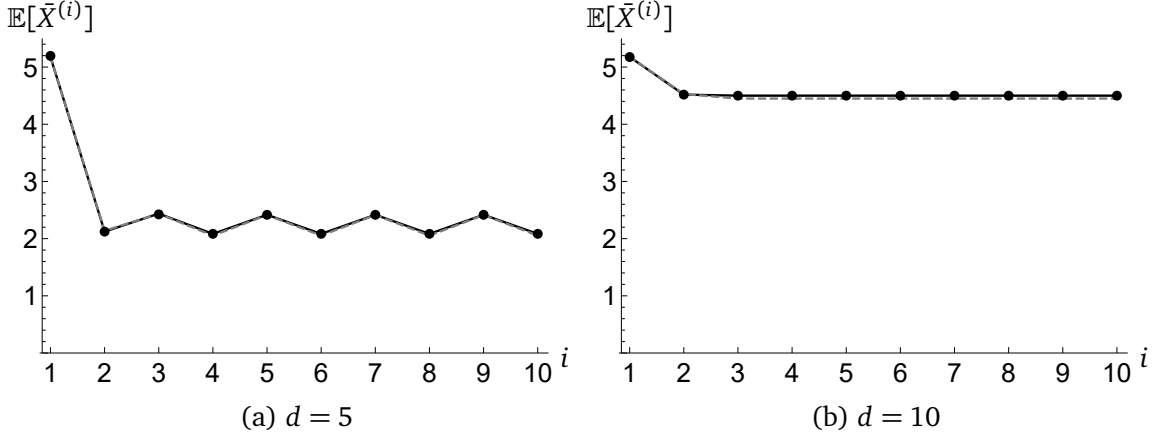

(a) $d = 5$ (b) $d = 10$

Figure 4: Overall mean queue lengths in the network without side traffic in Section 4.2. The dots represent the network simulation, the black solid and gray dashed lines are obtained via the decomposition method and the classic FCTL analysis, respectively. Figures 4(a) and 4(b) depict the expected overall mean queue length at intersection i ($i = 1, 2, \ldots, 10$), $\mathbb{E}[\bar{X}^{(i)}] := \sum_{k=1}^{c} \mathbb{E}[X_k^{(i)}]/c$ for the cases $d = 5$ and $d = 10$ respectively.

We now extend the previous example by merging traffic in the main flow with side traffic from minor roads, travelling towards the same destination as illustrated in Figure 3. Each intersection has two flows, the main flow and the minor flow with side traffic. Note that the arrival patterns of queues in the main flows are similar to those in Section 2.2, with one large batch of vehicles arriving from the upstream main flow, and one smaller batch from the upstream minor flow. Each of the batches may be followed by vehicles arriving in the free flow. The specific input settings in this example can be found in Appendix B.

We have implemented this network model using Algorithm 4, with the purpose of comparing the mean queue lengths of the main flows at each of the 10 intersections, but now for the cases $d = 0$ and $d = 5$. Due to the inflow of side traffic, the case $d = 0$ will no longer result in empty queues. The large batch of (maximally) ten vehicles will experience no delay *only if* no inflow from side traffic took place during the previous cycle. With $d = 5$, the small batch of (maximally) three vehicles will arrive exactly at the moment that the traffic signal of the main flow turns green. As a consequence, these settings are favorable for the small batch, but also the first five vehicles from the large batch might benefit. The mean queue lengths in Figures 5(a)–(d) and in Table 2 indicate that for intersections 1–7, the generalized FCTL queue gives extremely accurate approximations for the original network model, while for intersections 8–10, when the occupation rate ρ exceeds 0.75, the algorithmic method is less accurate but still reasonable.

By numeric inversion of the probability generating functions, we also obtain the queue length distributions at the end of each time slot. This gives more insight in how the number of vehicles fluctuates throughout a cycle. Of particular interest is the queue length distribution at the start
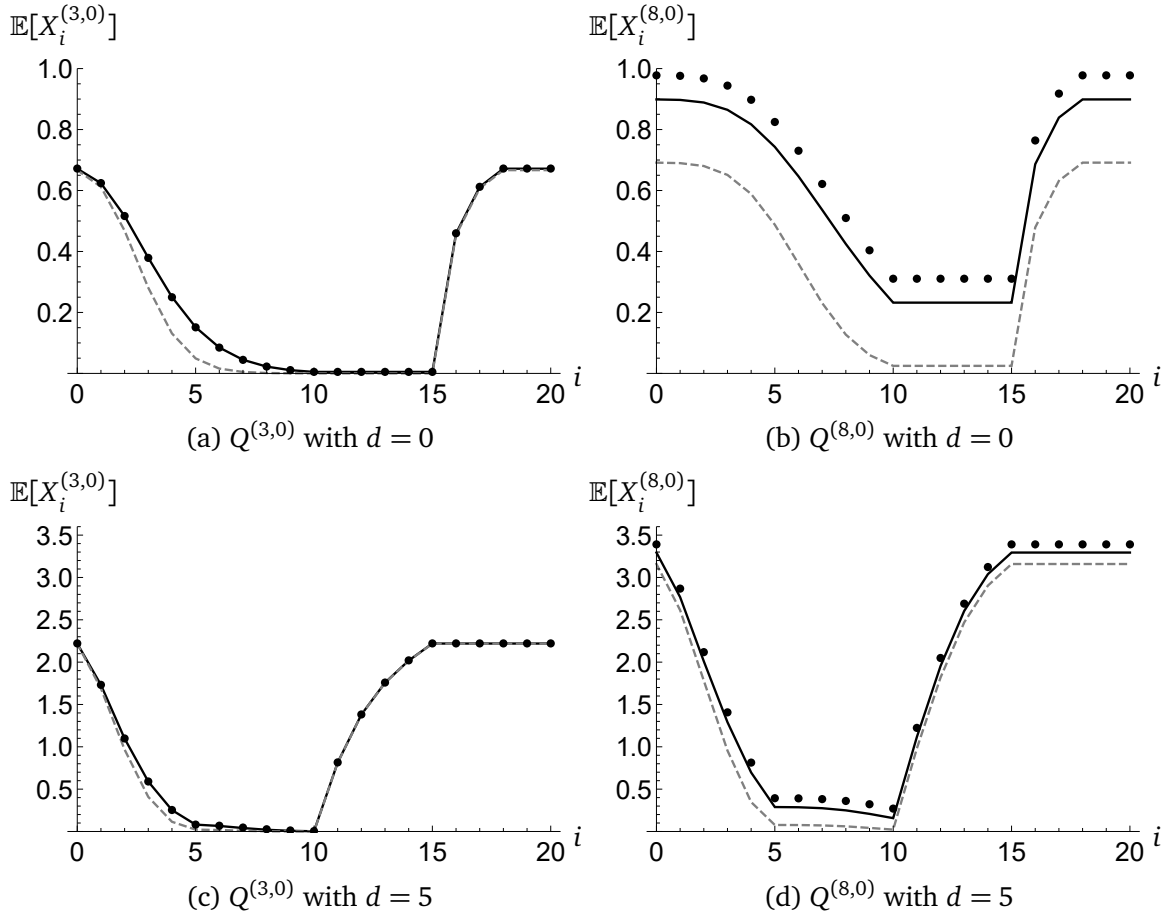
Figure 5: Mean queue lengths at time slots $i = 0, 1, \ldots, 20$ for intersections 3 and 8 in the network with side traffic in Section 4.2. The dots represent the network simulation, the black solid and gray dashed lines are obtained via the decomposition method and the classic FCTL analysis, respectively.

of a cycle, just before the queue discharges, when the expected queue length is at its peak. Just like before, our approach provides excellent estimates for the queue length probabilities for intersections 1–7. For intersections 8–10 the differences between the simulated and approximated values become larger, but the general shape of the probability mass function (pmf) is preserved. Consider, as an example, the case $d = 5$. In Figure 6 we have plotted the pmf of the queue length at the start of a cycle for intersections 8 and 10. In these plots we can find the probability that the number of vehicles exceeds a specific threshold. For example, for intersection 10 (with $\rho = 0.9$) the simulation yields $\mathbb{P}(X_0^{(10,0)} > 5) = 0.37$, which should be interpreted as the fraction of cycles with more than five vehicles waiting when the signal turns green. The value obtained by our algorithm is $\mathbb{P}(X_0^{(10,0)} > 5) = 0.24$, while the classic FCTL method gives an estimate of 0.07.

When the load increases, the queues become longer and the oscillations become higher. In reality, once a queue has grown beyond a certain threshold, it may take a long time for this queue to completely disappear. This results in multiple consecutive cycles of large platoons of departing vehicles, which form a highly correlated arrival process at the next intersection. In our decomposed

19

Mean queue length $\bar{X}^{(i,0)}$ for $d = 0$										
Intersection	1	2	3	4	5	6	7	8	9	10
Load ρ	0.3	0.37	0.43	0.5	0.57	0.63	0.7	0.77	0.83	0.9
Simulation	0.493	0.232	0.260	0.293	0.335	0.395	0.491	0.665	1.046	2.192
Decomposition	0.493	0.231	0.260	0.292	0.333	0.386	0.464	0.588	0.810	1.323

Mean queue length $\bar{X}^{(i,0)}$ for $d = 5$										
Intersection	1	2	3	4	5	6	7	8	9	10
Load ρ	0.3	0.37	0.43	0.5	0.57	0.63	0.7	0.77	0.83	0.9
Simulation	0.493	0.359	1.161	0.821	1.549	1.287	1.984	1.939	2.814	3.829
Decomposition	0.493	0.359	1.159	0.819	1.534	1.273	1.920	1.835	2.478	2.858

Table 2: The mean queue lengths at each of the ten intersections, for the network with side traffic of Section 4.2.

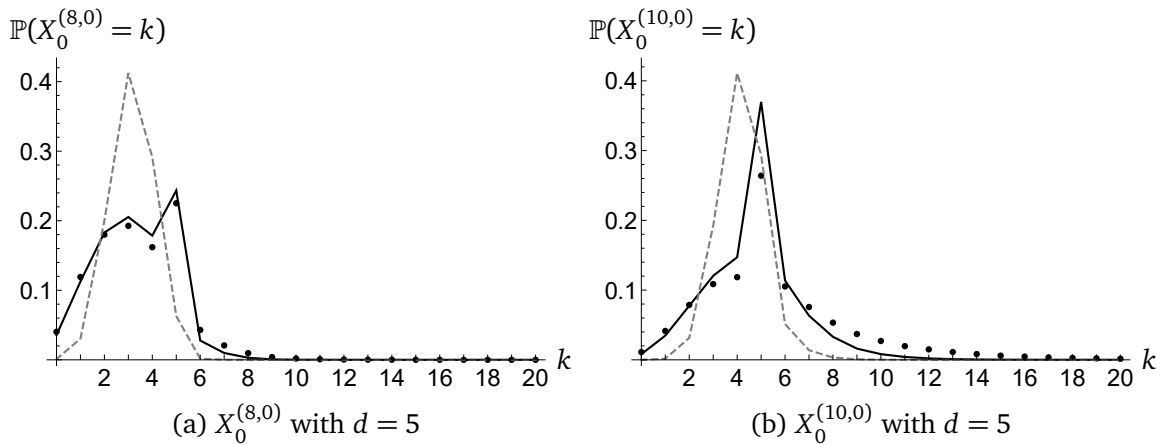(a) $X_0^{(8,0)}$ with $d = 5$ (b) $X_0^{(10,0)}$ with $d = 5$

Figure 6: Queue length distributions at the end of a red period for intersections 8 and 10 in the network with side traffic in Section 4.2. The dots represent the network simulation, the black solid and gray dashed lines are obtained via the decomposition method and the classic FCTL analysis, respectively.

model we can capture the high intensity of this arrival process, but not the correlation between successive batches. This phenomenon causes the deterioration in performance in situations close to saturation. Compared to the classical FCTL model with independent arrivals, represented by the gray dashed line, there is a big improvement gained by considering batches of platooned arrivals. In the next subsection we elaborate on the impact of dependencies in the arrival pattern on the performance of our algorithm.

4.3 Correlation

In the previous example it could be observed that traditional models for the FCTL queue, which assume independent arrivals per time slot, are not very applicable in a network setting where the traffic flow becomes a mixture of platoons and vehicles in free flow. The algorithm presented in the

present paper provides a way to capture the correlation between arrivals in successive time slots, but is limited to a single cycle. This implies that dependencies between arrivals in successive cycles cannot be handled by our method. In order to gain more insight in the impact of this limitation and to better understand the results of the Section 4.2, we use our simulation to estimate the autocorrelation function of the sequence of successive arrivals, both *within* and *between* cycles. In more detail, we keep track of the number of arrivals in each time slot and compute

$$\hat{\rho}(1) = \frac{\sum_{i=1}^{n-1}(x_i - \bar{x})(x_{i+1} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

i.e. the sample autocorrelation at lag 1, where $x_1, x_2, \ldots, x_n$ first represent the number of arrivals in time slot i (with n equal to the number of simulated cycles multiplied by c), and subsequently represent the total number of arrivals in cycle i (with n equal to the number of simulated cycles). We revisit the second example from Section 4.2, which discussed a network of ten intersections with inflow. For illustration purposes, we focus on the case $d = 5$. The simulated autocorrelations can be found in Table 3, first between arrivals in successive time slots, then between successive cycles. Clearly the autocorrelations for the first intersection, which has a Poisson process as input, are all equal to zero. Then, gradually, the autocorrelations increase for downstream intersections due to the forming of platoons. As the results indicate, this increase is much faster for the correlation between time slots than between cycles. This is the reason why our method is a big improvement compared to traditional models, but still becomes less accurate as the load increases beyond 0.75. Combining the results from Tables 2 and 3, it appears that this occurs when the autocorrelation between cycles grows beyond 0.20.

Correlation in the arrival pattern for $d = 5$										
Intersection	1	2	3	4	5	6	7	8	9	10
Load ρ	0.3	0.37	0.43	0.5	0.57	0.63	0.7	0.77	0.83	0.9
Between time slots	0.00	0.24	0.29	0.44	0.44	0.52	0.51	0.58	0.58	0.63
Between cycles	0.00	0.00	0.03	0.02	0.07	0.05	0.12	0.12	0.20	0.21

Table 3: Correlation between the number of arrivals in successive time slots (first row) and between successive cycles (second row), respectively, for the numerical example with $d = 5$ in Section 4.2.

One possible way to improve our algorithm even further is to take larger batches in the underlying generalized FCTL queue, spanning more than just one cycle. One would have to extend the model to allow for multiple green periods *within* one cycle. From a methodological point, this is not all too challenging, but it would require significantly more computational power to deal with the growing number of elements in the sets $\mathcal{G}_{j,l}$. A better alternative might be to simplify or approximate (3.7) for specific arrival patterns, such as platoons and free flow. However, this is beyond the scope of the present paper.

5   Conclusions

Classical models for fixed-cycle intersections do not cover scenarios where arriving traffic is correlated, as typically encountered in traffic networks. Motivated by this observation, we introduced

a generalized fixed-cycle traffic model that can deal with an arbitrary arrival patterns such as platoons and non-homogeneous traffic intensities. Generating-function methods for the stationary queue length distribution were used to develop an efficient numerical scheme for the performance analysis of an isolated intersection. We also proposed a network algorithm, which decomposes a network of intersections into separate generalized fixed-cycle network models, whose distributional output and input are matched according to the network layout. Extensive simulation experiments for a line network of multiple intersection support the network algorithm, and reveal only a slight loss of accuracy in heavy-traffic scenarios. We expect our network algorithm to work for many realistic network configurations, the study of which is an interesting topic for further research.

Acknowledgments

A Additional algorithms

Algorithm 5 Finding the roots inside the unit circle of $z^g - Y(z,z,\ldots,z)$

1: Input g and $Y(z,z,\ldots,z)$
2: Set $D(z) = z^g - Y(z,z,\ldots,z)$
3: Set the truncation parameter $n = g + 90$
4: Set $R = \{\}$
5: **while** $|R| \neq g - 1$ **do**
6:      $n = n + 10$
7:      Numerically compute $\tilde{D}(z) := \sum_{k=0}^{n} a_k z^k$,
8:          the n-th order Taylor expansion of $D(z)$ about $z = 0$
9:      Find the roots of $\tilde{D}(z)$ using a standard numerical polynomial root finding algorithm
10:     Denote these roots by $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n$
11:     $R = \{\}$
12:     **for** $i = 1$ to n **do**
13:         Numerically find a root z_i for $D(z)$ using starting point $z = \tilde{z}_i$
14:         **if** $z_i \notin R$ *and* $|z_i| < 1$ **then**
15:             add z_i to R
16:         **end if**
17:     **end for**
18: **end while**
19: **return** R

Algorithm 6 Creating all nonempty sets $\mathcal{G}_{j,l}$

1: $G_{0,0} = \{()\}$
2: **for** $l = 1$ to $g - 1$ **do**
3:      **for** $j = l$ to $g - 1$ **do**
4:          $G_{j,l} = \{\}$
5:          **for** $n_1 = \max(0, 2 - l)$ to $j - l$ **do**
6:              **for** $n_2 = \max(0, 3 - l - n_1)$ to $j - l - n_1$ **do**
7:                  $\vdots$
8:                      **for** $n_{j-2} = \max(0, j - 1 - l - n_1 - \cdots - n_{j-3})$ to $j - l - n_1 - \cdots - n_{j-3}$ **do**
9:                          $n_{j-1} = \max(0, j - l - n_1 - \cdots - n_{j-2})$
10:                         $n_j = 0$
11:                         $G_{j,l} = G_{j,l} \cup \{(n_1, n_2, \ldots, n_j)\}$
12:                     **end for**
13:                 $\vdots$
14:             **end for**
15:         **end for**
16:     **end for**
17: **end for**

B Details of numerical examples

For completeness and reproducibility, we give a detailed overview of the input parameters of all numerical examples in this paper.

Example Subsection 2.2. In this example we consider a generalized FCTL queue with a fixed-cycle of $c = g + r = 10 + 10 = 20$ time slots. Vehicles may arrive in time slots 1–10 and 16–18. The pgf of the joint distribution of $Y_1, Y_2, \ldots, Y_c$ is given by

$$Y(z_1, z_2, \ldots, z_{20}) = O^{(1)}(z_1, z_2, \ldots, z_{10})O^{(2)}(z_{16}, z_{17}, z_{18}), \tag{B.1}$$

where

$$
\begin{aligned}
O^{(1)}(z_1, z_2, \ldots, z_{10}) ={}& 0.0476\mathrm{e}^{0.3(z_1+z_2+z_3+z_4+z_5+z_6+z_7+z_8+z_9+z_{10}-10)} \\
&+ 0.107z_1\mathrm{e}^{0.3(z_2+z_3+z_4+z_5+z_6+z_7+z_8+z_9+z_{10}-9)} \\
&+ 0.143z_1z_2\mathrm{e}^{0.3(z_3+z_4+z_5+z_6+z_7+z_8+z_9+z_{10}-8)} \\
&+ 0.151z_1z_2z_3\mathrm{e}^{0.3(z_4+z_5+z_6+z_7+z_8+z_9+z_{10}-7)} \\
&+ 0.138z_1z_2z_3z_4\mathrm{e}^{0.3(z_5+z_6+z_7+z_8+z_9+z_{10}-6)} \\
&+ 0.114z_1z_2z_3z_4z_5\mathrm{e}^{0.3(z_6+z_7+z_8+z_9+z_{10}-5)} \\
&+ 0.0887z_1z_2z_3z_4z_5z_6\mathrm{e}^{0.3(z_7+z_8+z_9+z_{10}-4)} \\
&+ 0.0657z_1z_2z_3z_4z_5z_6z_7\mathrm{e}^{0.3(z_8+z_9+z_{10}-3)} \\
&+ 0.0470z_1z_2z_3z_4z_5z_6z_7z_8\mathrm{e}^{0.3(z_9+z_{10}-2)} \\
&+ 0.0328z_1z_2z_3z_4z_5z_6z_7z_8z_9\mathrm{e}^{0.3(z_{10}-1)} \\
&+ 0.0655z_1z_2z_3z_4z_5z_6z_7z_8z_9z_{10}, \\
O^{(2)}(z_1, z_2, z_3) ={}& 0.255\mathrm{e}^{0.075(z_1+z_2+z_3-3)} + 0.317z_1\mathrm{e}^{0.075(z_2+z_3-2)} \\
&+ 0.223z_1z_2\mathrm{e}^{0.075(z_3-1)} + 0.205z_1z_2z_3.
\end{aligned}
$$

Note that the pgfs $O^{(1)}(\cdot)$ and $O^{(2)}(\cdot)$ both have the form of the joint output pgf in (4.4). Each of them can be interpreted as the output of an upstream intersection with independent Poisson arrivals with intensities respectively 0.3 and 0.075 and green periods of respectively ten and three time slots. The probabilities $0.0476, 0.107, \ldots, 0.0655$ and $0.255, \ldots, 0.205$ correspond to the probability mass functions of the lengths of the effective green periods of the two upstream flows. Equivalently, one can also consider them as the distributions of the platoon sizes B_1 and B_2, see Figure 7.

Network example without side traffic, Subsection 4.2. In this network all ten intersections have synchronized settings with $c = g + r = 10 + 10 = 20$. The first intersection has external arrivals from a Poisson process with intensity $\lambda = 0.45$,

$$Y^{(1)}(z_1, z_2, \ldots, z_{20}) = \mathrm{e}^{0.45(z_1+z_2+\cdots+z_{20}-20)}.$$

Let $\rho = \lambda c/g$ denote the occupation rate, which can be interpreted as the fraction of utilized capacity at each intersection. Take $\lambda = 0.45$ and hence $\rho = 0.9$.
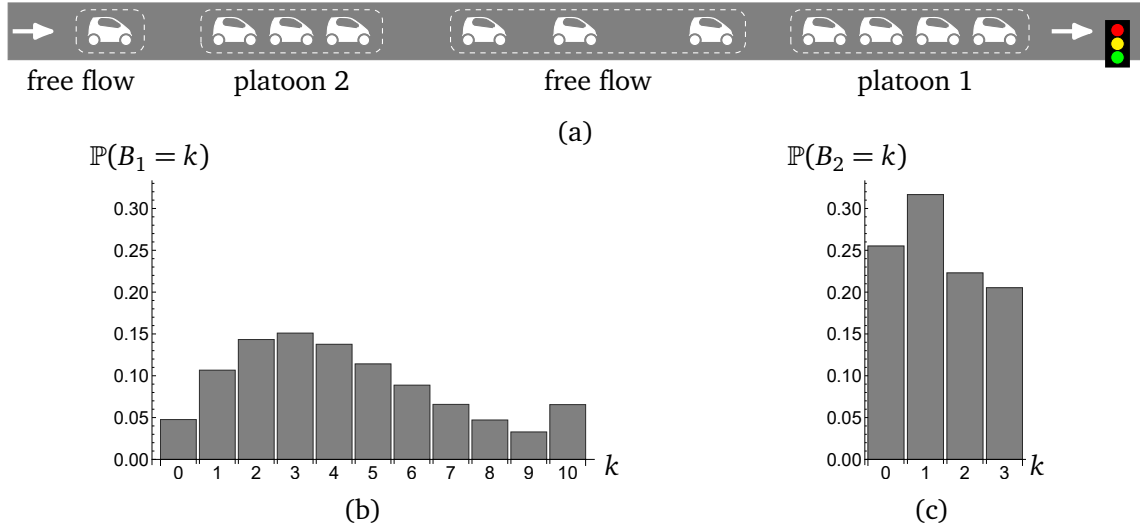
Figure 7: The arrival pattern in Section 2.2, and the probability distributions of the two platoon sizes B_1 and B_2. (a) Graphical representation of the arrival pattern; (b) Distribution of platoon 1 size; (c) Distribution of platoon 2 size.

The arrival patterns at the downstream intersections depend on the delay parameter d. For example, when $d = 5$, we have

$$
\begin{aligned}
Y^{(2)}(z_1, z_2, \ldots, z_{20}) = {} & 0.0052 e^{0.45(z_6+z_7+z_8+z_9+z_{10}+z_{11}+z_{12}+z_{13}+z_{14}+z_{15}-10)} \\
& + 0.015 z_6 e^{0.45(z_7+z_8+z_9+z_{10}+z_{11}+z_{12}+z_{13}+z_{14}+z_{15}-9)} \\
& + 0.028 z_6 z_7 e^{0.45(z_8+z_9+z_{10}+z_{11}+z_{12}+z_{13}+z_{14}+z_{15}-8)} \\
& + 0.039 z_6 z_7 z_8 e^{0.45(z_9+z_{10}+z_{11}+z_{12}+z_{13}+z_{14}+z_{15}-7)} \\
& + 0.048 z_6 z_7 z_8 z_9 e^{0.45(z_{10}+z_{11}+z_{12}+z_{13}+z_{14}+z_{15}-6)} \\
& + 0.054 z_6 z_7 z_8 z_9 z_{10} e^{0.45(z_{11}+z_{12}+z_{13}+z_{14}+z_{15}-5)} \\
& + 0.057 z_6 z_7 z_8 z_9 z_{10} z_{11} e^{0.45(z_{12}+z_{13}+z_{14}+z_{15}-4)} \\
& + 0.058 z_6 z_7 z_8 z_9 z_{10} z_{11} z_{12} e^{0.45(z_{13}+z_{14}+z_{15}-3)} \\
& + 0.057 z_6 z_7 z_8 z_9 z_{10} z_{11} z_{12} z_{13} e^{0.45(z_{14}+z_{15}-2)} \\
& + 0.055 z_6 z_7 z_8 z_9 z_{10} z_{11} z_{12} z_{13} z_{14} e^{0.45(z_{15}-1)} \\
& + 0.583 z_6 z_7 z_8 z_9 z_{10} z_{11} z_{12} z_{13} z_{14} z_{15}.
\end{aligned}
$$

Again, $0.0052, 0.015, \ldots, 0.583$ represent the probabilities that the platoon from the first intersection consists of, respectively, 0, 1, 2, ..., 10 vehicles. Due to the fact that the first intersection operates close to its maximum capacity, the maximum platoon size of ten vehicles is most likely to occur.

Network example with side traffic, Subsection 4.2. In this network we have ten queues in the main flow, $Q^{(1,0)}, \ldots, Q^{(10,0)}$, with synchronized settings $c^{(i,0)} = g^{(i,0)} + r^{(i,0)} = 10 + 10 = 20$ for

$i = 1, 2, \ldots, 10$. The first intersection has external arrivals from a Poisson process with intensity $\lambda = 0.15$,

$$Y^{(1,0)}(z_1, z_2, \ldots, z_{20}) = e^{0.15(z_1 + z_2 + \cdots + z_{20} - 20)}.$$

For $i = 2, 3, \ldots, 10$, the input of $Q^{(i,0)}$ consists of the delayed output of $Q^{(i-1,0)}$ and $Q^{(i-1,1)}$, see Figure 3.

The minor flows $Q^{(1,1)}, \ldots, Q^{(9,1)}$ have cycle lengths of twenty time slots, which makes it possible to synchronize them with the major flows. However, they have different green periods of *three* time slots taking place in time slots 16, 17, and 18 to avoid conflicts with the main traffic flow. The input to each of these minor flows is a Poisson process with intensity $\lambda^{(i,1)} = 1/30$, for $i = 1, 2, \ldots, 9$,

$$Y^{(i,1)}(z_1, z_2, \ldots, z_{20}) = e^{\frac{1}{30}(z_1 + z_2 + \cdots + z_{20} - 20)}.$$

We use $\rho^{(i,0)} = \mathbb{E}[Y^{(i,0)}]/g^{(i,0)}$ to denote the occupation rate of the main flow of intersection i. Due to the inflow of side traffic, the occupation rate of the main flows increases linearly from $\rho^{(1,0)} = 3/10$ to $\rho^{(10,0)} = 9/10$.

These are all the required input values. The performance of the network is determined in Section 4.2 using Algorithm 4.

References

[1] J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10:5–88, 1992.

[2] I. J. B. F. Adan, J. S. H. van Leeuwaarden, and E. M. M. Winands. On the application of Rouché's theorem in queueing theory. *Operations Research Letters*, 34(3):355–360, 2006.

[3] L. M. A. Bettencourt. The origins of scaling in cities. *Science*, 340(6139):1438–1441, 2013.

[4] G. Bitran and S. Dasu. A review of open queueing network models of manufacturing systems. *Queueing Systems*, 12:95–134, 1992.

[5] M. S. van den Broek, J. S. H. van Leeuwaarden, I. J. B. F. Adan, and O. J. Boxma. Bounds and approximations for the fixed-cycle traffic light queue. *Transportation Science*, 40(4):484–496, 2006.

[6] G. L. Choudhury and W. Whitt. Computing distributions and moments in polling models by numerical transform inversion. *Performance Evaluation*, 25:267–292, 1996.

[7] S. Çolak, A. Lima, and M. C. González. Understanding congested travel in urban areas. *Nature Communications*, 7, 2016.

[8] J.N. Darroch. On the traffic light queue. *Ann. Math. Statist.*, 35:380–388, 1964.

[9] F. Dion, H. Rakha, and Y.-S. Kang. Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections. *Transportation Research Part B*, 38:99–122, 2004.

[10] D. Heidemann. Queue length and delay distributions at traffic signals. *Transportation Research Part B*, 5:377–389, 1994.

[11] W. Henderson and P. G. Taylor. Product form in networks of queues with batch arrivals and batch services. *Queueing Systems*, 6(1):71–87, 1990.

[12] A. J. E. M. Janssen and J. S. H. van Leeuwaarden. Analytic computation schemes for the discrete-time bulk service queue. *Queueing Systems*, 50(2-3):141–163, 2005.

[13] A. J. E. M. Janssen and J. S. H. van Leeuwaarden. Back to the roots of the M/D/s queue and the works of Erlang, Crommelin and Pollaczek. *Statistica Neerlandica*, 62(3):299–313, 2008.

[14] P. Kühn. Approximate analysis of general queuing networks by decomposition. *IEEE Transactions on Communications*, 27:113–126, 1979.

[15] K. Laevens and H. Bruneel. Discrete-time multiserver queues with priorities. *Performance Evaluation*, 33(4):249–275, 1998.

[16] H. Levy and M. Sidi. Polling systems with simultaneous arrivals. *IEEE Transactions on Communications*, 39(6):823–827, 1991.

[17] D. R. McNeill. A solution to the fixed-cycle traffic-light problem for compound Poisson arrivals. *J. Appl. Probab.*, 5:624–635, 1968.

[18] G. K. S. Mung, C. K. Poon, and W. H. K. Lam. Distributions of queue lengths at fixed time traffic signals. *Transportation Research Part B*, 30(6):421–439, 1996.

[19] G. F. Newell. Approximation methods for queues with application to the fixed-cycle traffic light. *SIAM Review*, 7(2):223–240, 1965.

[20] C. Osorio and G. Flötteröd. Capturing dependency among link boundaries in a stochastic dynamic network loading model. *Transportation Science*, 49(2):420–431, 2015.

[21] C. Osorio, G. Flötteröd, and M. Bierlaire. Dynamic network loading: A stochastic differentiable model that derives link state distributions. *Transportation Research Part B*, 45, 2011.

[22] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti. Revisiting street intersections using slot-based systems. *PloS one*, 11(3):e0149607, 2016.

[23] J. S. H. van Leeuwaarden. Delay analysis for the fixed-cycle traffic-light queue. *Transportation Science*, 40(2):189–199, 2006.

[24] F. Viti and H. J. Van Zuylen. Probabilistic models for queues at fixed control signals. *Transportation Research Part B*, 44:120–135, 2010.

[25] F. V. Webster. Traffic signal settings. Technical Paper 39, Road Research Laboratory, 1958.

[26] W. Whitt. The queueing network analyzer. *Bell System Technical Journal*, 62(9):2779–2815, 1983.