



| | |
|--------------------------|--|
| Code | JMTS_2202_1_0R JMTS_2202_2402 |
| Name | Technical Report of JMTS_2202 Project |
| Deliberation Date | 2024.2.20 |
| Effective Date | 2024.2.25 |
| Content Structure | 5 Chapters |
| Authorization | JLRC ITC |
| Implementation | JMTS |
| Supervision | JNSC Concentrator of Academic Integrity (JCAI) |
| Accordance | JL000-202302 - 24 JL001-202302 - 89 (X) JL003-202308 JL007-202308 JL009-202308 |
| Publication Level | II-c2 |

The Theoretical Basis of A Matrix-based Coding Tool for Establishing Urban Road Network Topology Using Traffic Counts at Signalized Intersections

A Technical Report for JMTS_2202 Project
(Applicable to version 1.0)

Yunqing Jia

in

Intelligent Transportation System Research Center, Southeast University

Supervised by

Dr. Xiao Chen

in

College of Transportation Engineering, Chang'an University



Feb. 25th, 2024

ACKNOWLEDGEMENTS

The JMTS_2202 Project (initiated in February 2022) is an extended project based on the bachelor senior project of Yunqing Jia in the year 2021 supervised by Dr. Xiao Chen. Throughout the subsequent development and application process, Yunqing Jia is responsible for explaining all specific detailed problems as the principal investigator. All rights and the entire ownership of the project belong to Dr. Xiao Chen and College of Transportation Engineering, Chang'an University.

This report is edited on the basis of Latex template ETM Thesis Template IEST from Overleaf.

ABSTRACT

This report aims to reveal and demonstrate the theoretical basis of a matrix-based coding tool that can effectively generate matrices that capture the supply and demand pattern of urban road networks. The basic assumption of the tool is the hierarchical network topology considering flow conservation. The supply pattern is captured by several structural matrices while the demand pattern is described by model parameter matrices. The tool can be applied to macroscopic traffic modeling and this report uses the finite capacity queuing network model as an example. A detailed application process of the proposed coding tool is given for a toy urban network. The process is aligned with the step-by-step guidance in the user manual, attaching a thorough explanation of obtaining corresponding values for variables and elements in matrices.

Keywords: Matrix, urban road network, topology, traffic count, traffic signal control.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Acronyms and abbreviations | 2 |
| 1.2 | Notations | 2 |
| 2 | Network Topology Considering Flow Conservation | 5 |
| 2.1 | Hierarchical network topology | 5 |
| 2.2 | Consideration of flow conservation | 8 |
| 2.3 | Structural matrix for urban road network | 8 |
| 2.3.1 | Topological matrix | 8 |
| 2.3.2 | Traffic signal control matrix | 9 |
| 3 | Apply to Macroscopic Traffic Modeling: Example of Finite Capacity Queuing Network Model | 12 |
| 3.1 | Between-queue interactions | 12 |
| 3.2 | System of equations | 13 |
| 3.2.1 | Notations | 13 |
| 3.2.2 | System of equations | 14 |
| 3.3 | Determination of exogenous parameters using traffic counts | 16 |
| 3.3.1 | A simplified modeling method for large-scale queuing network | 16 |
| 3.3.2 | External arrival rate | 16 |
| 3.3.3 | Between-queue transition probability | 16 |
| 3.4 | General system of equations for grid-level programming | 17 |
| 4 | Case Study | 22 |
| 4.1 | Topological information | 23 |
| 4.1.1 | Channelization information for queues | 25 |

| | | |
|----------|---|-----------|
| 4.1.2 | Turning information for queues | 25 |
| 4.1.3 | Number of lanes information for queues | 25 |
| 4.1.4 | Signal phase information for queues | 26 |
| 4.1.5 | Segment information for queues | 26 |
| 4.1.6 | Relationship information for segments | 26 |
| 4.1.7 | Relationship information for queues | 27 |
| 4.2 | Flow information | 28 |
| 4.2.1 | External arrival rate | 28 |
| 4.2.2 | Transition probability | 29 |
| 4.3 | Signal timing information | 30 |
| 4.3.1 | Traffic signal information matrix for queues | 30 |
| 4.3.2 | Webster's method | 31 |
| 4.3.3 | Traffic signal information matrix for intersections | 33 |
| 5 | Conclusion | 34 |
| | References | 35 |
| A | Webster method | 36 |

List of Figures

| | | |
|-----|--|----|
| 2.1 | Hierarchical urban road network topology | 5 |
| 2.2 | Link-oriented network topology | 6 |
| 2.3 | Queue-oriented network topology | 7 |
| 4.1 | Example urban road network | 22 |
| 4.2 | Topological information | 23 |
| 4.3 | Queue traffic flow | 28 |

List of Tables

| | | |
|-----|---|----|
| 1.1 | Acronyms and abbreviations. | 2 |
| 1.2 | Notations (I) | 3 |
| 1.3 | Notations (II) | 4 |
| 4.1 | Traffic counts of the example network (Unit: veh/h) | 23 |
| 4.2 | Topological information and exogenous parameters | 24 |

Chapter 1

Introduction

When it comes to traffic management and control for urban road networks, the structural pattern of the network can be regarded as stable in certain periods (usually in months or years). Nevertheless, taking into account the differences among cities and districts, effectively forming customized macroscopic traffic models to provide the decision-making reference for urban traffic management control can contribute to more cost-effective congestion mitigation approaches. On the one hand, the considerable complexity of road network connectivity in large-scale metropolitan areas leads to tedious time for traffic signal engineers and researchers to build abstract models before obtaining the optimal management and control strategy. On the other hand, facing the time-varying demand (e.g., traffic volume) and supply (e.g., effective green time) pattern given the stable morphological characteristics of road networks within a monthly period, controllers need to correspondingly transfer the input raw data into estimated exogenous model parameters in a real-time basis so that the optimal solution can be adaptively obtained.

For large-scale urban road networks, the most common practice to collect traffic volume is to install electronic data-collecting devices like cameras or sensing loops on the road sections and around intersections. In this project, we use the original data collected from devices installed around the intersections. This kind of data is also known as traffic counts of turning volume. The road network topology and traffic signal control scheme is the supply side while the demand side can be described by traffic flows and/or Origin-Destination (OD) matrices. Compared with OD investigation, the traffic count data is more convenient to deal with. For those urban areas that fully facilitate loop detectors or video detection devices (e.g., license plate recognition (LPR) devices) around the signalized intersections,

it is easier and more accurate to capture the traffic demand by traffic counts of signalized intersections, especially under low connected vehicle (CV) penetration rates.

To effectively construct macroscopic traffic models for area-wide urban networks, a fast matrix-based method is proposed to model the topological structure of the road network which can be applicable to the finite capacity queuing network model. This report aims to demonstrate the technical details of the proposed method which thoroughly explains how the programming tool functions. The rest of the report is organized as follows: All acronyms, abbreviations, and notations related to this report are summarized in Chapter 1. Chapter 2 discusses the basic hierarchical framework of the proposed method considering flow conservation which is suitable for many urban road networks. Also, given a real-world road network, several structural matrices are introduced to capture the supply capacity of the network. In Chapter 3, we show the method of how to apply the proposed tool to the finite capacity queuing network model, in which the demand pattern of the road network is captured by the model's exogenous parameters, external arrival rates and transition probability. Chapter 4 provides a case study that designs a typical toy road network with assumed traffic counts. A step-by-step discussion can be found that explains where those output matrices come from and whether the automatically generated matrices are in accordance with manually calculated results that represent real-world circumstances. Chapter 5 gives the conclusion remarks and future works.

1.1 Acronyms and abbreviations

Table 1.1: Acronyms and abbreviations.

| Abbreviation | Definition | Abbreviation | Definition |
|--------------|--------------------|--------------|---------------------------|
| OD | Origin-Destination | LPR | License Plate Recognition |
| CV | Connected Vehicle | | |

1.2 Notations

Table 1.2: Notations (I)

| Basic element of the network | |
|---|---|
| i | index of one queue |
| I | total number of queues |
| N | set of natural numbers |
| \mathbb{R} | set of real numbers |
| A_I | maximum number of the upstream queues for a single queue |
| B_I | maximum number of the downstream queues for a single queue |
| S_{max} | maximum number of the corresponding signal phase for a single queue |
| P_M | total number of signal phases |
| I_n | total number of signalized intersections |
| i_n | maximum number of queues within one intersection |
| Structural matrix (supply pattern) | |
| A^0 | initial upstream queue index matrix |
| \bar{A}^0 | upstream relation matrix |
| A^1 | coded upstream queue index matrix |
| B^0 | initial downstream queue index matrix |
| \bar{B}^0 | downstream relation matrix |
| B^1 | coded downstream queue index matrix |
| S^0 | initial signal phase index matrix |
| \bar{S}^0 | signal relation matrix |
| S^1 | coded signal phase index matrix |
| \bar{S}^1 | signal control indication matrix |
| S^{01} | coded signal control indication matrix |
| \mathcal{P}^0 | original signal phase index matrix |
| \mathcal{P}^D | signal phase duration matrix |
| $\bar{\mathcal{P}}^0$ | intersection signal phase index matrix |
| $\bar{\mathcal{P}}^1$ | signal phase state matrix |
| $\bar{\mathcal{P}}^c$ | coded signal phase state matrix |
| Intersection-level variables | |
| g_{al} | actual green time of phase l [s] |
| g_{el} | effective green time of phase l [s] |
| q_l | flow rate of phase l |
| Y | sum of the flow ratio |
| L | total lost time [s] |
| t^l | lost time per phase [s] |
| n | number of phases of the intersection |
| AR | all-red time of the intersection [s] |
| \mathcal{P}_i^l | set of phase indices of queue i |
| n_i^l | number of lanes that share the same queue index i |
| A | yellow time [s] |

Table 1.3: Notations (II)

| Finite capacity queuing network model | |
|--|---|
| k_i | storage capacity [veh] |
| L_i | lane length [m] |
| d_{min} | minimal spacing between two vehicles [m] |
| \bar{d} | average length of a single vehicle [m] |
| γ_i | external arrival rate [veh/s] |
| λ_i | total arrival rate [veh/s] |
| μ | total service rate [veh/s] |
| $\tilde{\mu}_i$ | unblocking rate [veh/s] |
| $\hat{\mu}_i$ | effective service rate [veh/s] |
| ρ_i | traffic intensity |
| P_i^f | probability of being blocked by downstream spillback |
| N_i | total number of vehicles in queue [veh] |
| $P(N_i = k_i)$ | spillback probability |
| p_{ij} | transition probability from queue i to queue j |
| A_i | set of upstream queues of queue i |
| B_i | set of downstream queues of queue i |
| s | saturation flow rate [veh/s], usually 0.5 for a signalized lane |
| C_i | signal cycle length [s] |
| $\hat{\lambda}_i$ | effective arrival rate [veh/s] = $\lambda_i(1 - P(N_i = k_i))$ |
| $\hat{\rho}_i$ | effective traffic intensity = $\rho_i(1 - P(N_i = k_i))$ |
| Matrix in general system of equations | |
| K | general storage capacity matrix |
| N^l | general lane number matrix |
| Λ | general effective arrival rate matrix |
| P^N | general blocking probability matrix |
| ϱ | general traffic intensity matrix |
| Γ | general external arrival rate matrix |
| P^A | general upstream transition probability matrix |
| P^B | general downstream transition probability matrix |
| U | general effective upstream ingress matrix |
| D^1 | general egress spillback probability matrix |
| D^2 | general effective egress traffic intensity matrix |
| M | general service rate matrix |
| C | general cycle length matrix |

Chapter 2

Network Topology Considering Flow Conservation

2.1 Hierarchical network topology

The description of the road traffic network generally uses the method of graph theory. In the process of handling urban road network problems, usually, the transportation network is abstracted as an integration of nodes and links corresponding to the intersections and road sections. Figure 2.1 gives a general introduction to the hierarchy of the network topology. According to the graph theory, we define that the network is composed of nodes and links and the nodes are connected with each other by directed or undirected links.

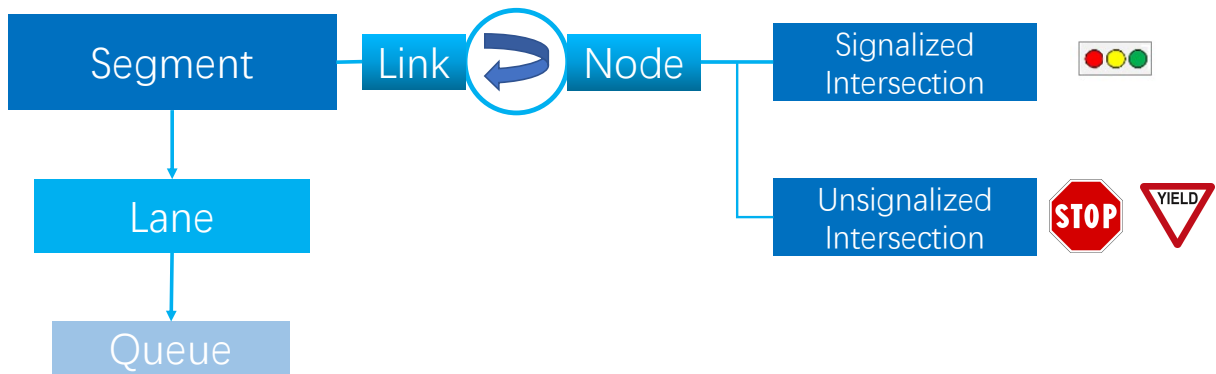


Figure 2.1: Hierarchical urban road network topology

Generally, intersections represented by nodes can be divided into signalized intersections and unsignalized intersections for various considerations such as cross-road hierarchy, traffic volume and safety. For those intersections that have high-class in-

intersecting roads or weak safety environments, traffic signal control is a friendly and cost-effective strategy for the traffic management sector to regulate high-volume traffic and protect vulnerable participants from severe accidents. For other intersections with lower traffic volume or slighter traffic conflict, there is less necessity for traffic signal control to improve the efficiency of the road network. In practice, yield-control design or stop-control design is set for safety considerations.

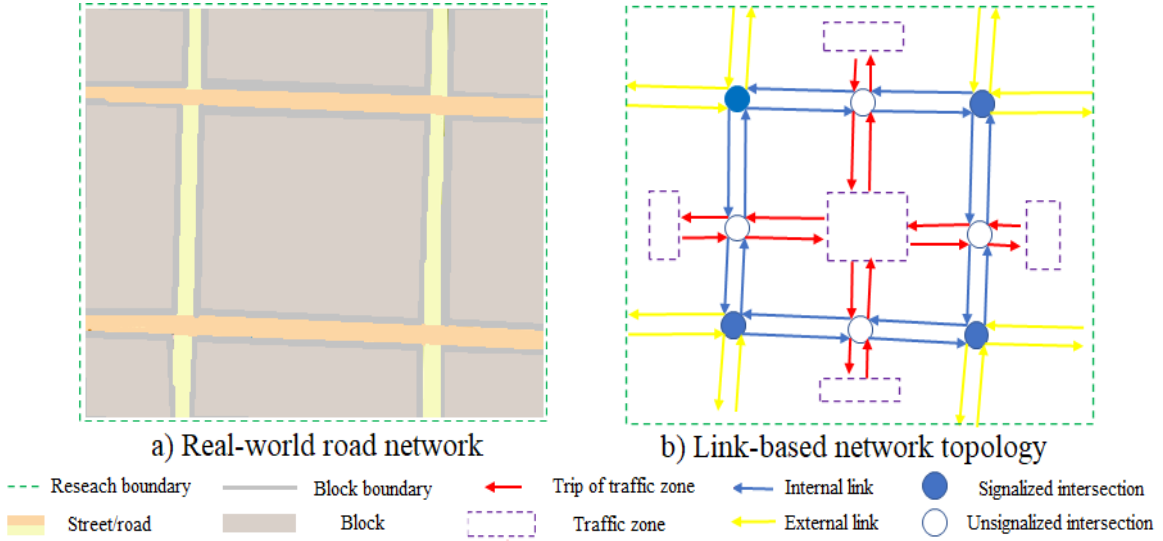


Figure 2.2: Link-oriented network topology

Once the signal arrangement is settled, the links between nodes should be analyzed. For the road network in the transportation system, each road section between two adjacent intersections is usually modeled as one or two directed links depending on one-way or two-way streets. Also, some road sections are connected with extraterritorial road sections which can be directly modeled as one link. From another perspective, the whole researched network is divided into many closed shapes by a set of links, and these shapes can represent blocks, communities, commercial districts and public institutions which can be abstracted as a set of traffic zones. In fact, the traffic zone is the point for trip attraction and generation followed by a set of OD pairs. It can be also described as that some unsignaled intersection is set for connecting internal links with the traffic zones from outside. Figure 2.2 illustrates the link-oriented network topology which gives an example of how a road network from the real world is transferred into analyzed network topology. The green dotted line for both sub-figures is the research boundary

which separates the research road network from the extraterritorial areas. The deep gray solid line is the boundary of the block, which is represented by a light gray rectangle. The flesh and light-yellow rectangles are streets and roads in the real world. For link-based network topology, the red arrows represent trips (traffic flows) generated from or attracted by the traffic zones (bounded by the purple dotted rectangle), and the yellow arrows represent the external links which connect research network with extraterritorial areas, both of them lead to external arrivals. The blue arrows are internal links that correspond to the internal arrivals between adjacent queues. The blue solid circle and the blue hollow circle represent signalized and unsignalized intersections, respectively.

Here, we primarily pay attention to the hierarchy of the road network topology. The interactions among traffic zones, links and extraterritorial road sections will be discussed in the next section.

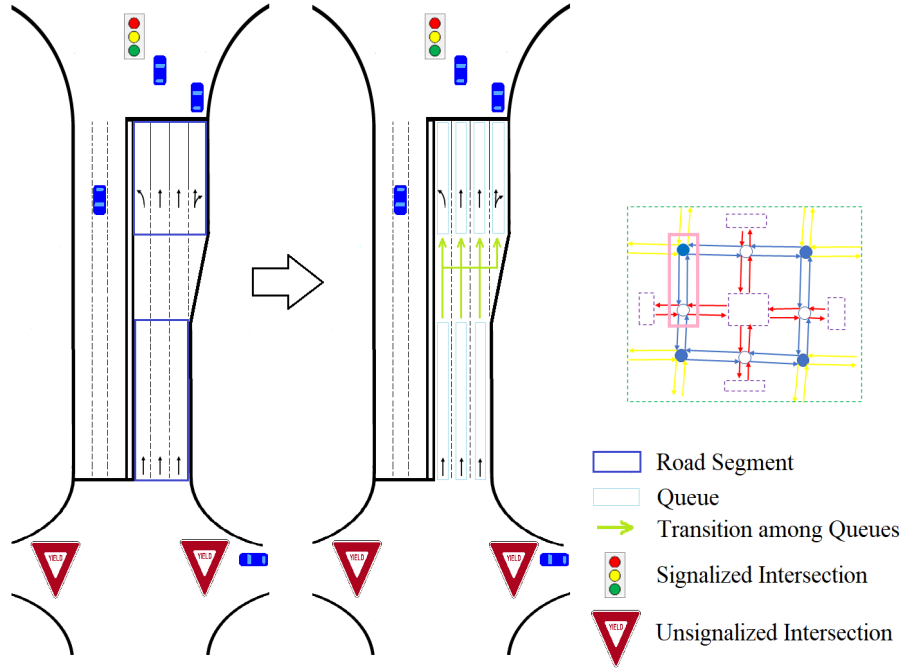


Figure 2.3: Queue-oriented network topology

Next, each link can be divided into one or a set of segments, where each road section (segment) has the same number of lanes. When it comes to an intersection or a variation of the number of lanes such as road widening or narrowing, the boundary of the segment occurs to separate one road section into many. Then, for each segment with the same number of lanes, each lane is modeled as a queue

regardless of its width, slope and pavement material. Figure 2.3 gives a typical example of how a road section is separated into a set of segments and how lanes in each segment are modeled into a set of queues.

2.2 Consideration of flow conservation

Besides capturing the interactions between traffic zones and the internal road links, another reason for setting an unsignalized intersection between two signalized intersections is to consider flow conservation. In most cases, the sum of the turning flows at the downstream signalized intersection (wishing to exit the link) does not equal the sum of the corresponding turning flows at the upstream signalized intersection wishing to enter the link. So, the unsignalized intersection is set to balance this difference. In this report, we assume that if the downstream flow sum is larger than the upstream flow sum, then the flow difference refers to those flows that enter the road network from the unsignalized intersection generated by the traffic zone. Conversely, if the downstream flow sum is smaller than the upstream flow sum, then the flow difference refers to those flows that exit the road network from the unsignalized intersection attracted by the traffic zone.

2.3 Structural matrix for urban road network

After the set of the queuing model is completed, we need to identify the upstream queues, downstream queues, and signal control mode for each queue. With this information, we can obtain the topological matrix and the traffic signal control matrix for the entire research road network. Note that i is the index of one queue and I represents all queues within the research network.

2.3.1 Topological matrix

The topological matrix consists of matrices that can capture upstream and downstream relations among queues. Given a queuing model with I queues in total, we first introduce the initial upstream queue index matrix $A^0 \in N^{I \times A_I}$, in which A_I is the maximum number of the upstream queues for a single queue. Each row represents a queue in the network model and each column represents an upstream queue. For those queues that have no upstream queues or have less than I upstream

queues, the corresponding upstream queue index is 0. Next, we can obtain a binary matrix $\bar{A}^0 \in N^{I \times A_I}$ (called upstream relation matrix) to capture the existence of the upstream queues.

$$\bar{A}_{ij}^0 = \begin{cases} 0, & A_{ij}^0 = 0, \\ 1, & A_{ij}^0 > 0. \end{cases} \quad (2.1)$$

For some matrix-based programming languages such as MATLAB (2022), the index for matrices can not be zero. So, we replace the initial upstream queue index matrix $A^0 \in N^{I \times A_I}$ with the coded upstream queue index matrix $A^1 \in N^{I \times A_I}$, in which

$$A_{ij}^1 = \begin{cases} 1, & A_{ij}^0 = 0, \\ A_{ij}^0, & A_{ij}^0 > 0. \end{cases} \quad (2.2)$$

Note that after the replacement, there are two different types of index ‘1’ in the coded upstream queue index matrix. One is that queue ‘1’ is one of the upstream queues for certain queues, with the corresponding element in $\bar{A}^0 \in N^{I \times A_I}$ equals to one. The other is that means no (or no longer) upstream queues for certain queues, with the corresponding element in $\bar{A}^0 \in N^{I \times A_I}$ equals zero. While indexing the upstream queues, the upstream relation matrix must be applied together.

Similarly, we can obtain the downstream relation matrix $\bar{B}^0 \in N^{I \times B_I}$ and the coded downstream queue index matrix $B^1 \in N^{I \times B_I}$ based on the initial downstream queue index matrix $B^0 \in N^{I \times B_I}$ (B_I is the maximum number of the downstream queues for a single queue).

$$\bar{B}_{ij}^0 = \begin{cases} 0, & B_{ij}^0 = 0, \\ 1, & B_{ij}^0 > 0. \end{cases} \quad (2.3)$$

$$B_{ij}^1 = \begin{cases} 1, & B_{ij}^0 = 0, \\ B_{ij}^0, & B_{ij}^0 > 0. \end{cases} \quad (2.4)$$

2.3.2 Traffic signal control matrix

In general, queues can be categorized into signalized and unsignalized queues. Signalized queues are usually controlled by at least one signal phase and each signal

phase corresponds to one signal timing. To link each signalized queue with the corresponding signal phase (timing), we first introduce the initial signal phase index matrix $\mathcal{S}^0 \in N^{I \times \mathcal{S}_{max}}$, in which \mathcal{S}_{max} is the maximum number of the corresponding signal phase for a single queue. For unsignalized queues or signalized queues that have less than \mathcal{S}_{max} signal phases, the corresponding index is 0. Similar to the upstream (downstream) queue index, we can also obtain a signal relation matrix $\bar{\mathcal{S}}^0 \in N^{I \times \mathcal{S}_{max}}$ and the coded signal phase index matrix $\mathcal{S}^1 \in N^{I \times \mathcal{S}_{max}}$ based on the initial signal phase index matrix.

$$\bar{\mathcal{S}}_{ij}^0 = \begin{cases} 0, & \mathcal{S}_{ij}^0 = 0, \\ 1, & \mathcal{S}_{ij}^0 > 0. \end{cases} \quad (2.5)$$

$$\mathcal{S}_{ij}^1 = \begin{cases} 1, & \mathcal{S}_{ij}^0 = 0, \\ \mathcal{S}_{ij}^0, & \mathcal{S}_{ij}^0 > 0. \end{cases} \quad (2.6)$$

Next, we introduce a binary matrix $\bar{\mathcal{S}}^1 \in N^{I \times \mathcal{S}_{max}}$ (signal control indication matrix) to identify whether each queue is signalized, in which each row represents a queue. For elements in the first column, if one queue is signalized, the element is 1, and 0 otherwise. To keep all signal control matrix dimensions consistent, all elements in other columns are filled with 0.

$$\bar{\mathcal{S}}_{ij}^1 = \begin{cases} -\bar{\mathcal{S}}_{ij}^0 + 1, & j = 1, \\ 0, & j > 1. \end{cases} \quad (2.7)$$

Then, another binary matrix called the coded signal control indication matrix $\mathcal{S}^{01} \in N^{I \times \mathcal{S}_{max}}$ is designed.

$$\mathcal{S}_{ij}^{01} = \begin{cases} 1, & j = 1, \\ \bar{\mathcal{S}}_{ij}^0, & j > 1. \end{cases} \quad (2.8)$$

Suppose we have an urban network that has P_M signal phases and I_n signalized intersections, with each phase being numbered accordingly as in the original signal phase index matrix $\mathcal{P}^0 \in N^{1 \times P_M} = [1 \ 2 \ \dots \ P_M]$ (also we have a signal phase duration matrix $\mathcal{P}^D \in N^{1 \times P_M} = [g_{a1} \ g_{a2} \ \dots \ g_{aP_M}]$, where g_{a1} is the actual green time of phase 1). Here, the intersection signal phase index matrix $\bar{\mathcal{P}}^0 \in N^{I_n \times P_m}$ is proposed (P_m is the maximum number of signal phases within one intersection). In this matrix,

each row represents one intersection. For those intersections that have less than P_m signal phases, the corresponding element is 0. Thus, we can also obtain a signal phase state matrix $\bar{\mathcal{P}}^1 \in N^{I_n \times P_m}$, where

$$\bar{\mathcal{P}}^1 = \begin{cases} 0, & \bar{\mathcal{P}}_{ij}^0 = 0, \\ 1, & \bar{\mathcal{P}}_{ij}^0 > 0. \end{cases} \quad (2.9)$$

Similarly, we have the coded signal phase state matrix $\bar{\mathcal{P}}^c \in N^{I_n \times P_m}$ as well.

$$\bar{\mathcal{P}}^c = \begin{cases} 1, & \bar{\mathcal{P}}_{ij}^0 = 0, \\ \bar{\mathcal{P}}_{ij}^0, & \bar{\mathcal{P}}_{ij}^0 > 0. \end{cases} \quad (2.10)$$

Note that in $\bar{\mathcal{P}}^c$, there are also two kinds of ‘1’: the one that represents phase 1 and those elements that do not correspond to any signal phases (with $\bar{\mathcal{P}}_{ij}^1 = 0$).

Chapter 3

Apply to Macroscopic Traffic Modeling: Example of Finite Capacity Queuing Network Model

For real-world urban road networks with intricate traffic behavior and sophisticated management strategy, the macroscopic traffic model provides a general and global approach to describe the status of the traffic flow through a simplified but analytical methodology that is in accordance with the actual physical tendency of the microscopic traffic model. In this section, we introduce the finite capacity queuing network model proposed in the previous research by Osorio and Bierlaire (2009) and Osorio (2010) which is known as a metamodel for solving large-scale traffic signal control problems.

3.1 Between-queue interactions

As mentioned above, for a set of intersections, we need to identify the upstream-downstream relationships between queues once the queue modeling is completed. Also, we need to distinguish the internal arrivals from the internal network and external network. For each queue in the network, it can be roughly categorized into two types: the one that connects with extraterritorial road sections and the one that connects with queues or traffic zones within the researched network. Nevertheless, each queue is modeled as an $M/M/1/k$ queue and has an arrival rate, a service rate, and the storage capacity k according to the finite capacity queuing model.

Each queue's arrival rate is composed of the external arrival rate and internal arrival rate, usually in the vehicle per hour or vehicle per second. The external

arrival rate means the customer arrives from the outside of the researched network and has no connections with the status of the upstream queues. The external arrival rate occurs when one queue's upstream connects with extraterritorial road sections or traffic zones that generate trips to the network. When there are upstream queues within the studied network, the internal arrival rate of the queue happens as the between-queue interactions of the arrival rate.

The service rate of each queue is the saturation flow rate multiplied by the corresponding green split, which is also dependent on the corresponding characteristics. If the downstream of the queue is connected with the extraterritorial road sections, unsignalized intersections or other queues from another segment, the green split equals 1. If the queue's downstream is connected with the signalized intersection, the green split equals the ratio of the summation of the green time of all phases which gives right-of-way to this lane (queue) and the total cycle time of the signalized intersection.

For storage capacity k , also known as the upper bound of each queue i , it is deeply connected with the lane's jam density which determines the number of vehicles in waiting when the queue is full. Equation 3.1 gives a calculation method (notice the one-to-one correspondence between lane and queue).

$$k_i = \lfloor \frac{L_i + d_{min}}{\bar{d} + d_{min}} \rfloor \quad (3.1)$$

where L_i represents the length of lane i (in meters), d_{min} is the minimal spacing between two vehicles (in meters), and \bar{d} denotes the average length of a single vehicle (in meters). The final result of the storage capacity is an integer, which means the maximum number of vehicles in queue i .

3.2 System of equations

3.2.1 Notations

The following notations are introduced ahead for queue i to illustrate the system of equations combined with the between-queue interactions mentioned above.

| | |
|-----------------|---|
| γ_i | external arrival rate [veh/s]; |
| λ_i | total arrival rate [veh/s]; |
| μ | total service rate [veh/s]; |
| $\tilde{\mu}_i$ | unblocking rate [veh/s]; |
| $\hat{\mu}_i$ | effective service rate [veh/s]; |
| ρ_i | traffic intensity; |
| P_i^f | probability of being blocked by downstream spillback; |
| N_i | total number of vehicles in queue [veh]; |
| $P(N_i = k_i)$ | spillback probability; |
| p_{ij} | transition probability from queue i to queue j ; |
| A_i | set of upstream queues of queue i ; |
| B_i | set of downstream queues of queue i . |

According to Chong (2017) and Chen et al. (2019), two additional variables are defined as follows to further simplify the queuing network model in the number of equations. The concept of effective means multiplying the probability of not occurring spillback.

| | |
|-------------------|---------------------------------|
| $\hat{\lambda}_i$ | effective arrival rate [veh/s]; |
| $\hat{\rho}_i$ | effective traffic intensity; |

3.2.2 System of equations

Next, we outline the core content of the finite capacity queuing network model which has been given in detail in Chapter 4 of Osorio (2010). The formulation is based on the physical components mentioned above which capture the entire arrival and departure process for each queue.

Equation (3.2a) follows the rule of flow conservation, which means that the arrival rate of queue i equals the summation of the arrivals from outside network and upstream queues. If the queue is blocked by the arrivals (also known as spillback), the subsequent arrivals can't arrive in this queue, so Equation (3.2a) provides a calculation approach that multiplies arrival rates with the probability that queue i is not full. Correspondingly, Equation (3.2b) calculates the unblocking rate which is the rate that dissipates the downstream spillback of queue i . Equation (3.2c) gives the definition of the effective service rate, which is the function of the ideal service rate of queue i , of the probability of being blocked at queue i because of

downstream spillback which is defined as Equation (3.2d), and of the unblocking rate for downstream spillback of queue i . Equation (3.2e) defines the spillback probability of queue i , which is calculated by the traffic density defined in Equation (3.2f).

$$\left\{ \begin{array}{l} \lambda_i = \gamma_i + \frac{\sum_{h \in A_i} p_{hi} \lambda_h (1 - P(N_h = k_h))}{1 - P(N_i = k_i)} \quad (3.2a) \\ \frac{1}{\tilde{\mu}_i} = \sum_{j \in B_i} \frac{\lambda_j (1 - P(N_j = k_j))}{\lambda_i (1 - P(N_i = k_i)) \hat{\mu}_i} \quad (3.2b) \\ \frac{1}{\hat{\mu}_i} = \frac{1}{\mu_i} + P_f^i \frac{1}{\tilde{\mu}_i} \quad (3.2c) \\ P_i^f = \sum_{j \in B_i} p_{ij} P(N_j = k_j) \quad (3.2d) \\ P(N_i = k_i) = \frac{1 - \rho_i}{1 - \rho_i^{k+1}} \rho_i^k \quad (3.2e) \\ \rho_i = \frac{\lambda_i}{\hat{\mu}_i} \quad (3.2f) \end{array} \right.$$

Then, combined with the original system of equations, the simplified tractable system of equations can be obtained as follows.

$$\left\{ \begin{array}{l} \hat{\lambda}_i = \gamma_i (1 - P(N_i = k_i)) + \sum_{h \in A_i} p_{hi} \hat{\lambda}_h \quad (3.3a) \\ \hat{\rho}_i = \frac{\hat{\lambda}_i}{\mu_i} + \left(\sum_{j \in B_i} p_{ij} P(N_j = k_j) \right) \left(\sum_{j \in B_i} \hat{\rho}_j \right) \quad (3.3b) \\ P(N_i = k_i) = \frac{1 - \hat{\rho}_i}{1 - \hat{\rho}_i^{k+1}} \hat{\rho}_i^k \quad (3.3c) \end{array} \right.$$

3.3 Determination of exogenous parameters using traffic counts

3.3.1 A simplified modeling method for large-scale queuing network

For high-density road networks with several lanes on each link, there is a way to simplify modeling and programming procedures while guaranteeing the model's accuracy. According to Osorio (2010), if a queue connects to a segment, it means that this queue is connected to all the queues in that segment. On this basis, if several queues have the same arrival rates (same upstream queues), service rates (same downstream queues and controlled by the same signal), and upper bound, they can share the same queue indices. Two common conditions are suitable for this simplification. One is that the approaching lanes of the intersection have the same function of turning, and the other is that the exit lanes of the intersection.

Although it appears that the number of the queue indices decreases after the simplification, it must be cautious while calculating the arrival rate and transition probability of each queue. Still, the whole process is governed by the rule of flow conservation. We then provide the calculation method for transferring traffic volume into arrival rate and transition probability considering the two conditions mentioned above.

3.3.2 External arrival rate

For queues on the exit lanes or do not directly connect with the traffic zones, the external arrival rate is zero. As mentioned in Section 2.2, the flow difference equals the outflow from the downstream signalized intersection minus the inflow from the upstream signalized intersection (the inflow of internal links is zero). If the difference is zero or negative, the external rate of the queue linking to the downstream intersection is 0. Otherwise, the difference should be distributed according to the downstream turning flow. Note that one queue corresponds to one lane, so the external arrival rates should be normalized as “veh/s/lane” following the internal arrival rates.

3.3.3 Between-queue transition probability

In this section, we mainly focus on the transition probability between the target queue and the upstream queue, which refers to the “ p_{hi} ” in Equation (3.2a) and

(3.3a). The transition probability between the target queue and the downstream queue can be obtained based on the “ p_{hi} ”. First, the transition probability of queues on the external link is zero for there is no upstream queue. Next, the transition probability of queues on the exit lanes equals the reciprocal of the number of exit lanes multiplied by the proportion of vehicles in the upstream queue turning to the target queue. If the upstream queue corresponds to a dedicated turning lane, the proportion is 1. Finally, the transition probability of queues corresponding to the approaching lanes equals the reciprocal of the number of lanes that share the same queue index multiplied by the proportion of vehicles in the upstream queue turning to the target queue.

3.4 General system of equations for grid-level programming

Combined with the structural matrix proposed in Section 2.3.2 and the simplified modeling method proposed in Section 3.3, we can formulate the general system of equations for the simplification of grid-level programming.

Let us assume that after the simplification method, the given urban road network can be modeled by I queues in total. On top of the following steps, we can initially determine the storage capacity k_i and the number of lanes that share with the same queue indices (i.e., have the same arrival and service rate) for each queue i , thus forming up the general storage capacity matrix $K = [k_1 \ k_2 \ \dots \ k_I], k_i \in \mathbb{R}^{1 \times I}, i \in \{1, 2, \dots, I\}$ and the general lane number matrix $N^l = [n_1^l \ n_2^l \ \dots \ n_I^l], n_i^l \in \mathbb{R}^{1 \times I}, i \in \{1, 2, \dots, I\}$, respectively. In addition, we set the general effective arrival rate matrix $\Lambda = [\hat{\lambda}_1 \ \hat{\lambda}_2 \ \dots \ \hat{\lambda}_I], \hat{\lambda}_i \in \mathbb{R}^{1 \times I}, i \in \{1, 2, \dots, I\}$, the general blocking probability matrix $P^N = [P(N_1 = k_1) \ P(N_2 = k_2) \ \dots \ P(N_I = k_I)], P(N_i = k_i) \in \mathbb{R}^{1 \times I}, i \in \{1, 2, \dots, I\}$, and the general traffic intensity matrix $\varrho = [\rho_1 \ \rho_2 \ \dots \ \rho_I], \rho_i \in \mathbb{R}^{1 \times I}, i \in \{1, 2, \dots, I\}$.

Besides, following the definitions in Section 2.3.2, we can obtain the upstream &

downstream relation matrix (i.e., $\bar{A}^0 =$

$$\begin{bmatrix} \bar{A}_{11}^0 & \cdots & \bar{A}_{1j}^0 & \cdots & \bar{A}_{1A_I}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{A}_{i1}^0 & \cdots & \bar{A}_{ij}^0 & \cdots & \bar{A}_{iA_I}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{A}_{I1}^0 & \cdots & \bar{A}_{Ij}^0 & \cdots & \bar{A}_{IA_I}^0 \end{bmatrix}$$

, $\bar{A}_{ij}^0 \in N^{I \times A_I}, i \in \{1, 2, \dots, I\}, j \in \{1, 2, \dots, A_I\}$ and $\bar{B}^0 =$

$$\begin{bmatrix} \bar{B}_{11}^0 & \cdots & \bar{B}_{1j}^0 & \cdots & \bar{B}_{1B_I}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{B}_{i1}^0 & \cdots & \bar{B}_{ij}^0 & \cdots & \bar{B}_{iB_I}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{B}_{I1}^0 & \cdots & \bar{B}_{Ij}^0 & \cdots & \bar{B}_{IA_I}^0 \end{bmatrix}$$

, $\bar{B}_{ij}^0 \in N^{I \times B_I}, i \in \{1, 2, \dots, I\}, j \in \{1, 2, \dots, B_I\}$, in which A_I and B_I are the maximum numbers of the upstream and downstream queues for a single queue, respectively). Similarly, we can also obtain the coded upstream & downstream queue index matrix (i.e., $A^1 =$

$$\begin{bmatrix} A_{11}^1 & \cdots & A_{1j}^1 & \cdots & A_{1A_I}^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{A}_{i1}^0 & \cdots & \bar{A}_{ij}^0 & \cdots & \bar{A}_{iA_I}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{A}_{I1}^0 & \cdots & \bar{A}_{Ij}^0 & \cdots & \bar{A}_{IA_I}^0 \end{bmatrix}$$

, $A_{ij}^1 \in N^{I \times A_I}, i \in \{1, 2, \dots, I\}, j \in \{1, 2, \dots, A_I\}$ and $B^1 =$

$$\begin{bmatrix} B_{11}^1 & \cdots & B_{1j}^1 & \cdots & B_{1B_I}^1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{B}_{i1}^0 & \cdots & \bar{B}_{ij}^0 & \cdots & \bar{B}_{iB_I}^0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{B}_{I1}^0 & \cdots & \bar{B}_{Ij}^0 & \cdots & \bar{B}_{IA_I}^0 \end{bmatrix}$$

, $B_{ij}^1 \in N^{I \times B_I}, i \in \{1, 2, \dots, I\}, j \in \{1, 2, \dots, B_I\}$).

Based on the traffic count data, we can first obtain the external arrival rate γ_i for each queue i , which is the element of the general external arrival rate matrix $\Gamma = [\gamma_1 \ \gamma_2 \ \dots \ \gamma_I], \gamma_i \in \mathbb{R}^{1 \times I}, i \in \{1, 2, \dots, I\}$. Then, the upstream transition probability p_{hi}^A and the downstream transition probability p_{ij}^B can be calculated referring to the rules in Section 3.3, and correspondingly can acquire the general upstream transition probability matrix $P^A =$

$$\begin{bmatrix} p_{11}^A & \cdots & p_{1i}^A & \cdots & p_{1A_I}^A \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{h1}^A & \cdots & p_{hi}^A & \cdots & p_{hA_I}^A \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{I1}^A & \cdots & p_{Ih}^A & \cdots & p_{IA_I}^A \end{bmatrix}$$

, $h \in \{1, 2, \dots, I\}, i \in \{1, 2, \dots, A_I\}$, and the general downstream transition probability matrix $P^B =$

$$\begin{bmatrix} p_{11}^B & \cdots & p_{1j}^B & \cdots & p_{1B_I}^B \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{i1}^B & \cdots & p_{ij}^B & \cdots & p_{iB_I}^B \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{I1}^B & \cdots & p_{Ij}^B & \cdots & p_{IB_I}^B \end{bmatrix}$$

, $i \in \{1, 2, \dots, I\}, j \in \{1, 2, \dots, B_I\}$.

Next, we introduce the general matrix U , D^1 , and D^2 for effective upstream ingress $\sum_{h \in A_i} p_{hi} \hat{\lambda}_h$, egress spillback probability $(\sum_{j \in B_i} p_{ij} P(N_j = k_j))$, and effective egress traffic intensity $(\sum_{j \in B_i} \hat{\rho}_j)$, respectively. Therefore, constrained by the rule of flow conservation, we can reorganize the system of equations of the finite queuing network model. The definition of the three general matrices is defined as follows:

$$U = \sum_{j=1}^{A_I} \bar{A}^0_{:,j} \circ p^A_{:,j} \circ (N^l(A_{1,A^1_{:,j}}) \circ \Lambda_{1,A^1_{:,j}}) \quad (3.4)$$

$$D^1 = \sum_{j=1}^{B_I} \bar{B}^0_{:,j} \circ p^B_{:,j} \circ (N^l(B_{1,B^1_{:,j}}) \circ P^N(B_{1,B^1_{:,j}})) \quad (3.5)$$

$$D^2 = \sum_{j=1}^{B_I} \bar{B}^0_{:,j} \circ (N^l(B_{1,B^1_{:,j}}) \circ \varrho(B_{1,B^1_{:,j}})) \quad (3.6)$$

Here, we define two additional rules of computation similar to the Hadamard (elementwise) product. It is well known that for two matrices $a, b \in \mathbb{R}^{I \times J}$, we have:

$$\begin{aligned} a \circ b &= \begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{I1} & \cdots & a_{Ij} & \cdots & a_{IJ} \end{bmatrix} \circ \begin{bmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i1} & \cdots & b_{ij} & \cdots & b_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{I1} & \cdots & b_{Ij} & \cdots & b_{IJ} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} \times b_{11} & \cdots & a_{1j} \times b_{1j} & \cdots & a_{1J} \times b_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} \times b_{i1} & \cdots & a_{ij} \times b_{ij} & \cdots & a_{iJ} \times b_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{I1} \times b_{I1} & \cdots & a_{Ij} \times b_{Ij} & \cdots & a_{IJ} \times b_{IJ} \end{bmatrix} \end{aligned} \quad (3.7)$$

Analogously, we define the elementwise quotient “ $\circ \div$ ” and the elementwise exponent “ \circ° ” as follows:

$$\begin{aligned} a \circ \div b &= \begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{I1} & \cdots & a_{Ij} & \cdots & a_{IJ} \end{bmatrix} \circ \div \begin{bmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i1} & \cdots & b_{ij} & \cdots & b_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{I1} & \cdots & b_{Ij} & \cdots & b_{IJ} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} \div b_{11} & \cdots & a_{1j} \div b_{1j} & \cdots & a_{1J} \div b_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} \div b_{i1} & \cdots & a_{ij} \div b_{ij} & \cdots & a_{iJ} \div b_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{I1} \div b_{I1} & \cdots & a_{Ij} \div b_{Ij} & \cdots & a_{IJ} \div b_{IJ} \end{bmatrix} \end{aligned} \quad (3.8)$$

$$\begin{aligned}
a \circ^\circ b &= \begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{I1} & \cdots & a_{Ij} & \cdots & a_{IJ} \end{bmatrix} \circ^\circ \begin{bmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1J} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i1} & \cdots & b_{ij} & \cdots & b_{iJ} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{I1} & \cdots & b_{Ij} & \cdots & b_{IJ} \end{bmatrix} \\
&= \begin{bmatrix} a_{11}^{b_{11}} & \cdots & a_{1j}^{b_{1j}} & \cdots & a_{1J}^{b_{1J}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1}^{b_{i1}} & \cdots & a_{ij}^{b_{ij}} & \cdots & a_{iJ}^{b_{iJ}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{I1}^{b_{I1}} & \cdots & a_{Ij}^{b_{Ij}} & \cdots & a_{IJ}^{b_{IJ}} \end{bmatrix}
\end{aligned} \tag{3.9}$$

Finally, we can formulate the general system of equations for grid-level programming:

$$\begin{cases} \Lambda = \Gamma \circ (E^{1 \times I} - P^N) + U & (3.10a) \\ \varrho = \Lambda \circ \div M + D^1 \circ D^2 & (3.10b) \\ P^N = (E^{1 \times I} - \varrho) \circ (\varrho \circ^\circ (K)) \circ \div (E^{1 \times I} - \varrho \circ^\circ (K + E^{1 \times I})) & (3.10c) \end{cases}$$

where M is the general service rate matrix $M = [\mu_1 \ \mu_2 \ \dots \ \mu_I]$, $\mu_i \in \mathbb{R}^{1 \times I}$, $i \in \{1, 2, \dots, I\}$. For queue i 's service rate μ_i , it is equal to the green split multiplied by the saturation flow rate s (\mathcal{P}_i^l is the set of phase indices of queue i).

$$\mu_i = s \frac{\sum_{l \in \mathcal{P}_i^l} g_{el}}{C_i} \tag{3.11}$$

For each queue, we can utilize the traffic signal control matrices established in Section 2.3.2 to simplify the calculation workload of the service rate. These matrices have classified all queues into signalized and unsignalized, and they also help to link each signalized queue with the corresponding phase duration.

$$M = s \left(\sum_{j=1}^{P_m} (\mathcal{P}^D(\mathcal{S}_{:,j}^1) \circ \bar{\mathcal{S}}_{:,j}^0 + \bar{\mathcal{S}}_{:,j}^1 \circ \mathcal{C}^T) \circ \mathcal{S}_{:,j}^{01} \right)^T \circ \div \mathcal{C} \tag{3.12}$$

where $\mathcal{C} = [\mathcal{C}_1 \ \mathcal{C}_2 \ \dots \ \mathcal{C}_I]$, $\mathcal{C}_i \in N^{1 \times I}$, $i \in \{1, 2, \dots, I\}$ is the general cycle length matrix.

Chapter 4

Case Study

We design a simple yet typical urban road network in this chapter to illustrate the applied procedure of the proposed matrix-based method. The topology of the example network is shown in Figure 4.1, which consists of two signalized intersections and one unsignalized intersection. The assumed traffic counts collected by traffic monitoring devices are listed in Table 4.1.

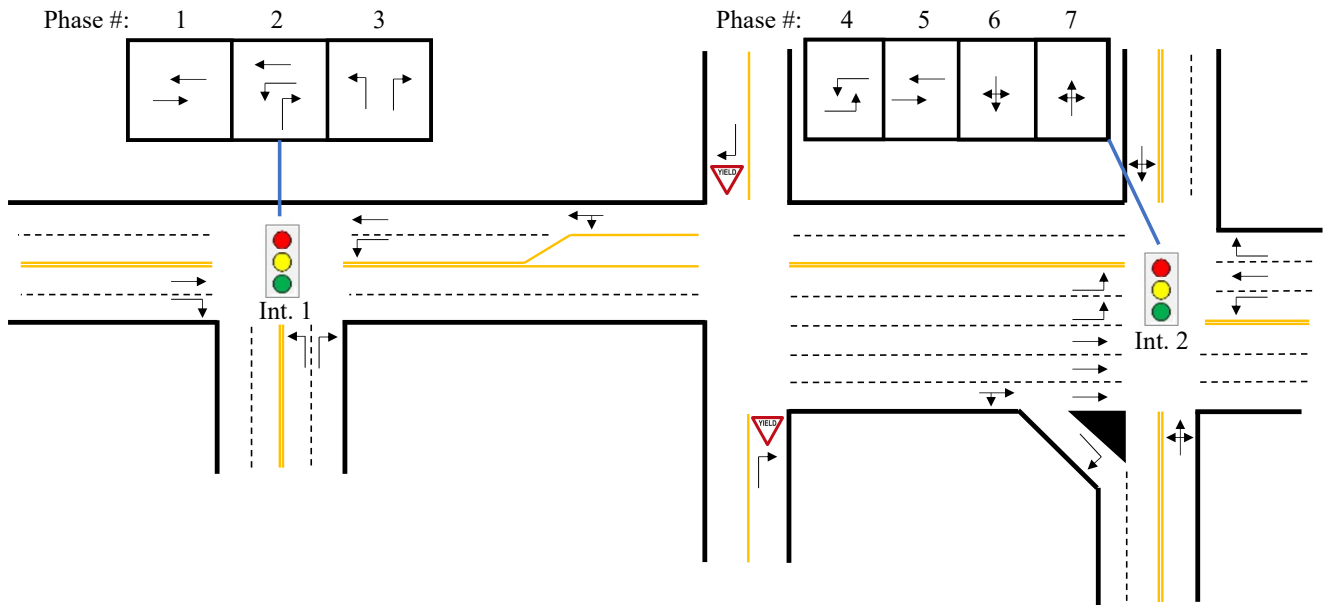


Figure 4.1: Example urban road network

Table 4.1: Traffic counts of the example network (Unit: veh/h)

| Intersection | Eastbound | | | Northbound | | | Westbound | | | Southbound | | |
|--------------|-----------|---------|-------|------------|---------|-------|-----------|---------|-------|------------|---------|-------|
| | Left | Through | Right | Left | Through | Right | Left | Through | Right | Left | Through | Right |
| 1 | 0 | 550 | 400 | 700 | 0 | 650 | 200 | 400 | 0 | 0 | 0 | 0 |
| 2 | 400 | 2100 | 300 | 50 | 40 | 30 | 200 | 800 | 100 | 30 | 40 | 50 |

4.1 Topological information

According to certain principles described in Section 2.1, lanes in the proposed urban road network can be modeled into several queues. Based on the simplification method proposed in Section 3.3, those queues that share the same arrival rate, service rate and storage capacity can be amalgamated (represented by one queue index). Hence, we can obtain the topological information as shown in Figure 4.2, modeling the network in 24 queues (total number of queues, $I = 24$). It can be also concluded from the figure that the total number of signalized intersections I_n is 2, while the maximum number of queues within one intersection i_n is 14 (see Intersection 2).

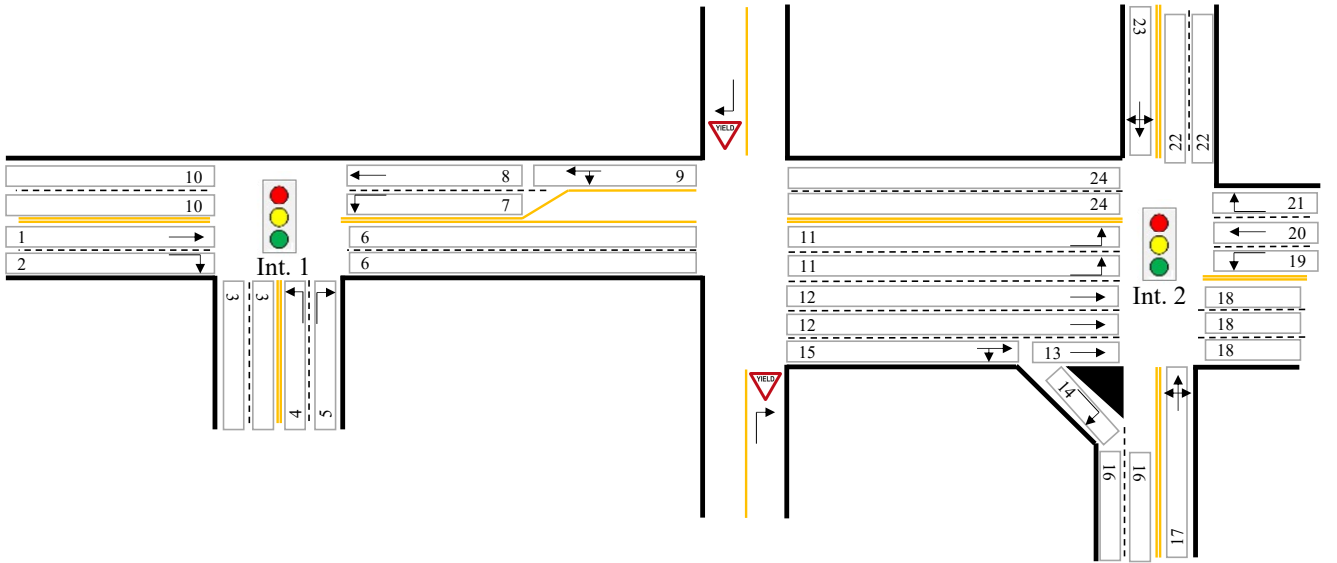


Figure 4.2: Topological information

Table 4.2 summarizes the overall topological information and exogenous parameters for all modeled queues. Here, we locally numbered 24 queues in a counter-clockwise order in each intersection as in the second column of the table (represent

corresponding (row, column) in the following matrices), and correspondingly numbered all queues globally from 1 to 24 as in the third column. For example, queue No. 11 is the first queue being numbered in Intersection 2, and it links with the 2^{nd} -row- 1^{st} -column element in the following matrices.

Table 4.2: Topological information and exogenous parameters

| Intersection indices | Queue indices (i) | | Turning | Number of lanes N^{lT} | Flow (veh/h/lane) | External arrival rates (veh/h/lane) Γ^T | Upstream queue indices (h) | p_{hi} |
|----------------------|-----------------------|----|---------|--------------------------|-------------------|--|--------------------------------|----------|
| 1 | (1,1) | 1 | T | 1 | 550 | 550 | - | 0 |
| | (1,2) | 2 | R | 1 | 400 | 400 | - | 0 |
| | (1,3) | 3 | E | 2 | 300 | 0 | (1,2) | 0.5 |
| | | | | | | | (1,7) | 0.5 |
| | (1,4) | 4 | L | 1 | 700 | 700 | - | 0 |
| | (1,5) | 5 | R | 1 | 650 | 650 | - | 0 |
| | (1,6) | 6 | E | 2 | 600 | 0 | (1,1) | 0.5 |
| | | | | | | | (1,5) | 0.5 |
| | (1,7) | 7 | L | 1 | 200 | 0 | (1,9) | 0.333 |
| | (1,8) | 8 | T | 1 | 400 | 0 | (1,9) | 0.667 |
| 2 | (1,9) | 9 | LT | 1 | 600 | 0 | (2,14) | 0.667 |
| | (1,10) | 10 | E | 2 | 550 | 0 | (1,4) | 0.5 |
| | | | | | | | (1,7) | 0.5 |
| | (2,1) | 11 | L | 2 | 200 | 114.286 | (1,6) | 0.071 |
| | (2,2) | 12 | T | 2 | 700 | 400 | (1,6) | 0.25 |
| | (2,3) | 13 | T | 1 | 700 | 0 | (1,9) | 0.7 |
| | (2,4) | 14 | R | 1 | 300 | 0 | (1,9) | 0.3 |
| | (2,5) | 15 | TR | 1 | 1000 | 571.428 | (2,14) | 0.357 |
| | (2,6) | 16 | E | 2 | 270 | 0 | (2,4) | 0.5 |
| | | | | | | | (2,9) | 0.5 |
| | | | | | | | (2,13) | 0.167 |
| | (2,7) | 17 | LTR | 1 | 120 | 120 | - | 0 |
| | (2,8) | 18 | E | 3 | 720 | 0 | (2,2) | 0.333 |
| | | | | | | | (2,3) | 0.333 |
| | | | | | | | (2,7) | 0.083 |
| | | | | | | | (2,13) | 0.083 |
| | (2,9) | 19 | L | 1 | 200 | 200 | - | 0 |
| | (2,10) | 20 | T | 1 | 800 | 800 | - | 0 |
| | (2,11) | 21 | R | 1 | 100 | 100 | - | 0 |
| | (2,12) | 22 | E | 2 | 270 | 0 | (2,1) | 0.5 |
| | | | | | | | (2,7) | 0.167 |
| | | | | | | | (2,11) | 0.5 |
| | (2,13) | 23 | LTR | 1 | 120 | 120 | - | 0 |
| | (2,14) | 24 | E | 2 | 450 | 0 | (2,7) | 0.208 |
| | | | | | | | (2,10) | 0.5 |
| | | | | | | | (2,13) | 0.208 |

4.1.1 Channelization information for queues

There are generally two kinds of channelization schemes: the left-turn storage lane and the right-turn channelization island (similar to the right-turn storage lane). For queues belonging to downstream lanes, the corresponding value of the element in the channelization information matrix equals 1, and 2 for upstream queues. All other normal queues and nonexistent queues are 0. So, the channelization information matrix of the example network is shown as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4.1.2 Turning information for queues

For those queues on approaching lanes, each queue has specific turning according to lane functions. The element of the turning information matrix is filled with different lane functions: left-turn (L), through (T), right-turn (R), left-through (LT), through-right (TR), left-right-turn (LR), and all turns (LTR) for a single lane. For queues on exit lanes, we use “E” as the element of the turning information matrix. Combined with the topological information in Figure 4.2, we can obtain the turning information matrix for the example network:

$$\begin{Bmatrix} T & R & E & L & R & E & L & T & LT & E & \square & \square & \square & \square \\ L & T & T & R & TR & E & LTR & E & L & T & R & E & LTR & E \end{Bmatrix}$$

4.1.3 Number of lanes information for queues

In the fifth column of Table 4.2, the number of lanes for each queue (see N^l) is listed according to the rules discussed in Section 3.3. The lane number matrix can be reshaped based on N^l as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 1 & 2 & 1 & 4 & 1 & 1 & 1 & 2 & 1 & 2 \end{bmatrix}$$

4.1.4 Signal phase information for queues

For two signalized intersections, let us assume that the designed signal phase sequences as Figure 4.1. Seven signal phases have been numbered accordingly (total number of signal phases $P_M = 7$ and maximum number of signal phases within one intersection $P_m = 4$ (in Intersection 2)). For un-signalized queues (e.g., queue No. 10), the element of queue signal phase matrix is 0.5 (also known as the default value of saturated flow rate 0.5 veh/s). If a queue is controlled by one phase (e.g., queue No. 1), that element is equal to the phase number. If a queue is controlled by two phases (e.g., queue No. 5), that element shall be written as ‘[phase-1 phase-2]’. In this case, the maximum number of the corresponding signal phase for a single queue, $S_{max} = 2$.

$$\left\{ \begin{array}{cccccccccccccccc} 1 & 0.5 & 0.5 & 3 & [2 & 3] & 0.5 & 2 & [1 & 2] & 0.5 & 0.5 & [] & [] & [] & [] \\ 4 & 5 & 5 & 0.5 & 0.5 & 0.5 & 7 & 0.5 & 4 & 5 & 0.5 & 0.5 & 6 & 0.5 \end{array} \right\}$$

4.1.5 Segment information for queues

Based on the segment-lane-queue hierarchy (Section 2.1), each queue is linked to a segment in an intersection. In this version of the tool, the segments are numbered counter-clockwise with the eastbound being 1. Hence, we can get the queue-segment matrix:

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 1 \end{bmatrix}$$

4.1.6 Relationship information for segments

Each segment is numbered locally in the last step, so we need two numbers to specifically represent segments as ‘[Intersection-index Segment number]’. As for the upstream-downstream relationships among segments, we use the segment relationship information matrix, in which the columns refer to all segment numbers and the rows represent all intersection indices. If the segment has no upstream segment, the element is 0. If it has, the element is the global number of the upstream segment ‘[Intersection-index Segment number]’. For the toy network, its segment relationship information matrix is:

$$\left\{ \begin{array}{ccccc} 0 & 0 & [2 & 1] & 0 \\ [1 & 3] & 0 & 0 & 0 \end{array} \right\}$$

4.1.7 Relationship information for queues

After the completion of the above steps, particularly with the assistance of the queue turning information and the segment relationship information, the between-queue relationships (i.e., column 8 of Table 4.2) can be reshaped into two matrices (the initial upstream & downstream queue index matrix A^0, B^0 in Section 2.3.2) that can show upstream-downstream relationship for all queues. Based on the network topology, for the example network, the maximum number of the upstream queues $A_I = 4$ and that of the downstream queues $B_I = 3$. And the initial upstream and downstream queue index matrix, A^0 and B^0 , can be obtained.

$$A^0 = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 1 & 9 & 9 & 24 & 4 & 6 & 6 & 15 & 15 & 6 & 14 & 0 & 12 & 0 & 0 & 0 & 11 & 0 & 17 \\ 0 & 0 & 7 & 0 & 0 & 5 & 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 19 & 0 & 13 & 0 & 0 & 0 & 17 & 0 & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 23 & 0 & 17 & 0 & 0 & 0 & 21 & 0 & 23 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 23 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$B^0 = \begin{bmatrix} 6 & 3 & 0 & 10 & 6 & 11 & 3 & 10 & 7 & 0 & 22 & 18 & 18 & 16 & 13 & 0 & 18 & 0 & 16 & 24 & 22 & 0 & 16 & 9 \\ 0 & 0 & 0 & 0 & 0 & 12 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 14 & 0 & 22 & 0 & 0 & 0 & 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 & 0 & 0 & 0 & 0 & 24 & 0 \end{bmatrix}^T$$

According to the rules in Section 2.3.2, we can get the upstream & downstream relation matrix (\bar{A}^0 and \bar{B}^0) and the coded upstream & downstream queue index matrix (A^1 and B^1).

$$\bar{A}^0 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\bar{B}^0 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$A^1 = \begin{bmatrix} 1 & 1 & 2 & 1 & 1 & 1 & 9 & 9 & 24 & 4 & 6 & 6 & 15 & 15 & 6 & 14 & 1 & 12 & 1 & 1 & 1 & 11 & 1 & 17 \\ 1 & 1 & 7 & 1 & 1 & 5 & 1 & 1 & 1 & 8 & 1 & 1 & 1 & 1 & 1 & 19 & 1 & 13 & 1 & 1 & 1 & 17 & 1 & 20 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 23 & 1 & 17 & 1 & 1 & 1 & 21 & 1 & 23 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 23 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$$

$$B^1 = \begin{bmatrix} 6 & 3 & 1 & 10 & 6 & 11 & 3 & 10 & 7 & 1 & 22 & 18 & 18 & 16 & 13 & 1 & 18 & 1 & 16 & 24 & 22 & 1 & 16 & 9 \\ 1 & 1 & 1 & 1 & 1 & 12 & 1 & 1 & 8 & 1 & 1 & 1 & 1 & 1 & 14 & 1 & 22 & 1 & 1 & 1 & 1 & 1 & 18 & 1 \\ 1 & 1 & 1 & 1 & 1 & 15 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 24 & 1 & 1 & 1 & 1 & 1 & 24 & 1 \end{bmatrix}^T$$

4.2 Flow information

Till now, the topological information details have been captured by six matrices. Next, Figure 4.3 assigned the traffic counts in Table 4.1 to each queue (column 6 in Table 4.2).

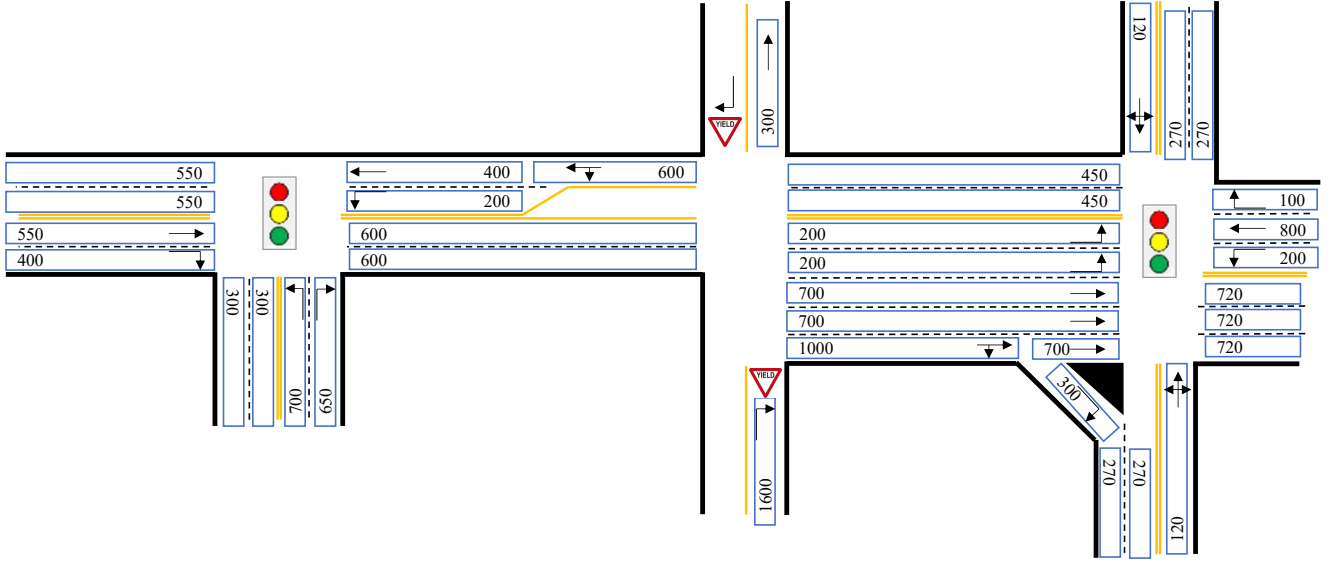


Figure 4.3: Queue traffic flow

4.2.1 External arrival rate

The external arrival rates for all queues are listed in column 7 of Table 4.2 (the general external arrival rate matrix Γ^T). Here, we show the detailed calculation

process of queue No. 11 (2,1) which represents the left-turn lanes of the eastbound Int.2: The segment [2 1] receives 1200 vehicles from the upstream segment [1 3] while departing 2800 vehicles, which indicates that 1600 vehicles enter from the outside network as external arrivals. Assume we allocate these external arrivals proportionally to queues No. 11, 12, and 15, The external arrival rate of queue No.11 can be calculated as the following equation:

$$\gamma_{11} = 1600 \times \frac{200}{200 + 200 + 700 + 700 + 1000} \approx 114.286 \text{ veh/h/lane} \quad (4.1)$$

4.2.2 Transition probability

The calculation of the transition probability is highly related to the upstream-downstream inter-queue relationships. The results are shown in column 9 of Table 4.2. We demonstrate the calculation process of two typical queues (No. 11 and No. 18).

Queue No. 11 has an external arrival rate of 114.286 veh/h, and its upstream queue is queue No. 6. So, we should exclude the external arrivals from the outside network while calculating the transition probability. Next, the flow of queue No. 6 is 600 veh/h, thus we can gain the final result as follows:

$$p_{6-11} = \frac{(200 - 114.286)/2}{600} = \frac{(85.714)/2}{600} \approx 0.071 \quad (4.2)$$

note that 85.714 vehicles are coming from the upstream segment [1 3] which has two lanes. For each lane, it uniformly transfers 85.714/2 vehicles to one lane of queue No. 11, and that is why we have that division by 2.

Queue No. 18 has no external arrivals but four upstream queues (No. 12, 13, 17, 23). All vehicles in Queue No. 12 and 13 go to Queue No. 18. For queue No. 12, two lanes' through flows connect with three lanes. So, for each lane of queue No. 18, we have:

$$p_{12-18} = \frac{(700)/3}{700} = \frac{(233.333)}{700} \approx 0.333 \quad (4.3)$$

and similarly:

$$p_{13-18} = \frac{(700)/3}{700} = \frac{(233.333)}{700} \approx 0.333 \quad (4.4)$$

Queue No. 17 has 30 veh/h right-turn flow with a total flow of 120 veh/h. These 30 vehicles will be uniformly distributed to three lanes of queue No. 18. As a result, we can get:

$$p_{17-18} = \frac{(30)/3}{120} = \frac{(10)}{120} \approx 0.083 \quad (4.5)$$

and similarly:

$$p_{23-18} = \frac{(30)/3}{120} = \frac{(10)}{120} \approx 0.083 \quad (4.6)$$

The general upstream & downstream transition probability matrix P^A and P^B are as follows:

$$A^0 = \begin{bmatrix} 0 & 0 & .5 & 0 & 0 & .5 & .333 & .667 & .667 & .5 & .071 & .25 & .7 & .3 & .357 & .5 & 0 & .333 & 0 & 0 & 0 & .5 & 0 & .208 \\ 0 & 0 & .5 & 0 & 0 & .5 & 0 & 0 & 0 & .5 & 0 & 0 & 0 & 0 & 0 & .5 & 0 & .333 & 0 & 0 & 0 & .167 & 0 & .5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .167 & 0 & .083 & 0 & 0 & 0 & .5 & 0 & .208 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .083 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$B^0 = \begin{bmatrix} .5 & .5 & 0 & .5 & .5 & .071 & .5 & .5 & .333 & 0 & .5 & .333 & .333 & .5 & .7 & 0 & .083 & 0 & .5 & .5 & .5 & 0 & .167 & .667 \\ 0 & 0 & 0 & 0 & 0 & .25 & 0 & 0 & .667 & 0 & 0 & 0 & 0 & 0 & .3 & 0 & .167 & 0 & 0 & 0 & 0 & .083 & 0 \\ 0 & 0 & 0 & 0 & 0 & .357 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .208 & 0 & 0 & 0 & 0 & .208 & 0 \end{bmatrix}^T$$

4.3 Signal timing information

4.3.1 Traffic signal information matrix for queues

Referring to Section 2.3.2, we first gain the initial signal phase index matrix \mathcal{S}^0 for the network. Here, the original signal phase index matrix $\mathcal{P}^0 = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$.

$$\mathcal{S}^0 = \begin{bmatrix} 1 & 0 & 0 & 3 & 2 & 0 & 2 & 1 & 0 & 0 & 4 & 5 & 5 & 0 & 0 & 0 & 7 & 0 & 4 & 5 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Therefore, we can also obtain the signal relation matrix $\bar{\mathcal{S}}^0$ and the coded signal phase index matrix \mathcal{S}^1 based on \mathcal{S}^0 :

$$\bar{\mathcal{S}}^0 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\mathcal{S}^1 = \begin{bmatrix} 1 & 1 & 1 & 3 & 2 & 1 & 2 & 1 & 1 & 1 & 4 & 5 & 5 & 1 & 1 & 1 & 7 & 1 & 4 & 5 & 1 & 1 & 6 & 1 \\ 1 & 1 & 1 & 1 & 3 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$$

Finally, we have the signal control indication matrix $\bar{\mathcal{S}}^1$ and the coded signal control indication matrix \mathcal{S}^{01}

$$\bar{\mathcal{S}}^1 = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 \end{bmatrix}^T$$

$$\mathcal{S}^{01} = \begin{bmatrix} 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

4.3.2 Webster's method

We apply Webster's method Webster (1958) to get the signal timings for the example network. A step-by-step introduction to the Webster method can be found in Appendix A. In this section, we provide the detailed calculation process for the toy network.

For both intersections, the sum of the flow ratio Y is calculated (assume the saturation flow rate equals 1800 veh/h/lane for all lanes):

$$Y_I = \frac{550 + 200 + 700}{1800} = 1450/1800 \quad (4.7)$$

$$Y_{II} = \frac{200 + 800 + 120 + 120}{1800} = 1240/1800 \quad (4.8)$$

Assume no all-red time for both intersections, and we can get the total lost time L .

$$L_I = 3 \times 3 = 9 \quad s \quad (4.9)$$

$$L_{II} = 4 \times 3 = 12 \quad s \quad (4.10)$$

Next, we can find the best cycle length C that can be divided evenly by 5.

$$C_I = \frac{1.5 \times 9 + 5}{1 - 1450/1800} \approx 100 \quad s \quad (4.11)$$

$$C_{II} = \frac{1.5 \times 12 + 5}{1 - 1240/1800} \approx 75 \quad s \quad (4.12)$$

Consequently, the general cycle length matrix $\mathcal{C} =$

$$[100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 100 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75 \quad 75]$$

Then, the effective green time g_e for each phase can be figured based on the total effective green time G_e :

$$G_{eI} = 100 - 9 = 91 \quad s \quad (4.13)$$

$$G_{eII} = 75 - 12 = 63 \quad s \quad (4.14)$$

$$g_{e1} = 91 \times \frac{550/1800}{1450/1800} \approx 34 \quad s \quad (4.15)$$

$$g_{e2} = 91 \times \frac{200/1800}{1450/1800} \approx 13 \quad s \quad (4.16)$$

$$g_{e3} = 91 \times \frac{700/1800}{1450/1800} \approx 44 \quad s \quad (4.17)$$

$$g_{e4} = 63 \times \frac{200/1800}{1240/1800} \approx 10 \quad s \quad (4.18)$$

$$g_{e5} = 63 \times \frac{800/1800}{1240/1800} \approx 41 \quad s \quad (4.19)$$

$$g_{e6} = g_{e7} = 63 \times \frac{120/1800}{1240/1800} \approx 6 \quad s \quad (4.20)$$

Finally, the actual green times are calculated and are also the elements of the signal phase duration matrix \mathcal{P}^D .

$$\mathcal{P}^D = [37 \ 16 \ 47 \ 13 \ 44 \ 9 \ 9]^T$$

4.3.3 Traffic signal information matrix for intersections

The phase indices for each intersection are arranged in order from smallest to largest in each row of the intersection signal phase index matrix $\bar{\mathcal{P}}^0$. The signal phase state matrix and the coded signal phase state matrix are also listed as follows.

$$\bar{\mathcal{P}}^0 = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 4 & 5 & 6 & 7 \end{bmatrix} \quad \bar{\mathcal{P}}^1 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \bar{\mathcal{P}}^c = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 7 \end{bmatrix}$$

Eventually, we can finish this case by calculating the service rate for all queues according to the method in Section 3.4.

$$M = [.17 \ .5 \ .5 \ .22 \ .3 \ .5 \ .065 \ .25 \ .5 \ .5 \ .067 \ .273 \ .273 \ .5 \ .5 \ .5 \ .04 \ .5 \ .067 \ .273 \ .5 \ .5 \ .04 \ .5]$$

Here, we show the calculation process of the service rate under Webster's scheme for Queue No. 5 and 17:

$$\mu_5 = 0.5 \times \frac{16 + 47 - 3}{100} = 0.03 \tag{4.21}$$

and similarly:

$$\mu_{17} = 0.5 \times \frac{9 - 3}{75} = 0.04 \tag{4.22}$$

Chapter 5

Conclusion

This report introduces the theoretical basis of a proposed matrix-based programming tool that can use network topology and traffic counts at signalized intersections to generate matrices that can capture the supply and demand pattern for large-scale urban networks. The hierarchical network topology has been discussed to decompose the urban road network while the rule of flow conservation is strictly followed. Based on the real-world network topology and traffic signal control schemes, several structural matrices are raised to link queues with their upstream & downstream queues and the corresponding signal phases. To apply the tool for macroscopic traffic modeling, we discuss the procedure of establishing the finite capacity queuing network model as an example. In this model, the demand pattern is mainly captured by two exogenous parameters, the external arrival rate and the transition probability, and the supply pattern is represented by the service rate, all of which can be effectively calculated via the proposed tool. A simple but typical case study is demonstrated in detail which validates the effectiveness of the proposed tool. The example network has been embedded in the open access code. Users can find that the matrices automatically generated by the tool are aligned with the results calculated by the step-by-step procedure.

Further study can focus on the influence of this tool on the accuracy of the macroscopic traffic model in terms of capturing network-level traffic flow dynamics. Also, with additional matrices involved in solving the system of equations, it is unknown whether the introduction of those matrices would significantly affect the computational efficiency. What's more, the U-turn will be included in the channelization information matrix which will further influence the traffic flow assignment among queues.

References

- Chen, X., Osorio, C., and Santos, B. F. (2019). Simulation-based travel time reliable signal control. *Transportation Science*, 53(2):523–544.
- Chong, L. (2017). *Computationally efficient simulation-based optimization algorithms for large-scale urban transportation problems*. PhD thesis, Massachusetts Institute of Technology.
- MATLAB (2022). *version 9.12.0.2039608 (R2022a)*. The MathWorks Inc., Natick, Massachusetts.
- Osorio, C. (2010). *Mitigating network congestion: analytical models, optimization methods and their applications*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne.
- Osorio, C. and Bierlaire, M. (2009). A surrogate model for traffic optimization of congested networks: an analytic queueing network approach. Technical report.
- Webster, F. V. (1958). Traffic signal settings. Technical report.

Appendix A

Webster method

We introduce how we calculate the signal timing by the Webster method (a traditional method for isolated intersections).

Step 1: Calculate the sum of the flow ratio Y according to Equation (A.1) based on Equation (A.2):

$$y_l = \frac{q_l}{s} \quad (\text{A.1})$$

$$Y = \sum_l y_l \quad (\text{A.2})$$

y_l : the flow rate of each phase; q_l : the traffic volume of critical lane l (veh/h); s : the saturation traffic flow rate (veh/h);

Step 2: Figure out the total lost time L according to Equation (A.3):

$$L = nt^l + AR \quad (\text{A.3})$$

t^l : the lost time per phase, usually equals 3 s if no accurate data (s); n : the number of phases of the intersection; AR : the all-red time for the intersection (s);

Step 3: Figure out the best cycle length calculated by the Webster method C according to Equation (A.4):

$$C = \frac{1.5l + 5}{1 - Y} \quad (\text{A.4})$$

Step 4: Figure out the total effective green time G_e by Equation (A.5):

$$G_e = C - L \quad (\text{A.5})$$

Step 5: Figure out the effective green time of phase l g_{el} by Equation (A.6):

$$g_{el} = G_e \frac{y_l}{Y} \quad (\text{A.6})$$

Step 6: Figure out the actual green time of phase l g_{al} by Equation (A.7):

$$g_{al} = g_{el} + A \quad (\text{A.7})$$

A : the yellow time (s), usually 3 seconds.